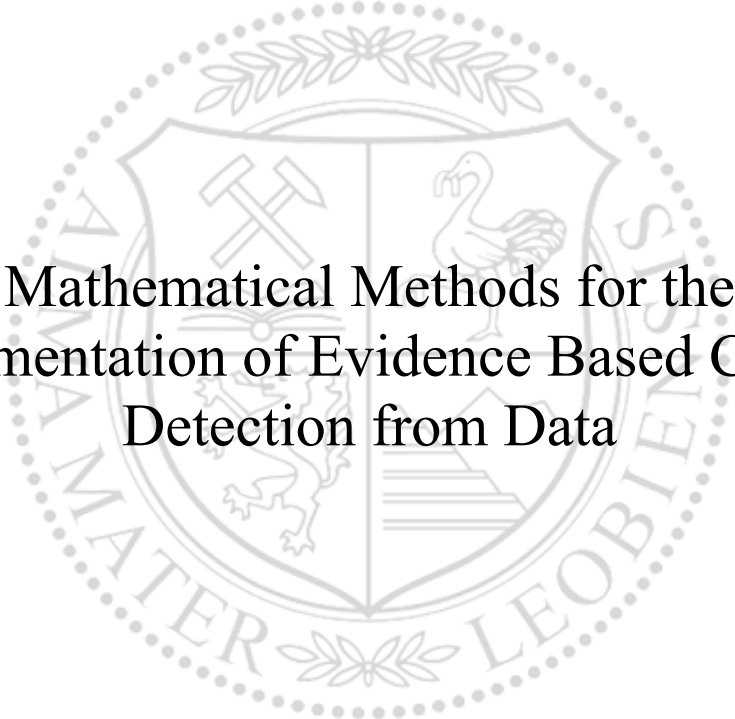




Chair of Automation

Doctoral Thesis



Mathematical Methods for the
Implementation of Evidence Based Change
Detection from Data

Dimitar Ninevski

November 2022



AFFIDAVIT

I declare on oath that I wrote this thesis independently, did not use other than the specified sources and aids, and did not otherwise use any unauthorized aids.

I declare that I have read, understood, and complied with the guidelines of the senate of the Montanuniversität Leoben for "Good Scientific Practice".

Furthermore, I declare that the electronic and printed version of the submitted thesis are identical, both, formally and with regard to content.

Date 09.11.2022

Signature Author
Dimitar Ninevski

Abstract

This thesis presents a collection of methods on the topic of evidence based change detection from discrete data. Some of the methods deal with detecting changes in the data and in its derivatives. Others are focused on modeling systems with the purpose of condition monitoring. The goal in all of them is to monitor and analyze the changes in behavior of mechanical systems governed by physical principles, which themselves are described by differential equations. The methods are demonstrated through a series of peer reviewed papers. The thesis is divided in the following chapters: polynomial methods, optimal control, the variable projection method, and industrial applications.

Matrix-algebraic formulations of polynomial problems allow for the development of new approaches in analyzing data. Using double-sided constrained Taylor approximations of discrete data, a measure of discontinuity is developed similar in concept to Lipschitz continuity. Additionally, using discrete orthogonal polynomials, a generalized framework for solving constrained inverse problems emanating from cyber-physical systems is presented.

With respect to optimal control, new algebraic formulations for discretizing the Euler-Lagrange equations are presented. These provide good approximations to solutions for numerically and physically stiff systems.

Further, it was shown how to use the variable projection method to model periodic functions with, or without a background signal. In this manner, spectral leakage and the Gibbs error were avoided, which led to highly stable and numerically efficient solutions.

Finally, through a consequent use of mathematical methods, the desired information was extracted from large volumes of industrial data, which can be corrupted by both statistical and systematic noise. With this, the interdisciplinary nature of data analysis in industrial applications was demonstrated.

Contents

1	Introduction	7
2	Polynomial Methods	9
2.1	Introduction to Polynomials	9
2.1.1	Linear Approximation	11
2.1.2	Polynomial Approximation	13
2.1.3	The Invertibility of the Matrix $\mathbf{A}^T \mathbf{A}$	14
2.1.4	Taylor Series	14
2.2	Discrete Data Analysis	16
2.2.1	Derivatives using Polynomials	16
2.2.2	Constrained Optimization	17
2.2.3	Discrete Orthogonal Polynomials	18
2.2.4	Covariance Propagation	21
	Detection of Derivative Discontinuities in Observational Data	23
1	Introduction	24
1.1	State of the Art	24
1.2	The New Approach	25
2	Detecting C^n Discontinuities	25
3	Constrained and Coupled Polynomial Approximation	27
3.1	Covariance Propagation	29
4	Error Analysis	29
4.1	Approximation Error	29
4.2	Combined Error	30
4.3	Extrapolation Error	31
5	Numerical Testing	31
5.1	Synthetic Data	31
5.2	Sensor Data	32
6	Conclusion and Future Work	33
	Bibliography	34

A Convolutional Method for the Detection of Derivative Discontinuities	36
1 Introduction	36
2 Detecting C^n discontinuities	37
3 Algebraic Formulation	39
4 Orthogonal Residualization	40
5 Example Convolutional Sequences	41
6 Test Applications	41
6.1 Sensor Data	41
6.2 Cubic Spline Data	41
7 Other Physics Relevant Features	42
7.1 Convolutional Computation of Regularized Derivatives	42
7.2 Curvature	43
Bibliography	44
A Computational Framework for Generalized Constrained Inverse Problems	46
1 Introduction	46
2 Task at hand	48
3 Algebraic formulation	49
3.1 Pseudo-spectral derivative constraints	50
3.2 Derivatives of the basis	50
3.3 Relational constraints	51
3.4 The constrained basis functions	51
3.5 Covariance propagation	53
4 Numerical testing	53
5 Experimental verification	55
6 Conclusions	56
Bibliography	58
3 Optimal Control	61
1 Collocative and Interstitial Numerical Derivatives	61
1.1 Interstitial Derivatives	62
2 Singular Value Decomposition	62
2.1 Range and Rank of \mathbf{A}	63
2.2 Nullspace of \mathbf{A}	63
3 SVD and the Least Squares Problem	64
4 Numerical Solutions to Linear Systems of ODEs	65
5 Calculus of Variations	67
5.1 Euler-Lagrange Equations	67
5.2 Calculus of Variations in Optimal Control	68

A Novel Method for Solving an Optimal Control Problem for a Numerically Stiff Independent Metering System	72
1 Introduction	72
2 System Model	73
2.1 Mechanical System	73
2.2 Hydraulic System	74
2.3 System dynamics	74
3 Solution of the problem	76
3.1 Numerical Solution	77
4 Computation and Results	79
4.1 Computation	79
4.2 Results	79
5 Conclusions	81
Bibliography	82

Multidimensional Trajectory Tracking for Numerically Stiff Independent Metering System	84
1 Introduction	84
2 System Model	87
2.1 Mechanical System	87
2.2 Hydraulic System	87
2.3 Dynamics of the system	87
3 Solution of the problem	89
3.1 Numerical Solution	90
4 Computation and Results	92
4.1 Computation	92
4.2 Results	93
5 Conclusions and Future Works	96
5.1 Conclusions	96
5.2 Future Works	96
Bibliography	96

4 The Method of Variable Projections 99

Estimating Parameters of a Sine Wave by the Method of Variable Projection	102
1 Introduction	102
2 The four parameter sine-model in algebraic form	103
3 Method of variable projection	104
4 Range of convergence	105
5 Covariance propagation and noise bandwidth	107

6	Numerical testing	108
6.1	Residual error	109
6.2	Estimated frequency	109
6.3	Computational time	110
6.4	Residual error vs phase	111
7	Code implementation	111
8	Further improvements	112
9	Conclusions	113
	Bibliography	114

Decomposition of a Periodic Perturbed Signal

	with Unknown Perturbation Frequency by the Method of Variable Projection	116
1	Introduction	117
2	Separation Model	118
2.1	Periodic Component	118
2.2	Aperiodic Component	118
2.3	Combined Basis	119
3	Separation Procedure	119
4	Numerical Testing	121
4.1	Estimated frequency	122
4.2	Computational time	123
4.3	Separation task	124
5	Conclusion	126
	Bibliography	128

Real-Time Identification of Periodic Signals

	Using the Recursive Variable Projection Algorithm	131
1	Introduction	131
2	Recursive Variable Projection Algorithm	133
2.1	Signal Model	133
2.2	Recursive Levenberg-Marquardt Algorithm	133
2.3	Method of Variable Projection	135
2.4	Linear Parameter Update	137
2.5	Complete Algorithm	137
3	Numerical Testing	137
4	Experimental Verification	139
4.1	Signal with Phase Shift	139
4.2	Signal with Frequency Step	142
5	Conclusion	142
	Bibliography	145

5	Industrial Applications	148
1	Dimensionality Reduction	148
1.1	Principal Component Analysis	148
1.2	Partial Least Squares	149
2	Canonical Correlation Analysis	151
	Measurement of Relative Position and Orientation using UWB	154
1	Introduction	155
2	Experimental setup	156
3	Characterizing UWB measurement errors	156
4	Trilateration and Uncertainty	159
5	Positioning models	162
6	Constrained, covariance weighted, least squares model	163
7	Conclusions	164
	Bibliography	166
	Computational Methods for the Detection of Wear and Damage to Milling Tools	168
1	Introduction	169
2	Experimental Data Acquisition	171
3	Data pre-processing	173
3.1	Segmentation and resampling	174
4	Data analysis	174
4.1	Median curve and its properties	174
4.2	Torsion and tension	175
4.3	Comparison across multiple parts	176
4.4	Determining rotation and relative work	178
4.5	Fourier Spectrum and Fourier Elliptical Descriptors	178
4.6	Comparison of tool runout	179
4.7	Shape factor	180
5	Statistical changes in the data	180
6	Optical inspection	182
7	Discussion	184
8	Conclusion	186
	Bibliography	187
	Sensor-based Particle Size Determination of Shredded Mixed Commercial Waste based on two-dimensional Images	189
1	Introduction	190
2	Materials and methods	192
2.1	Preparation of material	194

2.2	Image acquisition and processing	194
2.3	Calculation of particle descriptors	197
2.4	Regression model	199
3	Results and discussion	201
4	Conclusions	205
	Bibliography	207
A Digital Twin for Deep Vibro Ground Improvement		211
1	Introduction	212
2	Current State of Ground Improvement Monitoring	212
3	Conceptual Design of the Toolbox	216
4	Implementation Strategy	219
5	Case Study	220
6	Conclusions and Outlook	222
	Bibliography	223
6	Summary and Future Work	225
1	Summary	225
2	Future Work	226

List of Figures

1	An example of interpolating (left) and approximating (right) noisy data. It can be seen that approximation is better when dealing with noisy data.	10
1	Schematic of a finite set of discrete observations (dotted circles) of a continuous function. The span of the observation is split into a left and right portion at the interstitial point (circle), with lengths l_L and l_R respectively. The left and right sides are considered to be the functions $f(x)$ and $g(x)$; modelled by the polynomials $f(x, \alpha)$ and $g(x, \beta)$ of degrees d_L and d_R	26
2	Schematic of the approximations around the interstitial point. Red: left polynomial approximation $f(x, \alpha)$; dotted red: extrapolation of $f(x, \alpha)$ to the RHS; blue: right polynomial approximation, $g(x, \beta)$; dotted blue: extrapolation of $g(x, \beta)$ to the LHS; ε_i is the vertical distance between the extrapolated value and the observation. The approximation is constrained with the conditions: $f(0, \alpha) = g(0, \beta)$ and $f'(0, \alpha) = g'(0, \beta)$	30
3	A piecewise polynomial of degree $d = 2$, created from the knots sequence $\mathbf{x}_k = [0, 0.3, 0.7, 1]$ with the corresponding values $\mathbf{y}_k = [0, 0.3, 0.7, 1]$. The end points are clamped with $y'(x)_{0,1} = 0$. Gaussian noise is added with $\sigma = 0.05$. Top: the circles mark the known points of C^2 discontinuity; the blue and red lines indicate the detected discontinuities; additionally the data has been approximated by the b-spline (red) using the detected discontinuities as knots. Bottom: shows $\Delta t_{fg}^{(n)} = t_f^{(n)} - t_g^{(n)}$, together with the two identified peaks.	32
4	The top-most graph shows a function $y(x)$, together with the detected C^1 discontinuity points. The middle graph shows the difference in the Taylor polynomials $\Delta t_{fg}^{(n)}$ calculated at every interstitial point. The red and blue circles mark the relevant local maxima and minima of the difference respectively. According to this, the red and blue lines are drawn in the top-most graph. The bottom graph shows the approximation error evaluated at every interstitial point.	33
5	The two error functions, E_e and E_{fg} as defined in Section 4, for the example from Fig. 4. One can see that the location of the peaks doesn't change, and the two errors don't differ significantly.	33

1	Principle of the computation at an interstitial point. The data to the left and right of the interstitial points are considered to be the functions $f(x)$ and $g(x)$; modelled by the polynomials $f(x, \alpha)$ and $g(x, \beta)$. The coupled approximation ensured that $f(x)$ and $g(x)$ are C^{n-1} continuous at the interstitial point.	38
2	The convolution sequences s_n for C^n discontinuity detection, each with support length of $l_s = 100$ points: (top) C^2 -; (middle) C^3 - and (bottom) C^4 -, discontinuity. Note if n is odd then s_n is anti-symmetric and for even n , s_n is symmetric. This ensures that the convolution operators have liner phase response.	41
3	(Top) - sensor data obtained during the observation of a geo-drilling process. (Middle) - Measure for the C^1 discontinuity and (bottom) measure of the C^2 discontinuity. These were computed with a support length $l_s = 10$	42
4	(Top) - A cubic spline $f(x)$ with double knots at $x = 2$ and $x = 3$. This ensures that $f(x)$ is C^2 discontinuous at those points. (Middle) - $f'(x)$, with it's C^1 discontinuities at the double knots. (Bottom) - a measure of the C^2 discontinuity of $f(x)$. This was computed with a support length $l_s = 10$	42
5	The convolutional computation of the first and second order derivatives as well as the curvature $\kappa(x)$ for the same data as shown in Figure 3. These computations are performed with a total support length of $l_s = 11$, i.e., a colocative result.	44
1	Example of a cyber-physical measurement system: a low number of spatially distributed sensors are used to monitor a tunnel structure, during construction in the vicinity. The figure shows from left to right: 1) Schematic of the tunnel profile; 2) Locations of the individual sensors on the tunnel supporting strut; 3) Concept for mounting the sensors of the strut and 4) a schematic reconstruction. The aim is to monitor the bending of the structural beam as a function of time. This involves solving the constrained inverse problem and interpolating to obtain the complete bending curve. In this example the constraints and sensors are not collocated, whereas the interpolation points possibly are; consequently, both interstitial and colocative methods are required.	48
2	Top: Example of a cantilever with additional support: This is a generic example, since arbitrary constraints associated with a specific structure are possible. Bottom: The first four homogeneously constrained discrete polynomials from B_c corresponding to the constraints defined above.	52

3	Results of the Monte Carlo test of the reconstruction algorithm for the cantilever shown in Figure 2. There were $n = 1000$ synthetic data sets, each with i.i.d. Gaussian noise with $\sigma = 5e - 3$, yielding a $SNR = 1.91$. The results of the reconstructions of the function $y(x)$, the first $y'(x)$ and second derivatives $y''(x)$ are organized from left to right. Top: result of the constrained approximation. Bottom: an unconstrained approximation. Red: Is the median value of the reconstructions; black: the upper and lower statistical bounds for outlier detection, i.e., $q_{75} + 1.5IQR$ and $q_{25} - 1.5IQR$. The circles indicate the constraint values in the respective domains, i.e., in the function values and its derivatives. The constrained reconstruction has zero error and no propagation of noise at the respective constraints.	54
4	Relative local uncertainty along the curve for: $y(x)$ (top), $y'(x)$ (middle) and for $y''(x)$ (bottom), for iid noise at the input. There results are consistent with those obtained by the Monte-Carlo simulation, see Fig. 3.	55
5	Laboratory test setup used to verify the method, consisting or a bending beam with an additional support. A light sectioning head on a linear drive to measure the actual deflection of the beam.	56
6	Reconstruction of the bending curve form the experimental data. Note: there are some measurements with larger errors, this is to be expected and desirable when testing the stability of the measurement algorithm.	57
7	Reconstruction of the bending curve using a subset $n = 10$ of the measurements selected from the data shown in Figure 6. Note the irregular spacing of the measurement data. This result verifies the stability of the algorithm with a low number of observations.	58
1	A hydraulic system with two independent metering valves, including a flow controller on the actuating (A) and a pressure controller on the back side (B). The load (m) of the system is position controlled. Auxiliary components, like pressure compensators and load sensing are not shown in this figure.	74
2	Block diagram for the system presented in Fig. 1. The system is controlled on both sides, where y_{set} and $P_{B,set}$ are the desired position and pressure values. Here k_A and k_B are the proportional parts of the PID controllers for both sides respectively. . . .	75
3	Results of the first simulation where all initial and final values are set to zero, except for the end position which is set to $y_1 = 0.01 m$. For the position and the pressure P_B it can be seen that the LQR and PID controller overshoot and will not end at the desired values. On the other hand, the optimal control algorithm meets the set limits at the exact positions. It also shows improved energy performance compared to PID and LQR, which can be seen in the curves of the velocity and the pressure P_A	80

4	Results of the second simulation where the $t \in [0, 0.6]s$. All initial and final values are set to zero, except for the final position which is set to $y_1 = 0.01m$ and the pressure which is set to $P_B = 10 \cdot 10^5 Pa$. For the position and the pressure P_B it can be seen that the LQR and PID controller have a steady state error and will not end at the desired values. On the other hand, the optimal control algorithm meets the set limits at the exact positions. It also shows improved energy performance compared to PID and LQR, which can be seen in the curves of the velocity and the pressure P_A	81
1	Simplified model of the hydraulic and mechanical system. The mechanical system is position controlled. The hydraulic system consist of two independent metering valves which are flow controlled (piston side) and pressure controlled (rod side). Auxiliary components, like pressure compensators and load sensing are not shown in this figure.	86
2	State of the art block diagram for the system presented in Fig. 1. The system is controlled on both sides, where y_{set} and P_{Bset} are the desired position and pressure values. Here k_A and k_B are the proportion parts of the PID controllers for both sides accordingly.	88
3	Implementation of the calculated control variables u_1 and u_2 into the simulation model of the system as a feed-forward.	93
4	(Top and middle graph) - Results of the path tracking algorithm for time $t \in [0, 4]s$. The states x_3 and x_2 (position and pressure respectively) are being tracked. The followed values and the simulated values deviate only slightly from the desired values. (Bottom graphs) - The behavior of the remaining two states of the system.	94
5	Error curves for the followed and simulated positions and pressures in Fig. 4 accordingly. It is clear that the inaccuracy for both graphs is smaller than 1%.	94
6	(Top and middle graph) - Results of the path tracking algorithm for time $t \in [0, 8]s$. The states x_3 and x_2 (position and pressure respectively) are being tracked. (Bottom graphs) - The system is tracking the same distance over a longer period of time, which results in smaller pressure values on the piston side and a decreased velocity.	95
7	Error curves for the followed and simulated positions and pressures in Fig. 6 accordingly. Here the path tracking method shows maximum offset of 0.005%.	95
1	Normalized cost functions $E(\omega)$ and $E_d(\omega)$ for integer $c_r = 10$. Note the C^1 discontinuity at the optimum.	105
2	Normalized cost functions $E(\omega)$ and $E_d(\omega)$ for non-integer $c_r = 10.5$. Note how the cost functions diverge slightly at the optimum.	106
3	Normalized cost functions $E(\omega)$ and $E_d(\omega)$ for low value of c_r , in this example $c_r = 0.5$. Note the minimum of $E_d(\omega)$ does not correspond to the c_r values.	106
4	Condition number $\kappa(\mathbf{B}(\omega))$ in a log scale as function of ω	107

5	Synthetic sine wave used to demonstrate the computation of covariances. The model parameters were: $\omega = 4.15$, $d = 0.5$, $a_c = a_s = 1/\sqrt{2}$ with iid noise with a standard deviation of $\sigma = 0.5$ and $n = 1000$ samples were used. The estimation results are: $\omega = 4.1530$, $d = 0.4837$, $a_c = 0.7064$ and $a_s = 0.7058$. The corresponding covariance of the linear coefficients is shown in Table 4.1.	108
6	Box plots of cost functions obtained from Monte Carlo simulation with $m = 500$ repetitions: (top) the new solution, (bottom) Chen's implementation [14]; note, that where no data is shown, Chen's method failed to converge at least once during the m simulations. The signals over the given frequency range were all generated with $n = 1000$ samples, and the parameters $d = 0.1$, $a_c = 0$, $a_s = 1$	109
7	Box plots for the error in the estimated frequency, results are from the same Monte Carlo simulations shown in Figure 6. No results are shown where Chen's method failed to converge.	110
8	Box plots for the execution times t_e for each algorithm: (top) new solution, (bottom) Chen. Note, the times for Chen's algorithm, although it did not converge, are shown, since the algorithm must run to this extent to be able to detect non-convergence. The results are from the same Monte Carlo simulations shown in Figure 6.	110
9	Box plots for the residual error $\varepsilon(\phi)$ with $c_r = 0.5$ for phase shifts in the range $0 \leq \phi \leq 2\pi$. Only the results for the new algorithm are shown, since this value of c_r the Chen approach does not converge.	111
1	First synthetic test signal, consisting of the periodic component $y_p(t)$ here with the number of cycles in record $c_r = 6.25$ and the trend component $y_t(t)$ created by a DOP of degree four. The final test signal $y(t)$ is the sum of those two components superimposed with i.i.d. Gaussian noise with a standard deviation of $\sigma = 0.02$. . .	122
2	Second synthetic test signal, consisting of the periodic component $y_p(t)$ here with the number of cycles in record $c_r = 6.25$ and the trend component $y_t(t)$ created by a DOP of degree 13. The final test signal $y(t)$ is the sum of those two components superimposed with i.i.d. Gaussian noise with a standard deviation of $\sigma = 0.02$. . .	123
3	Box plots of the estimated frequency deviation for the first test signal, which has an aperiodic component modeled by a DOP of degree four, obtained from Monte Carlo simulations: (top) the presented approach based on the Eckart-Young-Mirsky matrix approximation (Equ. (4.51)), (bottom) the classical variable projection approach (Equ. (4.37)).	124
4	Detailed view of Fig. 3 for $2.75 \leq c_r \leq 9.75$	124

5	Box plots of the the estimated frequency deviation for the second test signal, which has an aperiodic component modeled by a DOP of degree 13, obtained from Monte Carlo simulations: (top) the presented approach based on the Eckart-Young-Mirsky matrix approximation (Equ. (4.51)), (bottom) the classical variable projection approach (Equ. (4.37)).	125
6	Box plots of the execution times for the first test signal, which has an aperiodic component modeled by a DOP of degree four, obtained from Monte Carlo simulations: (top) the presented approach based on the Eckart-Young-Mirsky matrix approximation (Equ. (4.51)), (bottom) the classical variable projection approach (Equ. (4.37)).	125
7	Box plots of the execution times for the second test signal, which has an aperiodic component modeled by a DOP of degree 13, obtained from Monte Carlo simulations: (top) the presented approach based on the Eckart-Young-Mirsky matrix approximation (Equ. (4.51)), (bottom) the classical variable projection approach (Equ. (4.37)).	126
8	First synthetic test signal, consisting of the periodic component $y_p(t)$ here with the number of cycles in record being 6.25 and the trend component $y_t(t)$ created by a polynomial of degree four. The final test signal $y(t)$ is the sum of those two components superimposed with i.i.d. Gaussian noise.	127
9	Second synthetic test signal, consisting of the periodic component $y_p(t)$ here with the number of cycles in record being 6.25 and the trend component $y_t(t)$ created by a polynomial of degree 13. The final test signal $y(t)$ is the sum of those two components superimposed with i.i.d. Gaussian noise.	128
1	Box plots of of the parameter estimation error $\Theta = \ \hat{\theta} - \theta\ $ obtained from Monte Carlo simulations with $m = 500$ repetitions: (top) proposed RVP algorithm, (bottom) MGN algorithm [12]; note, that simulations where the algorithm failed to converge are not considered here. The signals with different SNR were all generated with $n = 200$ samples, and the parameters $\omega = 100\pi$, $a_c = 0.732$, $a_s = 0.682$ and $d = 0$	138
2	Comparison of the RVP and MGN algorithm with respect to their success rate depending on the SNR. Success rate is defined as the number of simulations for which the algorithm converged divided by the total number of simulations.	139
3	Laboratory test setup for the generation and acquisition of the test signals: (a) National Instruments data acquisition USB box, (b) signal conditioner, (c) 1-axis acceleration reference-sensor, (d) 10 DOF smart sensor system including a 3-axis accelerometer, (e) electrodynamic shaker.	140
4	Convergence of the parameter estimate for a measurement signal with a step change in the linear parameters a_c and a_s , using the RVP algorithm. After the covariance resetting at $t = 3.6$ s it takes about one cycle to converge to the new parameter values.	141

5	Measurement data with a change in the phase used to demonstrate the tracking ability of the RVP algorithm. (top) The measurement data y with a change in the phase at $t = 3.6$ s and the signal model determined by the RVP algorithm. (bottom) Residual r of the signal model.	142
6	Convergence of the parameter estimate for a measurement signal with a step change in the linear parameters a_c and a_s as well as the nonlinear parameter ω , using the RVP algorithm. After the covariance resetting at $t = 4.9$ s it takes about one second for all parameters to converge to the new values.	143
7	Measurement data with a change in the frequency, phase and amplitude of the signal used to demonstrate the tracking ability of the RVP algorithm. (top) The measurement data y with a change in the signal change at $t = 4.9$ s and the signal model determined by the RVP algorithm. (bottom) Residual r of the signal model. .	144
1	The used UWB modules on a tripod-mount in a previous experiment of range tests.	155
2	Measurement setup with two containers at a ship quay.	156
3	The two containers C-I and C-II with UWB module positions P_1, P_2 and Q_1, Q_2 , the 4 measured distances $l_{11} l_{12} l_{21} l_{22}$ and the calculated parameters $\delta x, \delta y$ and ϕ .	157
4	Box-and-whiskers plots of the measurement errors. The measurements are used to determine the position of the moving container between 2m to 10m in 1m intervals. The red dots are outliers. The offsets of the straight-line measurement errors ϵ_{11} and ϵ_{22} at at 2 m are smaller than the offsets of ϵ_{12} and ϵ_{21} of the diagonal measured distances.	157
5	Box-and-whiskers plots of the errors centred at their respective medians to visualize the difference in their spread.	158
6	The joint histograms of the measurements $[l_{11}, l_{21}]$ and $[l_{12}, l_{22}]$ taken at 4m. The shape resembles that of a bivariate normal distribution (considering the discretization of the measurements) and the Henze-Zirkler's test confirms this at the 95% significance level, as can be seen from the p-values.	159
7	The positioning results of uncertain measurements: The analytical result of possible positions is defined as the area bounded by the dashed circles. The practical results measured in this experiment are shown as black dots. Based on the principle of triangulation the measured lengths are interpreted as circles centred at P_1 and P_2 . The reference distances are represented by l_{11} and l_{21} . The uncertain distances are limited by $l_{11} \pm 2\sigma$ and $l_{21} \pm 2\sigma$, where σ is the standard deviation of all measurements. Both shapes are almost identical.	160

8	The accuracy of the distance measurement at 5 m depending on different antenna angles. The blue crosses are the differences of the median of the measured distances and the reference. The red line is a polynomial of degree 5, which approximates the data. The dashed lines are 2σ confidence intervals, where σ is the standard deviation of each measurement.	162
9	The comparison of positioning results for 3 different methods: Trilateration, constrained approximation and weighted constrained approximation using the lengths l_{11} , l_{12} , l_{21} and l_{22}	164
10	Comparison of the errors ϵ_d of positioning using different methods. The covariance weighting (red box-plots) increases the positioning accuracy compared to pure trilateration (blue box-plots) and unweighted approximation (green box-plots).	165
1	Schematic representation of relative spatial arrangement of the tool and workpiece. (Left) Shows the cutting orientation of the tool and (Right) defines the cutting parameters: a_p in the axial- and a_e in the lateral-direction, relevant for Job 3.	171
2	The first 20% of a complete time-series for one cut lane as acquired from the machine. The data is sampled at $1.6kHz$ and a typical data set has $m = 15000 \dots 20000$ samples depending on the length of the lane being cut. Here the <i>cutting-in</i> process can be seen, i.e., the time where the milling tool has not reached the phase with constant a_p , see Figure 1b. A similar phase can be observed at the end of the lane. This data set is from Part-1, Job-3 at Lane-2. The red dots indicate points where there was a brief loss of data, due to interference of the transmission.	172
3	Flower pattern for the time-series data shown in Figure 2. The colour of the scatter plot is used to represent normalized time when the data was created during the cutting process. The arms apparently spiraling into and out of the center are associated with the cutting-in and cutting-out phases.	173
4	Polar plot of the segmented and resampled moments in x and y . Note, the colour indicates the probability density of a measurement at that specific angle. That is the density of data points at a specific angle. The width of the line indicates the interquartile range at each angle, in this case it is almost uniform. This is for the data shown in Figure 2	175
5	Torsion (Left) and tension (Right) from the data binned into angular portion $\Delta\phi = 1^\circ$, i.e. 360 bins per revolution. Each figure shows the median values together with the Q25 and Q75 quantiles.	176
6	The median curves for all 18 parts manufactured. The data is in all cases for job 3 lane 2, so that it is compatible with previous figures. It can be observed that: the pattern rotates with the increasing number of parts produced, the area enclosed by the median curves grows larger and the pdf spreads more evenly.	177

7	Left scale: Work required to mill the lane relative to part one. Right scale: the relative rotation of the median curve. There are eight lanes milled for each of the 18 parts. The first lane in each part may have some variation due to misalignment of the metal block relative to the tool path.	177
8	The first $n = 32$ harmonic components of the moment $m(t)$. This spectrum is free from Gibbs and spectral leakage error, since the data is perfectly cyclic. Two main issues can be determined directly from the spectrum. 1) The fourth harmonic is due to the mill having four flutes. This harmonic and multiples of it (i.e. 4, 8, 12,...) describe the shape of the moment in terms of the Elliptical Fourier Descriptors EFD. It is a relative measure and independent of scale and rotation. 2) The first harmonic is that portion that changes once per revolution.	178
9	The first harmonic (runout) as a function of part and lane milled. Note: part 16 has an exceptional runout.	179
10	The ratio of the power in the fourth harmonic to the power in the first 7-overtones. This is a shape factor which is scale and phase invariant. As can be seen, the signal becomes more sinusoidal up to part 16. See Figure 6 for a comparison of parts 1 to 18.	180
11	Moments for parts 1 and 18. The rotation of the flower pattern is seen here as a phase shift. Additionally, the curve is more sinusoidal for part 18, indicating lower harmonic components.	181
12	Change in performance as determined by statistics. (Left) Bivariate distribution of moment and torsion for parts 1 and 13. (Right) The corresponding histograms when projected onto the first principal component.	181
13	Histograms for the first principal component of moments and torsion as a function of part and lane milled. Part number noted on the top and lane number at the bottom. Note: there is a major change in the distribution at part 13. See Figure 12 for a comparison of part 1 and 13.	182
14	Width of the respective histogram distribution (data) as a function of the number of parts manufactured, an exponential model for the data and the determination of the $3dB$ break point, $n_{3dB} = 12.1$	182
15	Magnitudes of the two dominant peaks of the bimodal histograms as a function of part number. Note: the convergence of the two values close to part 13. See Figure 12 for a comparison of distributions for parts 1 and 13.	183
16	An image of the milling tool with the region of interest marked with a red rectangle.	183
17	Edge-wear-width as measured from the images of the tool after manufacturing each part. Note: for small wear the width could not be reliably measured also due to some uncertainty introduced by volatile workpiece material transfer features on the tool surface with similar reflectivity as hard metal tool material. The wear width as a function of part production appears to be well-modelled by an exponential function.	184

18	Digital images of the milling tool after having milled different amounts of workpieces. Black arrows indicate the positions of breakouts at the cutting edge that influence the distributions of the sensor signals discussed in the current work. The increase in wear and damage feature size is visibly increasing with the number of workpieces milled, which is expected.	185
19	Optical: normalized edge-wear-width as manually measured from the images. Work: normalized relative work required to manufacture a part, measured from the bending moments. Stats: The width of the respective histogram of the first principal components.	185
1	Schematic layout of the individual applied steps with an overview of the used MATLAB functions for the image-processing step.	193
2	RBG image (left), greyscale image (middle), and binary image (right) of singlified waste particle.	196
3	Left: Particle with polygon perimeter (solid line), max. Feret diameter (dashed line) and min. Feret diameter (dotted line); right: Particle outline with exemplary bounding shapes (bounding box, bounding triangle, bounding circle (dashed line), inscribed circle of the circumscribing polygon (dash-dotted line))	199
1	Vibro-Replacement production process as an example for ground improvement. . .	213
2	Example for an element protocol of a vibro-replacement column	214
3	2D- and 3D- evaluation of the parameter "built-in gravel material per column" with the VibroScan-tool.	215
4	Geo referenced simultaneous presentation of two KPIs - one as color, one as symbol size for each point (column) of a construction site. Note: in this example, a series of planned but not executed points can be seen at the top (white circles). Furthermore, a systematic variation of the KPI (color) can be seen from the top left towards the bottom right; this reveals a systematic change in underground properties. The panels on the right permit the definition of filters for the KPIs, with the aim of detecting and marking outlier points. Additionally, points manually flagged as being exceptional can be indicated.	216
5	Heat map representation of selected KPIs for all points of the construction site (same data as in Fig. 4); sorted according to when they were executed. The non executed points can be seen on the right of this table. The heat map supports the visual detection of patterns over multiple KPIs.	217

6	Visualization of a selected sub set of the real time machine data; here for point 190. The real time machine data for each point in Fig. 4 and Fig. 5 are available in the system. They are directly linked to support the visualization of detailed machine behavior. The data is automatically segmented to indicate the penetration and compaction phases. This example shows a pre-drilled point, identified by the fact that the vibrator enters the ground without significant pull down force while the electric motor was even turned off. Anomalous points can be flagged and commented in this view.	218
7	Outliers in the KPI penetration time of the executed points ; the concentration on the left part of the site	220
8	Penetration phase KPI: ratio of traveled length to max. penetration depth of the vibrator . This is a result for a reciprocating motion being required to penetrate the ground; with this, the cause for the loss in efficiency as seen in Fig. 7 has been identified.	221
9	Element protocol of a standard point on the test site shown in Fig. 7	221
10	Element protocol of an outlier in penetration time of Fig. 7 and a high ratio in travelled length to max. penetration depth of Fig. 8 (point No. 465). Note: the vibrator need ed to be extracted during the penetration phase at time 15 15, this is due to encountering soil with an unexpectedly high stiffness. This is also evidenced in the pull down force and vibrator amperage, both indicating that a very high stiffness had be en encountered in the soil.	222

List of Tables

2.1	Comparison of the constrained and unconstrained reconstruction wrt. 2-norm of the IQR of the uncertainty obtained from the Monte Carlo simulation.	56
4.1	Table of covariances for the test case shown in Figure 5. The results are scaled by 10^3 since they were very small.	108
5.1	Correlation coefficients ρ of the measurement sets used to determine $Q_1(l_{11}, l_{21})$ and $Q_2(l_{12}, l_{22})$. The correlation coefficients are small enough to justify the assumption that the measurements are uncorrelated at any given distance.	158
5.2	The HDOP evaluated at 1 m intervals from 2 m to 10 m. Low values (< 5) correspond to smaller dilution and more precise results.	161
5.3	Standard deviations for the lengths l_{11} , l_{12} , l_{21} and l_{22} , together with and 95% confidence intervals. These results are computed over the ensemble of all measurements, i.e., there are $n = 8793$ individual measurements for each σ . The standard deviations are given in millimetres.	163
5.4	The comparison between three different methods of positioning: Unweighted r_n ; unconstrained weighted r_{uw} ; constrained weighted r_{cw} . σ_p represents the standard deviation of all positioning errors.	165

Chapter 1

Introduction

This thesis presents a series of methods on the topic of evidence based change detection from discrete data, presented through a collection of peer reviewed papers. In the beginning, the methods deal with detecting changes in the data and its derivatives, which is a fundamental task when detecting changes in systems governed by physical laws. Such systems have to be continuous by definition, so detecting discontinuities is of great importance. Later, methods focusing on modeling mechanical systems with the purpose of condition monitoring are presented. The goal is to observe the behaviors and changes of such systems, which can facilitate well-informed decision making.

The common theme throughout all methods is the rigorous matrix-algebraic formulation of the problems, which allows for various mathematical methods to be utilized when solving engineering problems. Additionally, the use of matrices and vectors allows for easy implementation of these methods using powerful tools, such as MATLAB or Python. The papers are grouped in four different chapters, according to the main methods used in them:

1. Polynomial Methods
2. Optimal Control
3. Variable Projection Method
4. Industrial Applications

The methods described in the first chapter are essential for the entire thesis. When dealing with discrete data sets, working with polynomials is a computationally efficient and mathematically accurate way to model the data locally or globally. The concise matrix-algebraic formulations allow for the definitions of new approaches for detecting discontinuities in discrete data, which can be thought of as a discrete version of the Lipschitz continuity definition. Additionally, a generalized framework for solving inverse problems emanating from cyber-physical systems was introduced, which can deal with many different kinds of data sets as well as constraints imposed on them.

In the chapter on optimal control, solutions to two optimal control problems for a numerically and physically stiff system are introduced using interstitial derivatives. Each physical system has a state space representation which has the form of a system of differential equations. The goal of optimal control is to determine the optimal input with which a system will be brought from one state to another. Since the input of a system can change with time and be a function itself, calculus of variation is a very useful tool for solving such problems. The method to use the calculus of variation and the Euler-Lagrange equations to solve these problems, instead of the Hamiltonian, has been developed at the Chair of Automation at the University of Leoben. After discretizing and taking numerical derivatives, one can transform the system of differential equations into a system of ordinary equations, which can be solved using standard methods. The added benefit of the papers presented here is in the discretization using interstitial derivatives, which enables the accurate numerical solutions of physically and numerically stiff systems.

Chapter 3 deals with modeling periodic functions without spectral leakage or Gibbs error. This is achieved using the variable projection method. It is used when the model equation is a linear combinations of nonlinear functions. The variable projection method can utilize this special structure to deal with the two problems separately, thus reducing the dimensionality of the nonlinear problem which needs to be solved. This results in highly stable, numerically efficient solutions.

Finally, Chapter 4 contains the papers focused on consequent use of mathematical methods to extract useful information from large volumes of industrial data. In some cases, the data is very noisy and one needs to separate the data from the systematic noise as well as the statistical noise. In others, the data isn't necessarily corrupted by noise, but severely affected by human behavior and as such proper classification is necessary. The chapter includes papers dealing with the following topics

- Milling and analyzing data emanating from milling processes;
- Determining position and orientation using UWB modules;
- Particle size determination of shredded mixed commercial waste;
- Analyzing data and building a digital twin for a ground improvement process.

The methods used in the papers above show the interdisciplinary nature of data science and wide applicability of mathematics in industrial applications.

Chapter 2

Polynomial Methods

This chapter addresses basic polynomial methods necessary for understanding the papers included in it. Polynomials are a very powerful tool for handling discrete data. This is due to the Weierstrass approximation theorem, which states that any real-valued function defined on a finite interval can be approximated by a polynomial to any accuracy. Hence, understanding the basics of polynomials is crucial to handling data and detecting changes in it.

2.1 Introduction to Polynomials

A *monomial* is an expression of the form $a_n x^n$ where n is a non-negative integer, a_n is called a constant, or a coefficient, and x is the variable. Consequently, a *polynomial* $p(x)$ is a sum of monomials

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = \sum_{i=0}^n a_i x^i. \quad (2.1)$$

Note that equation (2.1) represents a polynomial in one variable (monovariate), x , but polynomials can have more variables than one. The degree of a monomial is defined as the sum of the exponents of its variables. For example, $x^4 y^4 z^3$ is a monomial of degree 11. The degree or order of a polynomial is defined as the highest degree of its monomials. The polynomial in equation (2.1) is of degree n .

Suppose a dataset of m distinct discrete points is given as $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$. Additionally, let $p(x)$ be a polynomial of degree $n - 1$

$$p(x) = a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_1 x + a_0 \quad (2.2)$$

for which the coefficients a_i are unknown. In order to determine the coefficients, assuming that the polynomial goes through the given points, one has to solve the following system of m equations in

n variables

$$\begin{aligned}
 a_{n-1}x_1^{n-1} + a_{n-2}x_1^{n-2} + \dots + a_1x_1 + a_0 &= y_1 \\
 a_{n-1}x_2^{n-1} + a_{n-2}x_2^{n-2} + \dots + a_1x_2 + a_0 &= y_2 \\
 &\vdots \\
 a_{n-1}x_m^{n-1} + a_{n-2}x_m^{n-2} + \dots + a_1x_m + a_0 &= y_m
 \end{aligned}
 \tag{2.3}$$

Depending on the relationship between m and n , there are the three cases:

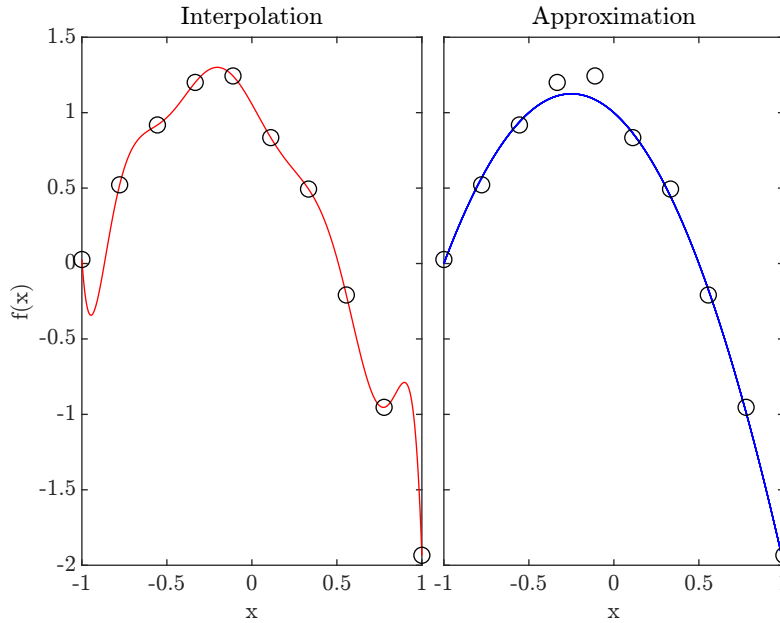


Figure 1: An example of interpolating (left) and approximating (right) noisy data. It can be seen that approximation is better when dealing with noisy data.

- If $m < n$, the system is underdetermined and there are infinitely many solutions, often called a *family of solutions*. This case is of particular interest in practical applications when dealing with constrained optimization.
- If $m = n$, the system has a unique solution and there is a single polynomial which goes through the given points. This polynomial is called an interpolating polynomial and determining the coefficients is called *polynomial interpolation*.
- If $m > n$, the system is overdetermined and there is no solution which satisfies all equations. One can solve the system approximately, by looking at polynomials which fit all the points the "best" way. This is done by minimizing some pre-defined cost function. The resulting polynomial is called an approximating polynomial and the process is called *polynomial approximation*.

The focus in the rest of this chapter will be on polynomial approximation. This is particularly useful in industrial applications when a dataset is corrupted by noise. If a noisy data set is interpolated, the resulting polynomial will not only have a higher degree than necessary, it will also not give accurate results since it follows the noisy data exactly. This is called overfitting. Hence, polynomial approximation is the better option. The difference can be seen in Fig. 1.

2.1.1 Linear Approximation

The simplest case one can start with is a polynomial of degree 1 which is just a line. Given a noisy data set,

$$\{(x_k, y_k), k = 1, \dots, n\}. \quad (2.4)$$

assume that the data comes from some true equation, but is affected by Gaussian noise, namely

$$\mathbf{y} = \mathbf{y}_t + \epsilon \quad (2.5)$$

where \mathbf{y}_t represents the true (but unknown) values, \mathbf{y} are the measured values and ϵ represents the Gaussian noise. In other words, ϵ can be thought of as a random variable having a Gaussian distribution $\epsilon \sim N(\mu, \sigma^2)$, where μ and σ represent the mean and standard deviation of the distribution respectively. The goal here is to model the data with a line,

$$\hat{\mathbf{y}} = a\mathbf{x} + b \quad (2.6)$$

where $\hat{\mathbf{y}}$ is the modelled (predicted) value. Ideally $\hat{\mathbf{y}} = \mathbf{y}_t$, However, since \mathbf{y}_t isn't known, the following difference is of interest

$$\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}. \quad (2.7)$$

If the model is good, then the residual vector \mathbf{r} should be Gaussian with mean 0. This means, that for every k

$$r_k \sim N(0, \sigma^2). \quad (2.8)$$

However, normally there is just one value for r_k and in order to check if this value comes from a normally distributed variable, one needs to utilize the likelihood function. The likelihood that a value r_k comes from a random variable $N(0, \sigma^2)$ is

$$\mathcal{L}(r_k, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{r_k^2}{2\sigma^2}}. \quad (2.9)$$

Consequently, the likelihood that all values r_1, r_2, \dots, r_n come from independent, identically distributed (i.i.d.) Gaussian variables is

$$\mathcal{L}(r_1, r_2, \dots, r_n, \sigma) = \prod_{k=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{r_k^2}{2\sigma^2}} = \left(\frac{1}{\sqrt{2\pi}\sigma} \right)^n e^{-\frac{\sum r_k^2}{2\sigma^2}} \quad (2.10)$$

This can be written in terms of the unknown parameters, a and b , using 2.7 as

$$\mathcal{L}(a, b) = \left(\frac{1}{\sqrt{2\pi\sigma}} \right)^n e^{-\frac{\sum (y_k - ax_k - b)^2}{2\sigma^2}}. \quad (2.11)$$

Taking logarithms, one gets the logarithmic likelihood function

$$\log \mathcal{L}(a, b) = -n \ln(\sqrt{2\pi\sigma}) - \frac{1}{2\sigma^2} \sum (y_k - ax_k - b)^2. \quad (2.12)$$

Since the residuals should be normally distributed, one should maximize the likelihood function, or the logarithmic likelihood (since the logarithmic function is an increasing function). Maximizing the logarithmic function is equivalent to minimizing the sum in equation 2.12. In other words, one should minimize the squares of the difference of the modeled and measured values. This is where the name **least squares** comes from. The difference can be written in vector form as

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \quad \text{or} \quad (2.13)$$

$$\mathbf{r} = \mathbf{y} - \mathbf{A}\boldsymbol{\alpha}.$$

Having this, the cost function which needs to be minimized becomes

$$\begin{aligned} C(\boldsymbol{\alpha}) &= \sum_{k=1}^n d_k^2 = \mathbf{d}^T \mathbf{d} = (\mathbf{y} - \mathbf{A}\boldsymbol{\alpha})^T (\mathbf{y} - \mathbf{A}\boldsymbol{\alpha}) = (\mathbf{y}^T - \boldsymbol{\alpha}^T \mathbf{A}^T) (\mathbf{y} - \mathbf{A}\boldsymbol{\alpha}) = \\ & \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{A}\boldsymbol{\alpha} - \underbrace{\boldsymbol{\alpha}^T \mathbf{A}^T \mathbf{y}}_{=\mathbf{y}^T \mathbf{A}\boldsymbol{\alpha}} + \boldsymbol{\alpha}^T \mathbf{A}^T \mathbf{A}\boldsymbol{\alpha} = \mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{A}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{A}^T \mathbf{A}\boldsymbol{\alpha}. \end{aligned} \quad (2.14)$$

In the previous equation, the equality $\boldsymbol{\alpha}^T \mathbf{A}^T \mathbf{y} = \mathbf{y}^T \mathbf{A}\boldsymbol{\alpha}$ is true since $\boldsymbol{\alpha}^T \mathbf{A}^T \mathbf{y}$ is a scalar, and transposing a scalar doesn't change its value. Now, using some rudimentary matrix calculus one obtains,

$$\begin{aligned} \frac{dC}{d\boldsymbol{\alpha}} &= 0 - 2 \frac{d}{d\boldsymbol{\alpha}} (\mathbf{y}^T \mathbf{A}\boldsymbol{\alpha}) + \frac{d}{d\boldsymbol{\alpha}} (\boldsymbol{\alpha}^T \mathbf{A}^T \mathbf{A}\boldsymbol{\alpha}) = \\ & -2 (\mathbf{y}^T \mathbf{A})^T + (\mathbf{A}^T \mathbf{A} + (\mathbf{A}^T \mathbf{A})^T) \boldsymbol{\alpha} = -2\mathbf{A}^T \mathbf{y} + 2\mathbf{A}^T \mathbf{A}\boldsymbol{\alpha} = 0 \end{aligned} \quad (2.15)$$

thus

$$\mathbf{A}^T \mathbf{A}\boldsymbol{\alpha} = \mathbf{A}^T \mathbf{y} \quad \text{or} \quad (2.16)$$

$$\boldsymbol{\alpha} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}. \quad (2.17)$$

¹ Note that Eq. 2.16 is called a normal equation, since $\mathbf{y} - \mathbf{A}\boldsymbol{\alpha}$ is normal to the range of \mathbf{A} . Additionally, $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \triangleq \mathbf{A}^+$ is called the Moore-Penrose pseudo inverse of the matrix \mathbf{A} .

¹Note that this solution assumes that the matrix $\mathbf{A}^T \mathbf{A}$ is invertible. This fact is discussed later in the chapter.

Now that α is calculated, the estimated values can be computed as

$$\hat{\mathbf{y}} = \mathbf{A}\mathbf{x} + b = \mathbf{A}\boldsymbol{\alpha} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}. \quad (2.18)$$

It is easy to confirm that the matrix $\mathbf{P} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T = \mathbf{A}\mathbf{A}^+$ is idempotent, i.e. has the property $\mathbf{P}^2 = \mathbf{P}$, hence it is a projection matrix projecting onto the space spanned by the columns of the matrix \mathbf{A} . Thus, the modeled values $\hat{\mathbf{y}}$ are obtained by projecting the measured values \mathbf{y} onto the space spanned by the columns of \mathbf{A} . Similarly,

$$\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}} = \left(\mathbf{I} - \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\right)\mathbf{y} = (\mathbf{I} - \mathbf{A}\mathbf{A}^+)\mathbf{y} \quad (2.19)$$

and it can be proven that the matrix $\mathbf{I} - \mathbf{A}\mathbf{A}^+$ is a projection matrix onto the orthogonal complement of the space spanned by the columns of \mathbf{A} . So, to summarize, when the measured values in \mathbf{y} are projected onto the space generated by \mathbf{A} , one gets the modeled values $\hat{\mathbf{y}}$ and when the measured values \mathbf{y} are projected onto the orthogonal complement of \mathbf{A} , one gets the residual vector \mathbf{r} .

2.1.2 Polynomial Approximation

The problem now is similar to the one before, except a polynomial of degree n is used to model the data.

$$\hat{\mathbf{y}} = \alpha_n \mathbf{x}^n + \alpha_{n-1} \mathbf{x}^{n-1} + \dots + \alpha_1 \mathbf{x} + \alpha_0. \quad (2.20)$$

In the same manner as before, it can be shown that in order to maximize the likelihood function that the error is Gaussian, one needs to minimize the sum of the squares. In other words

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} - \begin{bmatrix} x_1^n & x_1^{n-1} & \dots & x_1 & 1 \\ x_2^n & x_2^{n-1} & \dots & x_2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_m^n & x_m^{n-1} & \dots & x_m & 1 \end{bmatrix} \begin{bmatrix} \alpha_m \\ \alpha_{m-1} \\ \vdots \\ \alpha_0 \end{bmatrix} \quad \text{or} \quad (2.21)$$

$$\mathbf{r} = \mathbf{y} - \mathbf{A}\boldsymbol{\alpha}$$

and

$$C(\boldsymbol{\alpha}) = \sum_{k=1}^n d_k^2 = \mathbf{d}^T \mathbf{d} = \mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{A}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{A}^T \mathbf{A}\boldsymbol{\alpha}. \quad (2.22)$$

The matrix \mathbf{A} is called the Vandermonde matrix. The solution will have the exact same form as before, since the only thing changed is the matrix \mathbf{A} . Namely,

$$\boldsymbol{\alpha} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \quad (2.23)$$

and

$$\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}} = \left(\mathbf{I} - \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T\right) \mathbf{y}. \quad (2.24)$$

2.1.3 The Invertibility of the Matrix $A^T A$

In both solutions the fact that the matrix $A^T A$ is invertible was used. This result follows from the following theorem.

Theorem 1. If the columns of A are linearly independent, then the matrix $A^T A$ is invertible.

In the case of polynomial fitting, the columns of the matrix A are just different powers of the vector x , as can be seen in 2.21. These columns are linearly independent, since if assumed otherwise, one would get something of the form

$$x^{(m)} = \alpha_0 x^{(0)} + \alpha_1 x^{(1)} + \dots + \alpha_{m-1} x^{(m-1)} \quad (2.25)$$

which means that a polynomial of degree m is equal to a polynomial of degree $m - 1$, which is only possible if all the x 's are 0. This means that in the case of polynomial fitting, the matrix $A^T A$ is invertible. At least theoretically. If, however, the matrix looks as follows

$$A = \begin{bmatrix} 0^d & 0^{d-1} & \dots & 1 \\ 0.1^d & 0.1^{d-1} & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0.9^d & 0.9^{d-1} & \dots & 1 \end{bmatrix} \quad (2.26)$$

then for high values of d , the two columns might appear the same numerically, and as such the matrix might not be invertible numerically. In general, numerical inversions of the Vandermonde matrix are computationally expensive and the rate of inaccuracy increases very rapidly with the increase of the number of points [1, 2].

2.1.4 Taylor Series

Taylor series provide a way to approximate a function around a point, with a polynomial of arbitrary degree. This is a very useful approximation and is especially used in non-linear fitting. Now, suppose a polynomial is given as

$$p(x) = c_0 + c_1 x + c_2 x^2 \quad (2.27)$$

and one wishes to examine the behavior of this polynomial around the point a . So in order to examine that, small number h is added to a . Namely,

$$\begin{aligned} p(a+h) &= c_0 + c_1(a+h) + c_2(a+h)^2 = c_0 + c_1 a + c_1 h + c_2 a^2 + 2c_2 a h + c_2 h^2 \\ p(a+h) &= \underbrace{(c_0 + c_1 a + c_2 a^2)}_{p(a)} + h \underbrace{(c_1 + 2c_2 a)}_{p'(a)} + h^2 \underbrace{c_2}_{\frac{1}{2}p''(a)} \end{aligned} \quad (2.28)$$

In the case of a higher order polynomial, the result would have the form

$$p(a+h) = p(a) + p'(a)h + p''(a)\frac{h^2}{2!} + p'''(a)\frac{h^3}{3!} + \dots \quad (2.29)$$

This representation is true for any function which is infinitely times differentiable. Additionally, in the case of $a = 0$, the Taylor series is called a Maclaurin series.

Example 1. Here are the Maclaurin series for some standard functions.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad (2.30)$$

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{x^n}{n} \quad (2.31)$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} \quad (2.32)$$

The same can be done for a function in multiple variables. Again, the start is with a polynomial of degree two for simplicity.

$$p(x,y) = \alpha_{00} + \alpha_{10}x + \alpha_{01}y + \alpha_{20}x^2 + \alpha_{11}xy + \alpha_{02}y^2 \quad (2.33)$$

and it is examined in an area around the point (a, b) .

$$\begin{aligned} p(a+h, b+k) = & \\ & \alpha_{00} + \alpha_{10}(a+h) + \alpha_{01}(b+k) + \alpha_{20}(a+h)^2 + \alpha_{11}(a+h)(b+k) + \alpha_{02}(b+k)^2 = \\ & (\alpha_{00} + \alpha_{10}a + \alpha_{01}b + \alpha_{20}a^2 + \alpha_{11}ab + \alpha_{02}b^2) + h(\alpha_{10} + 2\alpha_{20}a + \alpha_{11}b) + \\ & k(\alpha_{01} + \alpha_{11}a + 2\alpha_{02}b) + hk(\alpha_{11}) + h^2(\alpha_{20}) + k^2(\alpha_{02}) \end{aligned} \quad (2.34)$$

and similarly as before one notices that

$$\begin{aligned} p(a+h, b+k) = & \\ & p(a,b) + p_x(a,b)h + p_y(a,b)k + \frac{1}{2}(p_{xx}(a,b)h^2 + 2p_{xy}(a,b)hk + p_{yy}(a,b)k^2) \end{aligned} \quad (2.35)$$

where p_x denotes the partial derivative with respect to x and p_{xy} denotes the second partial derivative with respect to x and y . Same as before, this can be generalized to any function, not just polynomials, in the following way:

$$f(\mathbf{x} + \Delta\mathbf{x}) = f(\mathbf{x}) + (f_{x_1}(\mathbf{x})\Delta x_1 + \dots + f_{x_m}(\mathbf{x})\Delta x_m) + \underbrace{\dots}_{\text{nonlinear part}} \quad (2.36)$$

This is particularly useful, since a very common technique for dealing with nonlinear problems is to linearize them and deal with them as linear problems later on. The Taylor expansion of a function an easy way to approximate a function by a linear function around a point. This also justifies the use of polynomials in many applications.

2.2 Discrete Data Analysis

When handling discrete sets of data, emanating from heavy machinery for example, one very often uses polynomials to model said data locally, due to their excellent approximating nature. This is partially due to the Taylor series in 2.1.4, but also due to the following fundamental mathematical theorem.

Theorem 2 (Weierstrass approximating theorem). If $f(x)$ is a continuous real-valued function defined on the real interval $x \in [a, b]$, then for every $\varepsilon > 0$, there exists a polynomial $p(x)$ such that for all $x \in [a, b]$, the supremum norm $\|f(x) - p(x)\|_\infty < \varepsilon$. That is, *any* function $f(x)$ can be approximated by a polynomial to an arbitrary accuracy ε given a sufficiently high degree.

2.2.1 Derivatives using Polynomials

Polynomials are very useful when approximating derivatives of discrete data. For example, assume a dataset is given as $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$. One way to approximate the derivative of this data at a point x_i is by interpolating the dataset with a polynomial $m - 1$ and taking the derivative of the polynomial at the point x_i . However, this isn't particularly useful in practical applications, since datasets emanating from heavy machinery are very often very long and additionally corrupted by noise. Instead, a better approach would be to approximate the data with a polynomial locally and use this to approximate the derivatives.

As an example, it is not difficult to check that a polynomial of degree 2 interpolating the first three points $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ has the form

$$p(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3. \quad (2.37)$$

The derivative of this polynomial can be calculated as

$$p'(x) = \frac{(x-x_2) + (x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1 + \frac{(x-x_1) + (x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2 + \frac{(x-x_1) + (x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3. \quad (2.38)$$

In industrial applications, the sampling rate of the sensors is very often constant, thus the emanating data is uniformly spaced, which makes calculations simpler. In this case, if one assumes that $x_1 = a, x_2 = a + h, x_3 = a + 2h$, the derivatives evaluated at the different points will have the form

$$\begin{aligned} p'(a) &= -\frac{3}{2h}y_1 + \frac{2}{h}y_2 - \frac{1}{h}y_3 \\ p'(a+h) &= -\frac{1}{2h}y_1 + \frac{1}{2h}y_3 \\ p'(a+2h) &= \frac{1}{2h}y_1 - \frac{2}{h}y_2 + \frac{3}{2h}y_3 \end{aligned} \quad (2.39)$$

which can be written compactly in matrix form as

$$\begin{bmatrix} p'(a) \\ p'(a+h) \\ p'(a+2h) \end{bmatrix} = \begin{bmatrix} -\frac{3}{2h} & \frac{2}{h} & -\frac{1}{h} \\ -\frac{1}{2h} & 0 & \frac{1}{2h} \\ \frac{1}{2h} & -\frac{2}{h} & \frac{3}{2h} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \frac{1}{2h} \underbrace{\begin{bmatrix} -3 & 4 & -1 \\ -1 & 0 & 1 \\ 1 & -4 & 3 \end{bmatrix}}_{D_1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}. \quad (2.40)$$

The benefit of using this formulation, is that the calculation only depends on the step size and the values y_i , hence one could easily generalize this calculation for the entire dataset. Namely, assuming that the quadratic interpolation is done at every three points locally, one can approximate the derivatives of the entire dataset by simply doing matrix multiplication.

$$\begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \\ \vdots \\ y'_{n-1} \\ y'_n \end{bmatrix} \approx \frac{1}{2h} \begin{bmatrix} -3 & 4 & -1 & 0 & \dots & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 0 & 1 \\ 0 & \dots & 0 & 1 & -4 & 3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} \quad (2.41)$$

Note that such calculations could also be done using polynomials of higher order. Additionally, the derivatives do not have to be evaluated at the sampling points (collocative), but could also be evaluated between the sampling points (interstitial).

2.2.2 Constrained Optimization

Constrained optimization is the problem of minimizing or maximizing a function given some constraints or conditions. In this thesis, two methods are used when dealing with constrained optimization problems: the method of Lagrange multipliers and the so-called subspace method.

Lagrange Multipliers

The method of Lagrange multipliers is used for solving constrained optimization problems of the form

Minimize $f(\mathbf{x})$ subject to the constraints $g(\mathbf{x}) = 0$.

In order to solve a problem using this method, one has to perform the following steps:

1. Formulate the Lagrange function

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x}) \quad (2.42)$$

2. Differentiate with respect to \mathbf{x} and λ

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{x}} &= \frac{df}{d\mathbf{x}} - \lambda \frac{df}{d\mathbf{x}} \\ \frac{\partial \mathcal{L}}{\partial \lambda} &= -g(\mathbf{x}) \end{aligned} \quad (2.43)$$

3. Equate the derivatives to zero

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{x}} &= 0, \\ \frac{\partial \mathcal{L}}{\partial \lambda} &= 0.\end{aligned}\tag{2.44}$$

4. Solve for \mathbf{x} and λ .

The advantage of this method is that it's very generally applicable and easy to understand. However, for each constraint one needs to introduce an additional variable and thus increasing the number of unknown variables. The subspace method circumvents this.

Subspace Method

This method has been mostly used for problems of the type

$$\min_{\alpha} \|\mathbf{y} - \mathbf{B}\alpha\| \tag{2.45}$$

$$\text{given } \mathbf{H}\alpha = \mathbf{w}. \tag{2.46}$$

From the second equation, one can write α as

$$\alpha = \underbrace{\mathbf{H}^+ \mathbf{w}}_{\text{Particular}} + \underbrace{\tilde{\mathbf{V}}\gamma}_{\text{Homogeneous}} \tag{2.47}$$

where $\tilde{\mathbf{V}}$ is a basis for the nullspace of \mathbf{H} and γ is the vector of coefficients for the remaining degrees of freedom. Consequently, this substitution transforms the minimization problem into

$$\min_{\alpha} \|\mathbf{y} - \mathbf{B}(\mathbf{H}^+ \mathbf{w} + \tilde{\mathbf{V}}\gamma)\| = \min_{\gamma} \|\mathbf{y} - \mathbf{B}\mathbf{H}^+ \mathbf{w} - \mathbf{B}\tilde{\mathbf{V}}\gamma\| = \min_{\gamma} \|\mathbf{y}_1 - \mathbf{B}\tilde{\mathbf{V}}\gamma\| \tag{2.48}$$

where $\mathbf{y}_1 \triangleq \mathbf{y} - \mathbf{B}\mathbf{H}^+ \mathbf{w}$. This transforms the problem from a constrained optimization problem into an unconstrained optimization problem of lower dimension, since the vector of unknowns γ has fewer dimensions than α .

2.2.3 Discrete Orthogonal Polynomials

Discrete orthogonal polynomials (DOP) are a type of polynomials that greatly simplify the calculations when performing any of the tasks mentioned so far, such as polynomial approximation or derivation. They simplify the algebraic formulations of said tasks as well as increase the computational speed by changing the properties of some of the matrices needed to be inverted. This text here is just a brief introduction to DOP, and for a more detailed approach to DOP, the reader is referred to [3–5].

Orthogonal Polynomials

The polynomials $p(x)$ and $q(x)$ are said to be *orthogonal* if

$$\int_a^b p(x)q(x)dx = 0. \quad (2.49)$$

Note that orthogonality can be defined with respect to any inner product, and $\int_a^b p(x)q(x)dx$ is just one example of such an inner product. For simplicity, in the remainder of the text, it is assumed that $a = 0, b = 1$. Using this definition, one can define a basis of orthogonal polynomials

$$p_0(x) = c_{00} \quad (2.50)$$

$$p_1(x) = c_{10} + c_{11}x \quad (2.51)$$

⋮

$$p_d(x) = c_{d0} + c_{d1}x + \dots + c_{dd}x^d \quad (2.52)$$

such that any other polynomial $f(x)$ can be written as a linear combination of them, namely

$$f(x) = a_0p_0(x) + a_1p_1(x) + \dots + a_dp_d(x). \quad (2.53)$$

The coefficients a_i can be calculated as [5]

$$a_i = \frac{1}{s_i} \int_0^1 f(x)p_i(x)dx, \quad \text{for } i = 0, \dots, d \quad (2.54)$$

where

$$s_i = \int_0^1 p_i^2(x)dx. \quad (2.55)$$

Since this is true for any polynomial, it must be true for the polynomial $xp_k(x)$ as well. Thus, since this is a polynomial of degree $k + 1$, it can be written as

$$xp_k(x) = a_0p_0(x) + a_1p_1(x) + \dots + a_{k+1}p_{k+1}(x). \quad (2.56)$$

The coefficients, according to equation 2.55, are calculated as

$$a_j = \frac{1}{s_j} \int_0^1 xp_k(x)p_j(x)dx, \quad \text{for } j = 0, \dots, k + 1. \quad (2.57)$$

Additionally, for the indices $j = 0, \dots, k - 2$, the term $xp_j(x)$ is a polynomial of degree smaller than k and can therefore be written as

$$xp_j(x) = b_0p_0(x) + b_1p_1(x) + \dots + b_m p_m(x), \quad \text{for } m < k. \quad (2.58)$$

Since $xp_j(x)$ is composed of the polynomials p_0, \dots, p_{k-1} at most, and all these are orthogonal to p_k , one obtains

$$a_j = 0, \quad \text{for } j = 0, \dots, k - 2. \quad (2.59)$$

Hence, one gets

$$xp_k(x) = a_{k-1}p_{k-1}(x) + a_k p_k(x) + a_{k+1}p_{k+1}(x) \quad (2.60)$$

and by rearranging this one finally gets the *three term recurrence relation* for orthogonal polynomials, namely

$$a_{k+1}p_{k+1}(x) = (x - a_k)p_k(x) - a_{k-1}p_{k-1}(x). \quad (2.61)$$

This recurrence relation is useful for generating orthogonal polynomials of very high degrees.

The Discrete Case

In the discrete case, the polynomials are represented by vectors and the discrete orthogonality condition can be written as

$$\sum_{k=1}^n p_i(x_k)p_j(x_k) = 0, \quad \text{for } i \neq j. \quad (2.62)$$

If one defines the vectors

$$\mathbf{p}_i = \begin{bmatrix} p_i(x_1) \\ p_i(x_2) \\ \vdots \\ p_i(x_n) \end{bmatrix}, \quad \text{for } i = 0, 1, \dots, d \quad (2.63)$$

and consequently the matrix \mathbf{P} as

$$\mathbf{P} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \dots \quad \mathbf{p}_d] \quad (2.64)$$

then the orthogonality condition can be written as

$$\mathbf{P}^T \mathbf{P} = \mathbf{S} = \text{diag}(s_0, s_1, \dots, s_d). \quad (2.65)$$

This is one of the reasons why discrete orthogonal polynomials are so computationally efficient. Namely, if a discrete dataset is approximated by a set of such polynomials, the coefficients will be calculated as

$$\boldsymbol{\alpha} = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T \mathbf{y} \quad (2.66)$$

as is shown in 2.1.2. However, since the matrix $\mathbf{P}^T \mathbf{P}$ is diagonal, inverting it is very simple and thus this calculation is much simpler as well.

A generalization of eq. 2.65 is

$$\mathbf{P}^T \mathbf{W} \mathbf{P} = \text{diag}(s_0, s_1, \dots, s_d), \quad (2.67)$$

which is used for *weighted discrete orthogonal polynomial approximation*. Here, the matrix \mathbf{W} contains the associated weights with the dataset. This is particularly useful when dealing with datasets with known, but varying levels of noise throughout. For more properties of discrete orthogonal polynomials and their derivatives, the reader is referred to [3–5].

2.2.4 Covariance Propagation

When fitting polynomials (or other functions) to a dataset, one might ask how the coefficients of the model change with respect to changes in the data, or in other words, how do the errors and the covariances propagate through the fitting procedure. Assume that the model coefficients can be calculated via the following equation:

$$\boldsymbol{\alpha} = \mathbf{M}\mathbf{x} + \mathbf{c}. \quad (2.68)$$

Additionally, assume that the measurements which give the vector \mathbf{x} have been performed n times, and since the vector \mathbf{x} is corrupted by noise, n different model coefficients are obtained $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_n$. The mean of the vectors $\boldsymbol{\alpha}_i$ can be calculated as

$$\boldsymbol{\mu}_\alpha = \frac{1}{n} \sum_{i=1}^n \boldsymbol{\alpha}_i = \frac{1}{n} \sum_{i=1}^n (\mathbf{M}\mathbf{x}_i + \mathbf{c}) = \mathbf{M} \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i + \frac{1}{n} \sum_{i=1}^n \mathbf{c} = \mathbf{M}\boldsymbol{\mu}_x + \frac{1}{n}n\mathbf{c} = \mathbf{M}\boldsymbol{\mu}_x + \mathbf{c} \quad (2.69)$$

and thus, the covariance matrix of the vectors $\boldsymbol{\alpha}_i$ is calculated as

$$\begin{aligned} \Lambda_\alpha &= \frac{1}{n} \sum_{i=1}^n (\boldsymbol{\alpha}_i - \boldsymbol{\mu}_\alpha) (\boldsymbol{\alpha}_i - \boldsymbol{\mu}_\alpha)^\top = \\ &= \frac{1}{n} \sum_{i=1}^n (\mathbf{M}\mathbf{x}_i + \mathbf{c} - \mathbf{M}\boldsymbol{\mu}_x - \mathbf{c}) (\mathbf{M}\mathbf{x}_i + \mathbf{c} - \mathbf{M}\boldsymbol{\mu}_x - \mathbf{c})^\top = \\ &= \frac{1}{n} \sum_{i=1}^n (\mathbf{M}\mathbf{x}_i + \mathbf{c} - \mathbf{M}\boldsymbol{\mu}_x - \mathbf{c}) (\mathbf{x}_i^\top \mathbf{M}^\top + \mathbf{c}^\top - \boldsymbol{\mu}_x^\top \mathbf{M}^\top - \mathbf{c}^\top) = \\ &= \frac{1}{n} \sum_{i=1}^n (\mathbf{M}\mathbf{x}_i - \mathbf{M}\boldsymbol{\mu}_x) (\mathbf{x}_i^\top \mathbf{M}^\top - \boldsymbol{\mu}_x^\top \mathbf{M}^\top) = \mathbf{M} \left\{ \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu}_x) (\mathbf{x}_i^\top - \boldsymbol{\mu}_x^\top) \right\} \mathbf{M}^\top. \end{aligned} \quad (2.70)$$

This leads to the following equation

$$\Lambda_\alpha = \mathbf{M}\Lambda_x\mathbf{M}^\top. \quad (2.71)$$

If the error in \mathbf{x} is i.i.d., meaning

$$\Lambda_x = \sigma^2 \mathbf{I} \quad (2.72)$$

one obtains

$$\Lambda_\alpha = \mathbf{M}\sigma^2\mathbf{I}\mathbf{M}^\top = \sigma^2\mathbf{M}\mathbf{M}^\top \quad (2.73)$$

Bibliography

- [1] A. Eisinberg, G. Franzé, and N. Salerno, “Rectangular vandermonde matrices on chebyshev nodes,” *Linear Algebra and its Applications*, vol. 338, no. 1, pp. 27–36, 2001. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S002437950100355X>
- [2] H. Wertz, “On the numerical inversion of a recurrent problem: The vandermonde matrix,” *IEEE Transactions on Automatic Control*, vol. 10, no. 4, pp. 492–492, 1965.
- [3] J. Baik, T. Kriecherbauer, K. D.-R. McLaughlin, and P. D. Miller, *Discrete Orthogonal Polynomials. (AM-164): Asymptotics and Applications (AM-164): Asymptotics and Applications (AM-164)*. Princeton University Press, 2007. [Online]. Available: <https://doi.org/10.1515/9781400837137>
- [4] G. Szegő, S. G. Szego, and A. M. Society, *Orthogonal Polynomials*, ser. American Math. Soc: Colloquium publ. American Mathematical Society, 1939. [Online]. Available: <https://books.google.at/books?id=ZOhmnsXlcY0C>
- [5] M. Harker, *Fractional Differential Equations: Numerical Methods for Applications*, ser. Studies in Systems, Decision and Control. Springer Cham.
- [6] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- [7] I. Gelfand and S. Fomin, *Calculus of Variations*, ser. Dover Books on Mathematics. Dover Publications, 2012. [Online]. Available: <https://books.google.at/books?id=CeC7AQAQBAJ>
- [8] G. H. Golub and V. Pereyra, “The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate,” *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 413–432, 1973.
- [9] G. Golub and V. Pereyra, “Separable nonlinear least squares: the variable projection method and its applications,” *Inverse Problems*, vol. 19, pp. R1–R26(1), 01 2003.

Detection of Derivative Discontinuities in Observational Data

Dimitar Ninevski⁰⁰⁰⁰⁻⁰⁰⁰³⁻⁰¹⁰¹⁻⁸⁶⁸⁶ and Paul O’Leary⁰⁰⁰⁰⁻⁰⁰⁰²⁻¹³⁶⁷⁻⁸²⁷⁰

University of Leoben, A8700 Leoben, Austria

automation@unileoben.ac.at

<http://automatiom.unileoben.ac.at>

Abstract

This paper presents a new approach to the detection of discontinuities in the n -th derivative of observational data. This is achieved by performing two polynomial approximations at each interstitial point. The polynomials are coupled by constraining their coefficients to ensure continuity of the model up to the $(n-1)$ -th derivative; while yielding an estimate for the discontinuity of the n -th derivative. The coefficients of the polynomials correspond directly to the derivatives of the approximations at the interstitial points through the prudent selection of a common coordinate system. The approximation residual and extrapolation errors are investigated as measures for detecting discontinuity. This is necessary since discrete observations of continuous systems are discontinuous at every point. It is proven, using matrix algebra, that positive extrema in the combined approximation-extrapolation error correspond exactly to extrema in the difference of the Taylor coefficients. This provides a relative measure for the severity of the discontinuity in the observational data. The matrix algebraic derivations are provided for all aspects of the methods presented here; this includes a solution for the covariance propagation through the computation. The performance of the method is verified with a Monte Carlo simulation using synthetic piecewise polynomial data with known discontinuities. It is also demonstrated that the discontinuities are suitable as knots for B-spline modelling of data. For completeness, the results of applying the method to sensor data acquired during the monitoring of heavy machinery are presented.

Keywords: Data analysis, Discontinuity detection, Free-knot splines.

1 Introduction

In the recent past *physics informed data science* has become a focus of research activities, e.g., [9]. It appears under different names e.g., *physics informed* [12]; *hybrid learning* [13]; *physics-based* [17], etc.; but with the same basic idea of embedding physical principles into the data science algorithms. The goal is to ensure that the results obtained obey the laws of physics and/or are based on physically relevant features. Discontinuities in the observations of continuous systems violate some very basic physics and for this reason their detection is of fundamental importance. Consider Newton's second law of motion,

$$F(t) = \frac{d}{dt} \left\{ m(t) \frac{d}{dt} y(t) \right\} = \dot{m}(t) \dot{y}(t) + m(t) \ddot{y}(t). \quad (2.74)$$

Any discontinuities in the observations of $m(t)$, $\dot{m}(t)$, $y(t)$, $\dot{y}(t)$ or $\ddot{y}(t)$ indicate a violation of some basic principle: be it that the observation is incorrect or something unexpected is happening in the system. Consequently, detecting discontinuities is of fundamental importance in physics based data science. A function $s(x)$ is said to be C^n discontinuous, if $s \in C^{n-1} \setminus C^n$, that is if $s(x)$ has continuous derivatives up to and including order $n - 1$, but the n -th derivative is discontinuous. Due to the discrete and finite nature of the observational data, only jump discontinuities in the n -th derivative are considered; asymptotic discontinuities are not considered. Furthermore, in more classical data modelling, C^n jump discontinuities form the basis for the locations of knots in B-Spline models of observational data [15].

1.1 State of the Art

There are numerous approaches in the literature dealing with estimating regression functions that are smooth, except at a finite number of points. Based on the methods, these approaches can be classified into four groups: local polynomial methods, spline-based methods, kernel-based methods and wavelet methods. The approaches vary also with respect to the available a priori knowledge about the number of points of discontinuity or the derivative in which these discontinuities appear. For a good literature review of these methods, see [3]. The method used in this paper is relevant both in terms of local polynomials as well as spline-based methods; however, the new approach requires no a priori knowledge about the data.

In the local polynomial literature, namely in [8] and [14], ideas similar to the ones presented here are investigated. In these papers, local polynomial approximations from the left and the right side of the point in question are used. The major difference is that neither of these methods use constraints to ensure that the local polynomial approximations enforce continuity of the lower derivatives, which is done in this paper. As such, they use different residuals to determine the existence of a change point. Using constrained approximation ensures that the underlying physical properties of the system are taken into consideration, which is one of the main advantages of the approach presented here. Additionally, in the aforementioned papers, it is not clear whether only

co-locative points are considered as possible change points, or interstitial points are also considered. This distinction between colocative and interstitial is of great importance. Fundamentally, the method presented here can be applied to discontinuities at either locations. However, it has been assumed that discontinuities only make sense between the sampled (co-locative) points, i.e., the discontinuities are interstitial.

In [11] on the other hand, one polynomial instead of two is used, and the focus is mainly on detecting C^0 and C^1 discontinuities. Additionally, the number of change-points must be known a-priori, so only their location is approximated; the required a-priori knowledge make the method unsuitable in real sensor based system observation.

In the spline-based literature there are heuristic methods (top-down and bottom-up) as well as optimization methods. For a more detailed state of the art on splines, see [2]. Most heuristic methods use a discrete geometric measure to calculate whether a point is a knot, such as: discrete curvature, kink angle, etc, and then use some (mostly arbitrary) threshold to improve the initial knot set. In the method presented here, which falls under the category of bottom-up approaches, the selection criterion is based on calculus and statistics, which allows for incorporation of the fundamental physical laws governing the system, in the model, but also ensures mathematical relevance and rigour.

1.2 The New Approach

This paper presents a new approach to detecting C^n discontinuities in observational data. It uses constrained coupled polynomial approximation to obtain two estimates for the n^{th} Taylor coefficients and their uncertainties, at every interstitial point. These correspond approximating the local function by polynomials, once from the left $f(x, \alpha)$ and once from the right $g(x, \beta)$. The constraints couple the polynomials to ensure that $\alpha_i = \beta_i$ for every $i \in [0 \dots n - 1]$. In this manner the approximations are C^{n-1} continuous at the interstitial points, while delivering an estimate for the difference in the n^{th} Taylor coefficients. All the derivations for the coupled constrained approximations and the numerical implementations are presented. Both the approximation and extrapolation residuals are derived. It is proven that the discontinuities must lie at local positive peaks in the extrapolation error. The new approach is verified with both known synthetic data and on real sensor data obtained from observing the operation of heavy machinery.

2 Detecting C^n Discontinuities

Discrete observations $s(x_i)$ of a continuous system $s(x)$ are, by their very nature, discontinuous at every sample. Consequently, some measure for discontinuity will be required, with uncertainty, which provides the basis for further analysis.

The observations are considered to be the co-locative points, denoted by x_i and collectively by the vector \mathbf{x} ; however, we wish to estimate the discontinuity at the interstitial points, denoted by

ζ_i and collectively as ζ . Using interstitial points, one ensures that each data point is used for only one polynomial approximation at a time. Furthermore, in the case of sensor data, one expects the discontinuities to happen between samples. Consequently the data is segmented at the interstitial points, i.e. between the samples. This requires the use of interpolating functions and in this work we have chosen to use polynomials.

Polynomials have been chosen because of their approximating, interpolating and extrapolating properties when modelling continuous systems: The Weierstrass approximation theorem [16] states that if $f(x)$ is a continuous real-valued function defined on the real interval $x \in [a, b]$, then for every $\varepsilon > 0$, there exists a polynomial $p(x)$ such that for all $x \in [a, b]$, the supremum norm $\|f(x) - p(x)\|_\infty < \varepsilon$. That is *any* function $f(x)$ can be approximated by a polynomial to an arbitrary accuracy ε given a sufficiently high degree.

The basic concept (see Figure 1) to detect a C^n discontinuity is: to approximate the data to the left of an interstitial point by the polynomial $f(x, \alpha)$ of degree d_L and to the right by $g(x, \beta)$ of degree d_R , while constraining these approximations to be C^{n-1} continuous at the interstitial point. This approximation ensures that,

$$f^{(k-1)}(\zeta_i) = g^{(k-1)}(\zeta_i), \quad \text{for every } k \in [1 \dots n]. \quad (2.75)$$

while yielding estimates for $f^{(n)}(\zeta_i)$ and $g^{(n)}(\zeta_i)$ together with estimates for their variances $\lambda_{f(\zeta_i)}$ and $\lambda_{g(\zeta_i)}$. This corresponds exactly to estimating the Taylor coefficients of the function twice for each interstitial point, i.e., once from the left and once from the right. If they differ significantly, then the function's n^{th} derivative is discontinuous at this point. The Taylor series of a function $f(x)$

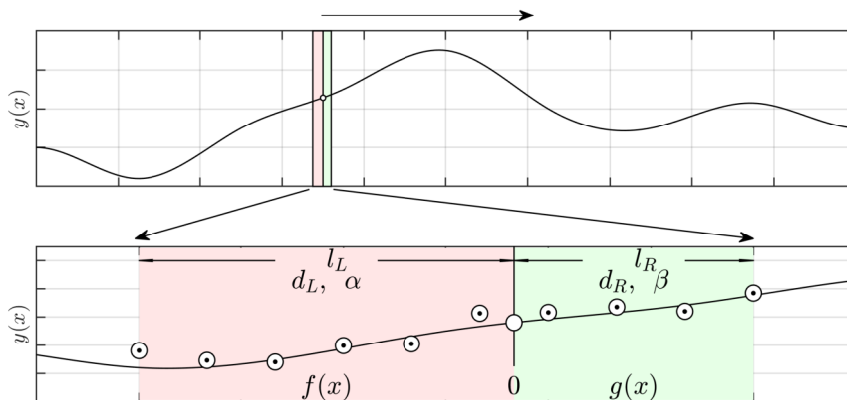


Figure 1: Schematic of a finite set of discrete observations (dotted circles) of a continuous function. The span of the observation is split into a left and right portion at the interstitial point (circle), with lengths l_L and l_R respectively. The left and right sides are considered to be the functions $f(x)$ and $g(x)$; modelled by the polynomials $f(x, \alpha)$ and $g(x, \beta)$ of degrees d_L and d_R .

around the point a is defined as,

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!} (x-a)^k \quad (2.76)$$

for each x for which the infinite series on the right hand side converges. Furthermore, any function which is $n + 1$ times differentiable can be written as

$$f(x) = \tilde{f}(x) + R(x) \quad (2.77)$$

where $\tilde{f}(x)$ is an n^{th} degree polynomial approximation of the function $f(x)$,

$$\tilde{f}(x) = \sum_{k=0}^n \frac{f^{(k)}(a)}{k!} (x-a)^k \quad (2.78)$$

and $R(x)$ is the remainder term. The Lagrange form of the remainder $R(x)$ is given by

$$R(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x-a)^{n+1} \quad (2.79)$$

where ξ is a real number between a and x .

A Taylor expansion around the origin (i.e. $a = 0$ in Equation 2.76) is called a Maclaurin expansion; for more details, see [1]. In the rest of this work, the n^{th} Maclaurin coefficient for the function $f(x)$ will be denoted by

$$t_f^{(n)} \triangleq \frac{f^{(n)}(0)}{n!}. \quad (2.80)$$

The coefficients of a polynomial $f(x, \alpha) = \alpha_n x^n + \dots + \alpha_1 x + \alpha_0$ are closely related to the coefficients of the Maclaurin expansion of this polynomial. Namely, it's easy to prove that

$$\alpha_k = t_f^{(k)}, \quad \text{for every } k \in [0 \dots n]. \quad (2.81)$$

A prudent selection of a common local coordinate system, setting the interstitial point as the origin, ensures that the coefficients of the left and right approximating polynomials correspond to the derivative values at this interstitial point. Namely, one gets a very clear relationship between the coefficients of the left and right polynomial approximations, α and β , their Maclaurin coefficients, $t_f^{(n)}$ and $t_g^{(n)}$, and the values of the derivatives at the interstitial point

$$t_f^{(n)} = \alpha_n = \frac{f^{(n)}(0)}{n!} \quad \text{and} \quad t_g^{(n)} = \beta_n = \frac{g^{(n)}(0)}{n!}. \quad (2.82)$$

From equation 2.82 it is clear that performing a left and right polynomial approximation at an interstitial point is sufficient to get the derivative values at that point, as well as their uncertainties.

3 Constrained and Coupled Polynomial Approximation

The goal here is to obtain $\Delta t_{fg}^{(n)} \triangleq t_f^{(n)} - t_g^{(n)}$ via polynomial approximation. To this end two polynomial approximations are required; whereby, the interstitial point is used as the origin in the

common coordinate system, see Figure 1. The approximations are coupled [6] at the interstitial point by constraining the coefficients such that $\alpha_i = \beta_i$, for every $i \in [0 \dots n - 1]$. This ensures that the two polynomials are C^{n-1} continuous at the interstitial points. This also reduces the degrees of freedom during the approximation and with this the variance of the solution is reduced. For more details on constrained polynomial approximation see [4, 7].

To remain fully general, a local polynomial approximation of degree d_L is performed to the left of the interstitial point with the support length l_L creating $f(x, \alpha)$; similarly to the right $d_R, l_R, g(x, \beta)$. The x coordinates to the left, denoted as x_L are used to form the left Vandermonde matrix V_L , similarly x_R form V_R to the right. This leads to the following formulation of the approximation process,

$$\mathbf{y}_L = V_L \boldsymbol{\alpha} \quad \text{and} \quad \mathbf{y}_R = V_R \boldsymbol{\beta}. \quad (2.83)$$

$$\begin{bmatrix} V_L & \mathbf{0} \\ \mathbf{0} & V_R \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_L \\ \mathbf{y}_R \end{bmatrix} \quad (2.84)$$

A C^{n-1} continuity implies $\alpha_i = \beta_i$, for every $i \in [0 \dots n - 1]$ which can be written in matrix form as

$$\left[\mathbf{0} \quad I_{n-1} \mid \mathbf{0} \quad -I_{n-1} \right] \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \mathbf{0} \quad (2.85)$$

Defining

$$\mathbf{V} \triangleq \begin{bmatrix} V_L & \mathbf{0} \\ \mathbf{0} & V_R \end{bmatrix}, \boldsymbol{\gamma} \triangleq \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}, \mathbf{y} \triangleq \begin{bmatrix} \mathbf{y}_L \\ \mathbf{y}_R \end{bmatrix} \text{ and } \mathbf{C} \triangleq \left[\mathbf{0} \quad I_{n-1} \mid \mathbf{0} \quad -I_{n-1} \right]$$

We obtain the task of least squares minimization with homogeneous linear constraints,

$$\boxed{\begin{array}{l} \min_{\boldsymbol{\gamma}} \quad \|\mathbf{y} - \mathbf{V} \boldsymbol{\gamma}\|_2^2 \\ \text{Given} \quad \mathbf{C} \boldsymbol{\gamma} = \mathbf{0}. \end{array}} \quad (2.86)$$

Clearly $\boldsymbol{\gamma}$ must lie in the null-space of \mathbf{C} ; now, given \mathbf{N} , an ortho-normal vector basis set for $\text{null}\{\mathbf{C}\}$, we obtain,

$$\boldsymbol{\gamma} = \mathbf{N} \boldsymbol{\delta}. \quad (2.87)$$

Back-substituting into Equation 2.86 yields,

$$\min_{\boldsymbol{\delta}} \|\mathbf{y} - \mathbf{V} \mathbf{N} \boldsymbol{\delta}\|_2^2 \quad (2.88)$$

The least squares solution to this problem is,

$$\boldsymbol{\delta} = (\mathbf{V} \mathbf{N})^+ \mathbf{y}, \quad (2.89)$$

and consequently,

$$\boxed{\boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \mathbf{N} (\mathbf{V} \mathbf{N})^+ \mathbf{y}} \quad (2.90)$$

Formulating the approximation in the above manner ensures that the difference in the Taylor coefficients can be simply computed as

$$\Delta t_{fg}^{(n)} = t_f^{(n)} - t_g^{(n)} = \alpha_n = \beta_n. \quad (2.91)$$

Now defining $\mathbf{d} = [1, \mathbf{0}_{d_L-1}, -1, \mathbf{0}_{d_R-1}]^T$, $\Delta t_{fg}^{(n)}$ is obtained from γ as

$$\Delta t_{fg}^{(n)} = \mathbf{d}^T \gamma = \mathbf{d}^T \mathbf{N} (\mathbf{V} \mathbf{N})^+ \mathbf{y}. \quad (2.92)$$

3.1 Covariance Propagation

Defining, $\mathbf{K} = \mathbf{N} (\mathbf{V} \mathbf{N})^+$, yields, $\gamma = \mathbf{K} \mathbf{y}$. Then given the covariance of \mathbf{y} , i.e., $\Lambda_{\mathbf{y}}$, one gets that,

$$\Lambda_{\gamma} = \mathbf{K} \Lambda_{\mathbf{y}} \mathbf{K}^T. \quad (2.93)$$

Additionally, from equation 2.92 one could derive the covariance of the difference in the Taylor coefficients

$$\Lambda_{\Delta} = \mathbf{d} \Lambda_{\gamma} \mathbf{d}^T \quad (2.94)$$

Keep in mind that, if one uses approximating polynomials of degree n to determine a discontinuity in the n^{th} derivative, as done so far, Λ_{Δ} is just a scalar and corresponds to the variance of $\Delta t_{fg}^{(n)}$.

4 Error Analysis

In this paper we consider three measures for error:

1. the norm of the approximation residual;
2. the combined approximation and extrapolation error;
3. the extrapolation error.

4.1 Approximation Error

The residual vector has the form

$$\mathbf{r} = \mathbf{y} - \mathbf{V} \gamma = \begin{bmatrix} \mathbf{y}_L - \mathbf{V}_L \alpha \\ \mathbf{y}_R - \mathbf{V}_R \beta \end{bmatrix}.$$

The approximation error is calculated as

$$\begin{aligned} E_a &= \|\mathbf{r}\|_2^2 = \|\mathbf{y}_L - \mathbf{V}_L \alpha\|_2^2 + \|\mathbf{y}_R - \mathbf{V}_R \beta\|_2^2 \\ &= (\mathbf{y}_L - \mathbf{V}_L \alpha)^T (\mathbf{y}_L - \mathbf{V}_L \alpha) + (\mathbf{y}_R - \mathbf{V}_R \beta)^T (\mathbf{y}_R - \mathbf{V}_R \beta) \\ &= \mathbf{y}^T \mathbf{y} - 2\alpha^T \mathbf{V}_L^T \mathbf{y}_L + \alpha^T \mathbf{V}_L^T \mathbf{V}_L \alpha - 2\beta^T \mathbf{V}_R^T \mathbf{y}_R + \beta^T \mathbf{V}_R^T \mathbf{V}_R \beta. \end{aligned}$$

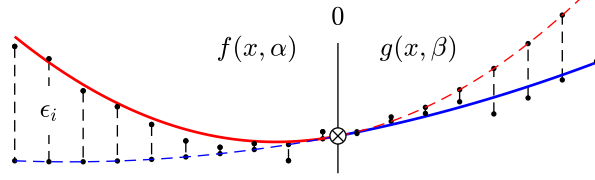


Figure 2: Schematic of the approximations around the interstitial point. Red: left polynomial approximation $f(x, \alpha)$; dotted red: extrapolation of $f(x, \alpha)$ to the RHS; blue: right polynomial approximation, $g(x, \beta)$; dotted blue: extrapolation of $g(x, \beta)$ to the LHS; ϵ_i is the vertical distance between the extrapolated value and the observation. The approximation is constrained with the conditions: $f(0, \alpha) = g(0, \beta)$ and $f'(0, \alpha) = g'(0, \beta)$.

4.2 Combined Error

The basic concept, which can be seen in Figure 2, is as follows: the left polynomial $f(x, \alpha)$, which approximates over the values x_L , is extended to the right and evaluated at the points x_R . Analogously, the right polynomial $g(x, \beta)$ is evaluated at the points x_L . If there is no C^n discontinuity in the system, the polynomials f and g must be equal and consequently the extrapolated values won't differ significantly from the approximated values.

Analytical Combined Error

The extrapolation error in a continuous case, i.e. between the two polynomial models, can be computed with the following 2-norm,

$$\epsilon_x = \int_{x_{min}}^{x_{max}} \{f(x, \alpha) - g(x, \beta)\}^2 dx. \quad (2.95)$$

Given, the constraints which ensure that $\alpha_i = \beta_i \ i \in [0, \dots, n-1]$, we obtain,

$$\epsilon_x = \int_{x_{min}}^{x_{max}} \{(\alpha_n - \beta_n) x^n\}^2 dx. \quad (2.96)$$

Expanding and performing the integral yields,

$$\epsilon_x = (\alpha_n - \beta_n)^2 \left\{ \frac{x_{max}^{2n+1} - x_{min}^{2n+1}}{2n+1} \right\} \quad (2.97)$$

Given fixed values for x_{min} and x_{max} across a single computation implies that the factor,

$$k = \frac{x_{max}^{2n+1} - x_{min}^{2n+1}}{2n+1} \quad (2.98)$$

is a constant. Consequently, the extrapolation error is directly proportional to the square of the difference in the Taylor coefficients,

$$\epsilon_x \propto (\alpha_n - \beta_n)^2 \propto \left\{ \Delta t_{fg}^{(n)} \right\}^2. \quad (2.99)$$

Numerical Combined Error

In the discrete case, one can write the errors of $f(x, \alpha)$ and $g(x, \beta)$ as

$$\mathbf{e}_f = \mathbf{y} - f(\mathbf{x}, \alpha) \quad \text{and} \quad \mathbf{e}_g = \mathbf{y} - g(\mathbf{x}, \beta) \quad (2.100)$$

respectively. Consequently, one could define an error function as

$$E_{fg} = \|\mathbf{e}_f - \mathbf{e}_g\|_2^2 = \|(a_n - b_n) \mathbf{z}\|_2^2 = (a_n - b_n)^2 \mathbf{z}^T \mathbf{z} = (a_n - b_n)^2 \sum x_i^n \quad (2.101)$$

where $\mathbf{z} \triangleq \mathbf{x} \cdot \hat{n}$. From these calculations it is clear that in the discrete case the error is also directly proportional to the square of the difference in the Taylor coefficients and that $E_{fg} \propto \epsilon_x$. This proves that the numerical computation is consistent with the analytical continuous error.

4.3 Extrapolation Error

One could also define a different kind of error, based just on the extrapolative properties of the polynomials. Namely, using the notation from the beginning of Section 3, one defines

$$\mathbf{r}_{ef} = \mathbf{y}_L - g(\mathbf{x}_L, \beta) = \mathbf{y}_L - \mathbf{V}_L \beta \quad \text{and} \quad \mathbf{r}_{eg} = \mathbf{y}_R - f(\mathbf{x}_R, \alpha) = \mathbf{y}_R - \mathbf{V}_R \alpha$$

and then calculates the error as

$$\begin{aligned} E_e &= \mathbf{r}_{ef}^T \mathbf{r}_{ef} + \mathbf{r}_{eg}^T \mathbf{r}_{eg} \\ &= (\mathbf{y}_L - \mathbf{V}_L \beta)^T (\mathbf{y}_L - \mathbf{V}_L \beta) + (\mathbf{y}_R - \mathbf{V}_R \alpha)^T (\mathbf{y}_R - \mathbf{V}_R \alpha) \\ &= \mathbf{y}^T \mathbf{y} - 2\beta^T \mathbf{V}_L^T \mathbf{y}_L + \beta^T \mathbf{V}_L^T \mathbf{V}_L \beta - 2\alpha^T \mathbf{V}_R^T \mathbf{y}_R + \alpha^T \mathbf{V}_R^T \mathbf{V}_R \alpha. \end{aligned}$$

In the example in section 5, it will be seen that there is no significant numerical difference between these two errors.

5 Numerical Testing

The numerical testing is performed with: synthetic data from a piecewise polynomial, where the locations of the C^n discontinuities are known; and with real sensor data emanating from the monitoring of heavy machinery.

5.1 Synthetic Data

In the literature on splines, functions of the type $y(x) = e^{-x^2}$ are commonly used. However, this function is analytic and C^∞ continuous; consequently it was not considered a suitable function for testing. In Figure 3 a piecewise polynomial with a similar shape is shown; however, this curve has C^2 discontinuities at known locations. The algorithm was applied to the synthetic data from

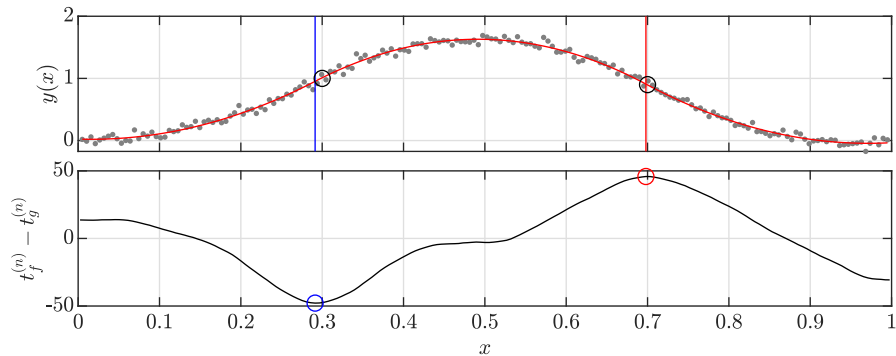


Figure 3: A piecewise polynomial of degree $d = 2$, created from the knots sequence $x_k = [0, 0.3, 0.7, 1]$ with the corresponding values $y_k = [0, 0.3, 0.7, 1]$. The end points are clamped with $y'(x)_{0,1} = 0$. Gaussian noise is added with $\sigma = 0.05$. Top: the circles mark the known points of C^2 discontinuity; the blue and red lines indicate the detected discontinuities; additionally the data has been approximated by the b-spline (red) using the detected discontinuities as knots. Bottom: shows $\Delta t_{fg}^{(n)} = t_f^{(n)} - t_g^{(n)}$, together with the two identified peaks.

the piecewise polynomial, with added noise with $\sigma = 0.05$ and the results for a single case can be seen in Figure 3. Additionally, a Monte Carlo simulation with $m = 10000$ iterations was performed and the results of the algorithm were compared to the true locations of the two known knots. The mean errors in the location of the knots are: $\mu_1 = (5.59 \pm 2.05) \times 10^{-4}$ with 95% confidence, and $\mu_2 = (-4.62 \pm 1.94) \times 10^{-4}$. Errors in the scale of 10^{-4} , in a support with a range $[0, 1]$, and 5% noise amplitude in the curve can be considered a highly satisfactory result.

5.2 Sensor Data

The algorithm was also applied to a set of real-world sensor data¹ emanating from the monitoring of heavy machinery. The original data set can be seen in Figure 4 (top). It has many local peaks and periods of little or no change, so the algorithm was used to detect discontinuities in the first derivative, in order to determine the peaks and phases. The peaks in the Taylor differences were used in combination with the peaks of the extrapolation error to determine the points of discontinuity. A peak in the Taylor differences means that the Taylor coefficients are significantly different at that interstitial point, compared to other interstitial points in the neighbourhood. However, if there is no peak in the extrapolation errors at the same location, then the peak found by the Taylor differences is deemed insignificant, since one polynomial could model both the left and right values and as such the peak isn't a discontinuity. Additionally, it can be seen in Figure 5 that both the extrapolation error and the combined error, as defined in Section 4, have peaks at the same locations, and as such the results they provide do not differ significantly.

¹For confidentiality reasons the data has been anonymized.

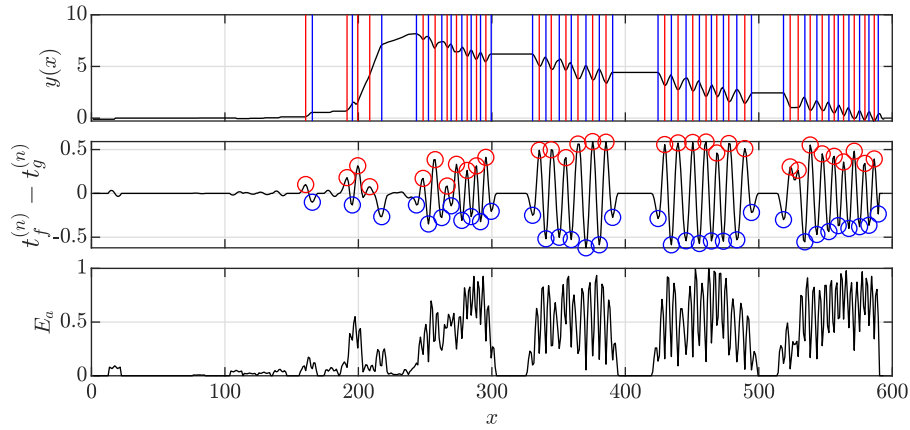


Figure 4: The top-most graph shows a function $y(x)$, together with the detected C^1 discontinuity points. The middle graph shows the difference in the Taylor polynomials $\Delta t_{fg}^{(n)}$ calculated at every interstitial point. The red and blue circles mark the relevant local maxima and minima of the difference respectively. According to this, the red and blue lines are drawn in the top-most graph. The bottom graph shows the approximation error evaluated at every interstitial point.

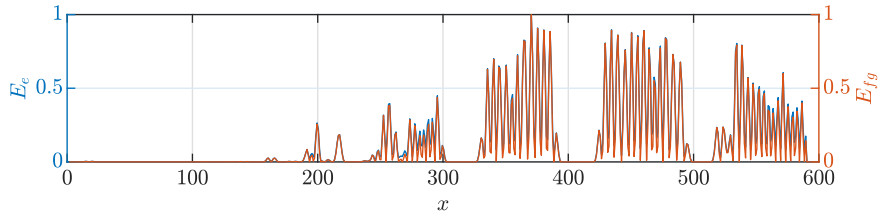


Figure 5: The two error functions, E_e and E_{fg} as defined in Section 4, for the example from Fig. 4. One can see that the location of the peaks doesn't change, and the two errors don't differ significantly.

6 Conclusion and Future Work

It may be concluded, from the results achieved, that the coupled constrained polynomial approximation yield a good method for the detection of C^n discontinuities in discrete observational data of continuous systems. Local peaks in the square of the difference of the Taylor polynomials provide a relative measure as a means of determining the locations of discontinuities.

Current investigations indicate that the method can be implemented directly as a convolutional operator, which will yield a computationally efficient solution. The use of discrete orthogonal polynomials [5, 10] is being tested as a means of improving the sensitivity of the results to numerical perturbations.

Acknowledgements.

This work was partially funded by:

1. The COMET program within the K2 Center “Integrated Computational Material, Process and Product Engineering (IC-MPPE)” (Project No 859480). This program is supported by the Austrian Federal Ministries for Transport, Innovation and Technology (BMVIT) and for Digital and Economic Affairs (BMDW), represented by the Austrian research funding association (FFG), and the federal states of Styria, Upper Austria and Tyrol.
2. The European Institute of Innovation and Technology (EIT), a body of the European Union which receives support from the European Union’s Horizon 2020 research and innovation programme. This was carried out under Framework Partnership Agreement No. 17031 (MaMMa - Maintained Mine & Machine).

The authors gratefully acknowledge this financial support.

Bibliography

- [1] Burden, R.L., Faires, J.D.: Numerical analysis. Pacific Grove, Calif Brooks/Cole, 9th edn. (2010)
- [2] Dung, V.T., Tjahjowidodo, T.: A direct method to solve optimal knots of b-spline curves: An application for non-uniform b-spline curves fitting. PLOS ONE **12**(3), 1–24 (03 2017). <https://doi.org/10.1371/journal.pone.0173857>, <https://doi.org/10.1371/journal.pone.0173857>
- [3] Gijbels, I., Goderniaux, A.C.: Data-driven discontinuity detection in derivatives of a regression function. Communications in Statistics - Theory and Methods **33**(4), 851–871 (2005). <https://doi.org/10.1081/STA-120028730>, <https://doi.org/10.1081/STA-120028730>
- [4] Klopfenstein, R.W.: Conditional least squares polynomial approximation. Mathematics of Computation **18**(88), 659–662 (1964), <http://www.jstor.org/stable/2002954>
- [5] O’Leary, P., Harker, M.: Discrete polynomial moments and Savitzky-Golay smoothing. International Journal of Computer and Information Engineering **4**(12), 1993 – 1997 (2010), <https://publications.waset.org/vol/48>
- [6] O’Leary, P., Harker, M., Zsombor-Murray, P.: Direct and least square fitting of coupled geometric objects for metric vision. Vision, Image and Signal Processing, IEE Proceedings - **152**, 687 – 694 (01 2006). <https://doi.org/10.1049/ip-vis:20045206>

- [7] O’Leary, P., Ritt, R., Harker, M.: Constrained polynomial approximation for inverse problems in engineering. In: Abdel Wahab, M. (ed.) *Proceedings of the 1st International Conference on Numerical Modelling in Engineering*. pp. 225–244. Springer Singapore, Singapore (2019)
- [8] Orvath, L., Kokoszka, P.: Change-point detection with non-parametric regression. *Statistics* **36**(1), 9–31 (2002). <https://doi.org/10.1080/02331880210930>, <https://doi.org/10.1080/02331880210930>
- [9] Owhadi, H.: Bayesian numerical homogenization. *Multiscale Modeling & Simulation* **13**(3), 812–828 (2015). <https://doi.org/10.1137/140974596>
- [10] Persson, P.O., Strang, G.: Smoothing by Savitzky-Golay and Legendre filters. In: *Mathematical Systems Theory in Biology, Communications, Computation, and Finance* (2003)
- [11] Qiu, P., Yandell, B.: Local polynomial jump-detection algorithm in nonparametric regression. *Technometrics* **40**(2), 141–152 (1998). <https://doi.org/10.1080/00401706.1998.10485196>, <https://doi.org/10.1080/00401706.1998.10485196>
- [12] Raissi, M., Perdikaris, P., Karniadakis, G.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* **378**, 686 – 707 (2019). <https://doi.org/https://doi.org/10.1016/j.jcp.2018.10.045>, <http://www.sciencedirect.com/science/article/pii/S0021999118307125>
- [13] Saxena, H., Aponte, O., McConky, K.T.: A hybrid machine learning model for forecasting a billing period’s peak electric load days. *International Journal of Forecasting* **35**(4), 1288 – 1303 (2019). <https://doi.org/https://doi.org/10.1016/j.ijforecast.2019.03.025>
- [14] Spokoiny, V.: Estimation of a function with discontinuities via local polynomial fit with an adaptive window choice. *Annals of Statistics* **26** (08 1998). <https://doi.org/10.1214/aos/1024691246>
- [15] Wahba, G.: *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics (1990). <https://doi.org/10.1137/1.9781611970128>, <https://epubs.siam.org/doi/abs/10.1137/1.9781611970128>
- [16] Weierstrass, K.: ber die analytische darstellbarkeit sogenannter willkrlicher functionen einer reellen vernderlichen. *Sitzungsberichte der Kniglich Preuischen Akademie der Wissenschaften zu Berlin*, 1885 (II) pp. 633–639, 789–805 (1885)
- [17] Yaman, B., Hosseini, S.A.H., Moeller, S., Ellermann, J., Uurbil, K., Akakaya, M.: Self-supervised physics-based deep learning mri reconstruction without fully-sampled data (2019)

A Convolutional Method for the Detection of Derivative Discontinuities

Dimitar Ninevski and Paul O’Leary
University of Leoben, A8700 Leoben, Austria
automation@unileoben.ac.at
<http://automatiom.unileoben.ac.at>

Abstract

This paper presents a convolutional method for the calculation of a measure for C-n discontinuities at interstitial points. This measure can be used to determine locations where the observational data is discontinuous in the n-th derivative. This is achieved by calculating the Taylor coefficients of the data from the left and right at each interstitial point. The calculation of the Taylor coefficients is constrained to ensure continuity of the model up to degree n-1, while yielding an estimate for the discontinuity of the n-th derivative. These constrains, together with a symmetric support length and a uniform spacing of the observational data, ensure a special block structure of the matrices in the least squares formulation of the minimization problem. This special structure enables an orthogonal residualization corresponding to a generalized Eckart-Young-Mirsky matrix approximation, which then yields a convolutional sequence required to calculate the difference of the n-th Taylor coefficients. Additionally, thanks to the symmetric support length and uniform spacing, it is seen that the convolutional sequences are symmetric or antisymmetric, which ensures a linear phase response. Finally, the results of applying the method to: sensor data emanating from the observation of a drilling process; and synthetic data from a cubic spline with known knots, are presented.

Keywords: Taylor coefficients, Convolution, Eckart-Young-Mirsky, Discontinuity, Local polynomials.

1 Introduction

The detection of discontinuities in observational data is a fundamental task; e.g., in the analysis of earth quake data [1]. In this case only jump discontinuities in the time series are considered. If a

more general class of discontinuities, i.e., C^n jump discontinuities, are considered then the derived methods become more generally applicable.

If a function $f(x)$ has continuous derivatives up to and including order $n - 1$, but the n^{th} derivative is discontinuous at the point x_a , meaning $f(x_a) \in C^{n-1} \setminus C^n$, this function is called C^n discontinuous at the point x_a .

The more general case of C^n discontinuities is also relevant when defining the locations of knots for B-spline functions when modelling observational data [2]. Furthermore, systems governed by differential equations, with some exceptions [3, 4], should not exhibit C^n discontinuities. Consequently, their detection in observational data is an important issue when examining anomalous behaviour. The issue is also important when defining features in data which are being evaluated in what is called *physics informed* [5] machine learning.

In [6] the authors presented a generalized method of detecting C^n discontinuities based on constrained local polynomial approximation. Although very general, it was numerically very intensive, since it involves the inversion of matrices for every interstitial point. Here the method is adapted to the special case more generally encountered in measurement and control:

1. It is assumed that the data is sampled at uniform intervals. This does not restrict the application in measurement and control; however, it leads to a uniform definition of the problem at every interstitial point.
2. To ensure linear phase response, symmetry around the interstitial point is enforced in terms of the support and local degrees of the polynomials.
3. A coefficient coupled, rather than a constrained, least squares approximation was introduced.

These conditions lead to a special structure of the matrices involved in the least squares approximation. This enables an orthogonal residualization of the design matrix corresponding to a generalized Eckart-Young-Mirsky matrix approximation. Solving in this manner yields a convolutional sequence s , which can be used to compute a measure for the C^n discontinuity. The value of the convolutional approach is seen in the high numerical efficiency with which they can be computed. First tests indicate an acceleration by a factor of $a \approx 300$ with respect to the solution presented in [6]. Furthermore, GUP systems offer a hardware parallel acceleration for the computation of convolutions.

2 Detecting C^n discontinuities

Discrete observations of sensor data are, by their very nature, discontinuous at every point. Consequently, it is necessary to introduce a *measure* for discontinuity. This is achieved by approximating the data to the left of an interstitial point x_a , by $f(x)$ and right by $g(x)$, with two continuous functions, in this case polynomials, see Figure 2.

Polynomials are the logical choice, since their coefficients relate directly to Taylor coefficients and secondly, the Weierstrass approximation theorem [7] states that: *any* function $f(x)$ can be approximated by a polynomial to an arbitrary accuracy ϵ given a polynomial of sufficiently high degree. That is, they are suitable for any function we might encounter locally in the data.

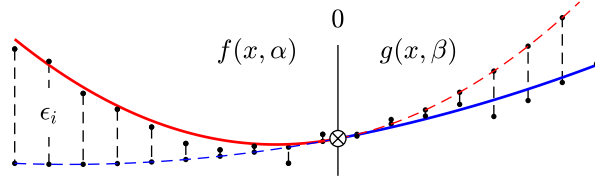


Figure 1: Principle of the computation at an interstitial point. The data to the left and right of the interstitial points are considered to be the functions $f(x)$ and $g(x)$; modelled by the polynomials $f(x, \alpha)$ and $g(x, \beta)$. The coupled approximation ensured that $f(x)$ and $g(x)$ are C^{n-1} continuous at the interstitial point.

The n term Taylor expansion of a function $f(x)$ around the point a is defined as:

$$f(x) = \sum_{k=0}^n \left\{ \frac{f^{(k)}(a)}{k!} \right\} (x-a)^k + R(x), \quad (2.102)$$

whereby, $R(x)$ is the residual error. Given a polynomial approximation for $f(x)$ at the location a , i.e., $f(x-a, \alpha)$,

$$f(x) \approx f(x-a, \alpha) = \sum_{k=0}^n \alpha_k (x-a)^k, \quad (2.103)$$

similarly for $g(x)$. A judicious selection of the interstitial point x_a as the origin for both the left and right approximations yields: $f(x, \alpha)$, $g(x, \beta)$. Furthermore, the Taylor series becomes a Maclaurin series, i.e.,

$$f(x) \approx \sum_{k=0}^n \left\{ \frac{f^{(k)}(0)}{k!} \right\} x^k. \quad (2.104)$$

When using a Taylor expansion to approximate a polynomial $f(x, \alpha)$, there is a close relationship between the coefficients of the polynomial α and the coefficients of the Taylor expansion. These relationships yield two estimates for the Taylor coefficients of the data at each interstitial point, one from the left and one from the right. Denoting $t_f^{(n)}$ as the n^{th} Taylor coefficient of $f(x, \alpha)$ and similarly $t_g^{(n)}$ from $g(x, \beta)$, one gets

$$t_f^{(n)} = \alpha_n \quad \text{and} \quad t_g^{(n)} = \beta_n. \quad (2.105)$$

In conjunction with the definition for C^n discontinuity, it is now possible to define a measure:

$$\Delta_t^{(n)} = t_f^{(n)} - t_g^{(n)}$$

given $t_f^{(k)} = t_g^{(k)}$ for all $k \in [0 \dots n-1]$. (2.106)

This measure considers jump discontinuities in the n -th derivative, asymptotic discontinuities are not considered. Furthermore, in more classical data modelling, C^n jump discontinuities form the basis for the locations of knots in B-Spline models of observational data [2].

3 Algebraic Formulation

The measure defined by Equation 2.106 implies that $f(x, \alpha)$ and $g(x, \beta)$ are both polynomial approximations of degree n ; whereby, their coefficients are coupled such that $\alpha_k = \beta_k$ for all $k \in [0 \dots n - 1]$. The support for the left and right approximations is defined as \mathbf{x}_L for $f(x, \alpha)$ and \mathbf{x}_R for $g(x, \beta)$; whereby they share the interstitial point as their common origin. Symmetric support lengths are selected in this paper to ensure either symmetric or antisymmetric convolutional sequences. This has been performed to obtain linear phase response. The cost function of this approximation task is

$$E = \|\mathbf{y}_L - \mathbf{V}_n(\mathbf{x}_L) \boldsymbol{\alpha}\|_2^2 + \|\mathbf{y}_R - \mathbf{V}_n(\mathbf{x}_R) \boldsymbol{\beta}\|_2^2$$

Consequently, the least squares minimization problem becomes

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} E &= \min_{\boldsymbol{\alpha}} \|\mathbf{y}_L - \mathbf{V}_n(\mathbf{x}_L) \boldsymbol{\alpha}\|_2^2 + \min_{\boldsymbol{\beta}} \|\mathbf{y}_R - \mathbf{V}_n(\mathbf{x}_R) \boldsymbol{\beta}\|_2^2 \\ \text{Given } \boldsymbol{\alpha}_k &= \boldsymbol{\beta}_k \quad \text{for all } k \in [0 \dots n - 1]. \end{aligned} \quad (2.107)$$

Since the constraint corresponds directly to equating the coefficients, the approximation can be implemented, by correct partitioning of the matrices, in the following manner:

$$\begin{bmatrix} x_L^n & \mathbf{0} & \mathbf{V}_{n-1}(\mathbf{x}_L) \\ \mathbf{0} & x_R^n & \mathbf{V}_{n-1}(\mathbf{x}_R) \end{bmatrix} \begin{bmatrix} \alpha_n \\ \beta_n \\ \alpha_{n-1} \\ \vdots \\ \alpha_0 \end{bmatrix} = \begin{bmatrix} \mathbf{y}_L \\ \mathbf{y}_R \end{bmatrix} \quad (2.108)$$

The structure of the matrices in this equation are very closely related to those encountered in coupled approximation of geometric objects [8]. The method proposed there, of orthogonal residualization, is utilized here to obtain an efficient solution.

Defining¹,

$$\begin{aligned} z_L &\triangleq \underbrace{\mathbf{x}_L \circ \mathbf{x}_L \dots \circ \mathbf{x}_L}_n, \quad z_R \triangleq \underbrace{\mathbf{x}_R \circ \mathbf{x}_R \dots \circ \mathbf{x}_R}_n \\ \mathbf{Z} &\triangleq \begin{bmatrix} z_L & \mathbf{0} \\ \mathbf{0} & z_R \end{bmatrix}, \quad \boldsymbol{\gamma} \triangleq \begin{bmatrix} \alpha_n \\ \beta_n \end{bmatrix}, \quad \boldsymbol{\delta} \triangleq \begin{bmatrix} \alpha_{n-1} \\ \vdots \\ \alpha_0 \end{bmatrix} \end{aligned}$$

¹The symbol \circ denotes the Hadamard product in the following equations.

$$\mathbf{V}_{n-1} \triangleq \begin{bmatrix} \mathbf{V}_{n-1}(\mathbf{x}_L) \\ \mathbf{V}_{n-1}(\mathbf{x}_R) \end{bmatrix} \quad \text{and} \quad \mathbf{y} \triangleq \begin{bmatrix} \mathbf{y}_L \\ \mathbf{y}_R \end{bmatrix}$$

yields a partitioned formulation of the least squares problem

$$\mathbf{Z} \boldsymbol{\gamma} + \mathbf{V}_{n-1} \boldsymbol{\delta} = \mathbf{y}. \quad (2.109)$$

4 Orthogonal Residualization

In the case being considered here, only the coefficients $\boldsymbol{\gamma}$ are required. Consequently, the common coefficients $\boldsymbol{\delta}$ can be eliminated from Equation 2.109. This is done by projecting \mathbf{y} onto the orthogonal complement of the subspace spanned by \mathbf{V}_{n-1} , as follows: the coefficients $\boldsymbol{\delta}$ are computed as follows

$$\boldsymbol{\delta} = \mathbf{V}_{n-1}^+ \mathbf{y}.$$

Substituting this into Equation 2.109 yields

$$\mathbf{Z} \boldsymbol{\gamma} + \mathbf{V}_{n-1} \mathbf{V}_{n-1}^+ \mathbf{y} = \mathbf{y}. \quad (2.110)$$

Defining $\mathbf{P}_{n-1} \triangleq \mathbf{V}_{n-1} \mathbf{V}_{n-1}^+$, which is the projection matrix associated with \mathbf{V}_{n-1} , and taking this to the right hand side gives

$$\mathbf{Z} \boldsymbol{\gamma} = \mathbf{y} - \mathbf{P}_{n-1} \mathbf{y} = \{\mathbf{I} - \mathbf{P}_{n-1}\} \mathbf{y}. \quad (2.111)$$

Clearly \mathbf{y} is being projected onto the orthogonal complement, denoted by $\mathbf{P}_{n-1}^\perp = \mathbf{I} - \mathbf{P}_{n-1}$, of the subspace spanned by \mathbf{V}_{n-1} , this corresponds to a generalized Eckart-Young-Mirsky matrix approximation [9]. Now the coefficient vector $\boldsymbol{\gamma}$ is calculated as

$$\boldsymbol{\gamma} = \mathbf{Z}^+ \mathbf{P}_{n-1}^\perp \mathbf{y}. \quad (2.112)$$

Finally,

$$\Delta_t^{(n)} = \alpha_n - \beta_n = [1, -1] \boldsymbol{\gamma} \quad (2.113)$$

$$= [1, -1] \mathbf{Z}^+ \mathbf{P}_{n-1}^\perp \mathbf{y}. \quad (2.114)$$

This shows that the convolutional sequence \mathbf{s} , required to calculate the difference of the n -th Taylor coefficients, is given by

$$\mathbf{s} = [1, -1] \mathbf{Z}^+ \mathbf{P}_{n-1}^\perp \mathbf{y}. \quad (2.115)$$

5 Example Convolutional Sequences

The convolution sequences required to detect C^2 , C^3 and C^4 discontinuities are shown in Figure 2. Each of these sequences has been computed with a support length $l_s = 100$ points. Note that for even or odd n , symmetric or antisymmetric convolutional sequences are obtained respectively. This ensures a linear phase response.

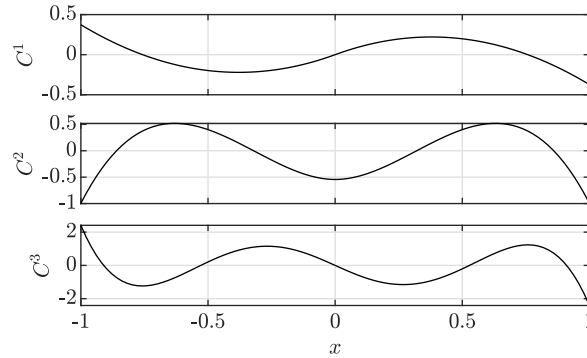


Figure 2: The convolution sequences s_n for C^n discontinuity detection, each with support length of $l_s = 100$ points: (top) C^2 -; (middle) C^3 - and (bottom) C^4 -, discontinuity. Note if n is odd then s_n is anti-symmetric and for even n , s_n is symmetric. This ensures that the convolution operators have linear phase response.

6 Test Applications

The proposed method has been tested on a set of sensor data obtained from the observation of a drilling process, see Figure 3; and on a set of synthetic data from a cubic spline with known knots, see Figure 4.

6.1 Sensor Data

For demonstration purposes, measures for C^1 and C^2 discontinuities were computed for all interstitial points in the sensor data, with a support length $l_s = 10$.

6.2 Cubic Spline Data

For the purpose of testing on a set of synthetic data, a cubic spline $f(x)$ was generated with double knots at $x = 2$ and $x = 3$, which ensured that $f(x)$ is C^2 discontinuous at these points. For better visualisation $f'(x)$ is also shown, with its C^1 discontinuities. In this case, measures for C^2 discontinuities were computed for all interstitial points with a support length $l_s = 10$.

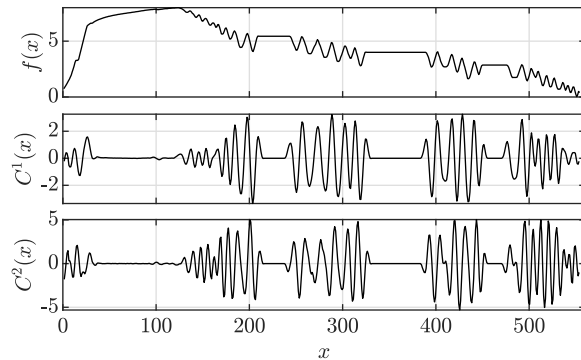


Figure 3: (Top) - sensor data obtained during the observation of a geo-drilling process. (Middle) - Measure for the C^1 discontinuity and (bottom) measure of the C^2 discontinuity. These were computed with a support length $l_s = 10$.

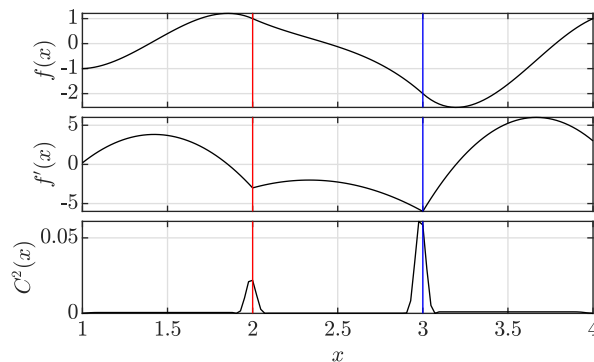


Figure 4: (Top) - A cubic spline $f(x)$ with double knots at $x = 2$ and $x = 3$. This ensures that $f(x)$ is C^2 discontinuous at those points. (Middle) - $f'(x)$, with its C^1 discontinuities at the double knots. (Bottom) - a measure of the C^2 discontinuity of $f(x)$. This was computed with a support length $l_s = 10$.

7 Other Physics Relevant Features

The approach presented here is easily applied to the convolutional computation of regularized derivatives. Furthermore, this leads to the derivation of a very efficient computation of local curvature $\kappa(x)$.

7.1 Convolutional Computation of Regularized Derivatives

In order to compute regularized derivatives using the approach presented here, one should initially define a local support for x_s , such that $-1 \leq x \leq 1$ with m_s samples; note, m_s odd leads to a colocative- and even to an interstitial-, computation respectively. The next step is computing a Vandermonde matrix V_d for x_s of degree d . For $m_s > d + 1$ regularized derivatives are the result.

Locally, the coefficients $\alpha = \mathbf{V}_d^+ y_s$ are obtained. In the case of the geometric polynomials, the derivatives $d^{(n)}$ at the origin of the local coordinate system are related to the coefficients as follows:

$$\mathbf{d}_n = \mathbf{M} \mathbf{V}_d^+ y_s, \quad (2.116)$$

whereby,

$$\mathbf{d}_n \triangleq \begin{bmatrix} d^{(n)} \\ \vdots \\ d^{(1)} \end{bmatrix} \quad \text{and} \quad \mathbf{M} \triangleq \begin{bmatrix} n! & 0 & 0 \\ & \ddots & \vdots \\ 0 & 1 & 0 \end{bmatrix}. \quad (2.117)$$

Consequently, the convolutional sequence $\mathbf{s}_n^{(k)}$ required to compute the k^{th} regularized derivative is,

$$\mathbf{s}^{(k)} = \mathbf{M} \mathbf{V}_d^+(d - k + 1, :). \quad (2.118)$$

That is, each row of $\mathbf{M} \mathbf{V}_d^+$ corresponds to a convolution sequence to compute a specific degree of derivative. This is effectively a generalization of the concept of Savitzky-Golay smoothing [10], to regularized derivatives. The method is demonstrated in Figure 5 where the first and second regularized derivatives are computed in this manner.

7.2 Curvature

Let $f(t) = (x(t), y(t))$ be a proper parametric representation of a twice differentiable planar curve. Here proper means that on the domain of the parametrization, the derivative $df(t)/dt$ is defined, differentiable and nowhere equal to the zero vector. Then the signed curvature is given by,

$$\kappa(t) = \frac{x' y'' - y' x''}{\{(x')^2 + (y')^2\}^{3/2}}. \quad (2.119)$$

In the case of uniformly sampled data, where x is the monotonic parameter, this simplifies to,

$$\kappa(x) = \frac{y''(x)}{\{1 + y'(x)^2\}^{3/2}}. \quad (2.120)$$

Given the numerically efficient method of computing the regularized derivatives, $y'(x)$ and $y''(x)$; computing $\kappa(x)$ becomes trivial, see Figure 5.

Acknowledgments

This work was partially funded by:

1. The COMET program within the K2 Center “Integrated Computational Material, Process and Product Engineering (IC-MPPE)” (Project No 859480). This program is supported by the Austrian Federal Ministries for Transport, Innovation and Technology (BMVIT) and for Digital and Economic Affairs (BMDW), represented by the Austrian research funding association (FFG), and the federal states of Styria, Upper Austria and Tyrol.

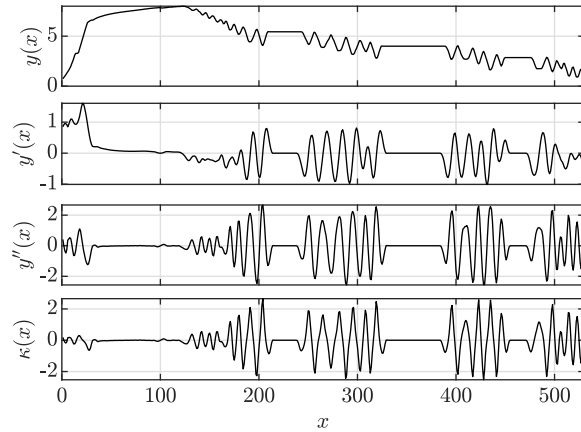


Figure 5: The convolutional computation of the first and second order derivatives as well as the curvature $\kappa(x)$ for the same data as shown in Figure 3. These computations are performed with a total support length of $l_s = 11$, i.e., a colocative result.

2. The European Institute of Innovation and Technology (EIT), a body of the European Union which receives support from the European Union’s Horizon 2020 research and innovation programme. This was carried out under Framework Partnership Agreement No. 17031 (MaMMa - Maintained Mine & Machine).

The authors gratefully acknowledge this financial support.

Bibliography

- [1] M. Roggero, “Discontinuity detection and removal from data time series,” in *Hotine-Marussi Symposium on Mathematical Geodesy. International Association of Geodesy Symposia*, C. M. S. F. Sneeuw N., Novák P., Ed. Heidelberg: Springer, Berlin, 2012, pp. 135–140.
- [2] G. Wahba, *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, 1990. [Online]. Available: <https://epubs.siam.org/doi/abs/10.1137/1.9781611970128>
- [3] O. Hájek, “Discontinuous differential equations, i,” *Journal of Differential Equations*, vol. 32, no. 2, pp. 149 – 170, 1979.
- [4] D. E. Stewart and M. Anitescu, “Optimal control of systems with discontinuous differential equations,” *Numerische Mathematik*, vol. 114, no. 4, pp. 653–695, Feb 2010. [Online]. Available: <https://doi.org/10.1007/s00211-009-0262-2>
- [5] M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial

- differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686 – 707, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999118307125>
- [6] D. Ninevski and P. O’Leary, “Detection of derivative discontinuities in observational data,” 2019, to be published.
- [7] K. Weierstrass, “Über die analytische darstellbarkeit sogenannter willkürlicher functionen einer reellen veränderlichen,” *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin, 1885 (II)*, pp. 633–639, 789–805, 1885.
- [8] P. O’Leary, M. Harker, and P. Zsombor-Murray, “Direct and least square fitting of coupled geometric objects for metric vision,” *IEE Proceedings - Vision, Image and Signal Processing*, vol. 152, no. 6, pp. 687–694, Dec 2005.
- [9] G. Golub, A. Hoffman, and G. Stewart, “A generalization of the Eckart-Young-Mirsky matrix approximation theorem,” *Linear Algebra and its Applications*, vol. 88, pp. 317–327, 1987.
- [10] A. Savitzky and M. Golay, “Smoothing and differentiation of data by simplified least squares procedures,” *Analytical Chemistry*, vol. 36 (8), p. 1627..1639, 1964.

A Computational Framework for Generalized Constrained Inverse Problems

Paul O’Leary and Dimitar Ninevski
University of Leoben, A8700 Leoben, Austria
automation@unileoben.ac.at
<http://automatiom.unileoben.ac.at>

Abstract

This paper generalizes and extends the theory of approximating measurement data with constrained basis functions. The new method is particularly well suited for modelling sensor data in cyber-physical systems, where the physics of the system being monitored needs to be embedded. The extension includes both co-located and interstitial constraints. The complete derivation of all required equations is presented in a matrix algebraic framework. The new approach enables the reconstruction of curves with a lower statistical uncertainty from fewer measurement points. The method is demonstrated here with examples from structural monitoring. A Monte Carlo simulation is presented where the new approach is compared with unconstrained approximation. The new approach has a significantly narrow band of uncertainty around the reconstructed curve. Furthermore, approximations are presented for real data acquired in a laboratory bending beam setup. This data is used to validate the claim that, with this method, reliable reconstructions can be obtained from a low number of measurement points.

Keywords: Constrained approximation, structural monitoring, bending beams, discrete orthogonal polynomials.

1 Introduction

A pertinent definition of CPS, is given by the IEEE [1] and ACM¹:

¹ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS) (iccps.acm.org)

‘A CPS is a system with a coupling of the cyber aspects of computing and communications with the physical aspects of dynamics and engineering that must abide by the laws of physics. This includes sensor networks, real-time and hybrid systems.’

Consequently, the approximations computed from the sensor data must obey the equations describing the physics of the system and the known constraints.

Recently, *physics informed data science* has become a focus of research, e.g., [2]; denoted by different names e.g., *physics informed* [3]; *hybrid-* [4]; or *physics based-learning* [5], etc. They embody the same basic idea of embedding physical principles into the data science algorithms. The goal is to ensure that the computed results are consistent with the laws of physics governing the system. These types of constrained inverse problems are very rarely addressed in literature on *mining sensor data*, see for example [6–8].

The advent of CPS is supporting the implementation of multi-sensor measurement systems; whereby the distributed sensor-data is synchronously sampled: yielding a vector of spatially sampled data at each sampling time point. An example of such a cyber-physical monitoring system is shown in Figure 1. The data now has both temporal and spatial aspects, which need to be correctly modelled. Temporal problems are often associated with a single sensor whose signal is sampled at uniform intervals, see for example [9], causal filtering [10] is an appropriate approach for such cases. In contrast spatial measurements, e.g. [11–15], are often at non-uniform intervals, due to the physical location of the sensors. Furthermore, the solution in the spatial domain is subject to the physical constraints of the system being observed. Spatial problems commonly lead to boundary value problems (BVP), as does a-causal filtering.

This paper proposes a computation framework for the definition and solution of generalized constrained inverse problems. Inverse problems associated with systems modelled by Sturm-Liouville [16] equations fall into this class of problems, making the methods widely applicable in physics. More specifically, there is a large class of constrained inverse problems associated with mechanical engineering and structure monitoring applications; one such example is used as a demonstration in this paper, see Figure 2.

The need for solutions at predetermined, but arbitrarily nodes² due to the placement of the sensors, precludes the use of the Chebyshev matrix approach, see Sezer [17] and others [18–21]. Collocation techniques [22] use piecewise polynomials functions to find a solution to a BVP; however, they do not provide a mechanism to deal with over-determined systems as encountered in inverse problems and data mining. A global basis function approach was published in the paper [11] and later extended in [15]. In those papers a finite difference approach was taken to computing discrete approximations. This approach works well for some specific types of constraint. However, finite differences lead to inaccurate estimates for derivatives in spatial problems, particularly when there is only a low number of measurement locations available. The main contributions of this paper

²Nodes define formally the points at which the basis functions are evaluated, for example the Fourier basis functions are discrete orthogonal polynomials evaluated on uniformity spaced points on the unit circle in the complex plane and the Gram polynomials are evaluated on points uniformly spaced on the real axis.

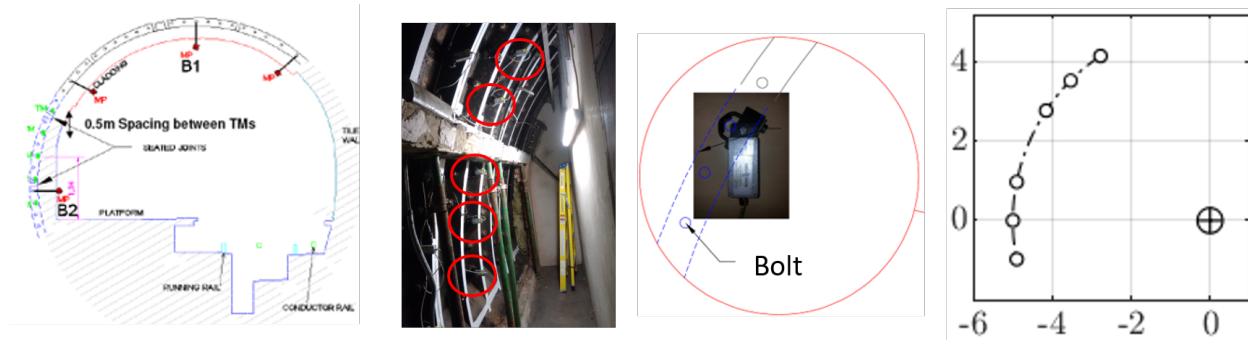


Figure 1: Example of a cyber-physical measurement system: a low number of spatially distributed sensors are used to monitor a tunnel structure, during construction in the vicinity. The figure shows from left to right: 1) Schematic of the tunnel profile; 2) Locations of the individual sensors on the tunnel supporting strut; 3) Concept for mounting the sensors of the strut and 4) a schematic reconstruction. The aim is to monitor the bending of the structural beam as a function of time. This involves solving the constrained inverse problem and interpolating to obtain the complete bending curve. In this example the constraints and sensors are not collocated, whereas the interpolation points possibly are; consequently, both interstitial and colocative methods are required.

are:

1. A new computation approach for the solution of inverse constrained problems; whereby the simultaneous use of both finite difference and pseudo-spectral constraints is permitted. This removes some limitations from the possible locations of the sensors.
2. The formulation of a pseudo-spectral approach to the derivatives of discrete orthogonal polynomials (DOP).
3. A clear and complete algebraic formulation of the problem, together with the derivation of the solution.
4. The formulation of the constrained approximation as a subspace problem, rather than in a Lagrange manner. This reduces the dimensionality of the problem; leading to a higher degree of regularization wrt. noise in the observational data. Additionally, the numerical computations are more stable.

2 Task at hand

A formal definition of the class of problems will make the versatility of the new approach visible. The task is to approximate a set of n measurement values \mathbf{y}_m , at the locations \mathbf{x}_m , by a smooth function $f(\boldsymbol{\alpha}, \mathbf{x})$, which fulfils a set of k generalized constraints. These constraints may be specific to a location \mathbf{x}_c or a relationship between different locations. The result of the computation

$f(\boldsymbol{\alpha}, \boldsymbol{x}_i)$ is required as a set of interpolative locations \boldsymbol{x}_i ; that is, the solution simultaneously implements both approximation and interpolation. This is, by definition, a constrained minimization problem and as such, is normally approached using Lagrange multipliers [23]. However, direct Lagrangian approaches lead to higher dimensionality in the solution space, since for each constraint the multiplier λ_i must be determined, in addition to the coefficients $\boldsymbol{\alpha}$.

The approach taken here is to start from a set of discrete orthogonal polynomials (DOP), which form a Hilbert space³ [24], as the columns of a matrix \boldsymbol{B} . Given the constraints, it is possible to compute a subspace of \boldsymbol{B} that fulfil the constraints. The constrained basis functions \boldsymbol{B}_c , also form a Hilbert space and correspond to a set of admissible functions that fulfil the constraints in a homogeneous manner. In this sub-space the problem is now an unconstrained approximation. This reduces the dimensionality of the coefficient space by the number of unique constraints; yielding a numerically more efficient solution.

3 Algebraic formulation

Given the locations \boldsymbol{x}_m , \boldsymbol{x}_c and \boldsymbol{x}_i , a unique sorted complete vector of locations \boldsymbol{x} is formed,

$$\boldsymbol{x} = \text{unique}([\boldsymbol{x}_c; \boldsymbol{x}_m; \boldsymbol{x}_i]). \quad (2.121)$$

This removes issues associated with co-locations. Note the vector \boldsymbol{x} will, in general, not be uniformly spaced.

A set of discrete orthogonal polynomials [25] are synthesized at the locations \boldsymbol{x} , up to degree d ; these form the columns of the matrix \boldsymbol{B} . This ensures that the functions are consistent across the measurement, constraint and interpolation locations. The property of DOP, i.e. $\boldsymbol{B}^T \boldsymbol{B} = \boldsymbol{I}$, ensures optimal conditioning of the matrix [26].

The elimination matrices \boldsymbol{E}_m and \boldsymbol{E}_i , are defined, such that:

$$\boldsymbol{x}_m = \boldsymbol{E}_m \boldsymbol{x}, \quad (2.122)$$

$$\boldsymbol{x}_i = \boldsymbol{E}_i \boldsymbol{x}. \quad (2.123)$$

These matrices permit the selection of the measurement and interpolation locations from the vector \boldsymbol{x} as required. They provide a clear formulation of the algebra; however, in code indexing is used since it is numerically more efficient, e.g., if i_m are the indices require to select \boldsymbol{x}_m , then $\boldsymbol{x}_m(i_m) = \boldsymbol{E}_m \boldsymbol{x}$.

The constrained minimization problems can be written as,

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{y}_m - \boldsymbol{E}_m \boldsymbol{B} \boldsymbol{\alpha}\|_2^2, \quad (2.124)$$

³The dimensions of the vectors and matrices vary according to the number of sensors and interpolation required. Consequently, in algebraic derivations it is established practice to formulate the matrices generically; without explicit definition of their dimensions.

given a set of k constraints each of the form,

$$\mathbf{c}_j^T \mathbf{B} \mathbf{M}_j \boldsymbol{\alpha} = w_j \quad j \in [1 \dots k]. \quad (2.125)$$

Each constraint definition forms a row vector and is later denoted by:

$$\mathbf{h}_j = \mathbf{c}_j^T \mathbf{B} \mathbf{M}_j. \quad (2.126)$$

The coefficients $\boldsymbol{\alpha}$ obtained from this approximation are used to compute the final result as follows

$$\mathbf{y}_i = \mathbf{E}_i \mathbf{B} \boldsymbol{\alpha}. \quad (2.127)$$

The definition (2.125) for the constraints is more general than past solutions [11, 15, 27]; since, it permits the simultaneous combination of finite difference and pseudo-spectral methods.

3.1 Pseudo-spectral derivative constraints

A generalized derivative constraint is defined here as:

$$y^{(\delta)}(x) = w. \quad (2.128)$$

This permits the constraining the δ^{th} derivative of the function $y(x)$ at the locations x to the value w . Multiple constrains of this form are permitted. This definition requires the computation of accurate derivatives of the basis functions at arbitrary locations within the support.

In the case at hand, there may be a low number of measurement points and constraints. As a result, finite differences may lead to inaccuracy in the computation of estimates for higher order derivatives. Pseudo-spectral techniques, on the other hand, permit the exact computation of the derivatives of the basis functions at arbitrary locations. Consequently, in this paper we propose the use of the pseudo-spectral approach to implement derivative constraints, while maintaining the finite difference approach where appropriate.

3.2 Derivatives of the basis

A discrete orthogonal polynomials \mathbf{B} can be computed from the three term relationship [25],

$$\mathbf{b}_n = \alpha \mathbf{x} \circ \mathbf{b}_{n-1} + \beta \mathbf{b}_{n-2}, \quad (2.129)$$

such that $\mathbf{B}^T \mathbf{B} = \mathbf{I}$, the symbol \circ denotes the Hadamard product of two vectors. The corresponding matrix of first derivatives $d\mathbf{B}$ is obtained using the chain rule,

$$\mathbf{b}'_n = \alpha \left(\mathbf{x}' \circ \mathbf{b}_{n-1} + \mathbf{x} \circ \mathbf{b}'_{n-1} \right) + \beta \mathbf{b}'_{n-2}. \quad (2.130)$$

The derivatives of polynomials are also polynomials; consequently, the matrix of derivatives $d\mathbf{B}$ can be regarded as linear combinations of the columns of \mathbf{B} . The matrix \mathbf{P} is determined such that $\mathbf{B}\mathbf{P} = d\mathbf{B}$; consequently,

$$\mathbf{P} = \mathbf{B}^T d\mathbf{B}. \quad (2.131)$$

Given \mathbf{P} any desired order of derivative k can be computed from the bases as,

$$\mathbf{B}^{(k)} = \mathbf{B}\mathbf{P}^k \quad (2.132)$$

whereby $\mathbf{B}^{(k)}$ denotes the k^{th} derivative of the columns of \mathbf{B} .

3.3 Relational constraints

The vector \mathbf{c}_j in Eqn. 2.125 also permits the definition of constraints on relationships between the solution at different locations. This is a common requirement when approximating data from systems governed by Sturm-Liouville equations. For example, if it is necessary to have the second derivative $y''(x)$ to have the same value at both ends of the support, the following values are required

$$\mathbf{c}_j^T = [1, 0, \dots, 0, -1], \quad \mathbf{M}_j = \mathbf{P}^2, \quad \text{and} \quad w_j = 0. \quad (2.133)$$

Substituting, these values into Eqn. 2.125, yields the required implementation of the constraint.

Integral constraints can also be implemented, if e.g.,

$$\mathbf{c}_j^T = [1/2, 1, \dots, 1, 1/2], \quad (2.134)$$

$\mathbf{M}_j = \mathbf{I}$ and w_j is assigned a value; then a constrain is placed on the Trapezoidal approximation to the integral over $y(x)$.

The simultaneous combination of: the rows of \mathbf{B} via \mathbf{c} and columns via \mathbf{M} enables the implementation of a very general class of constraints; a major extension over previous solutions.

3.4 The constrained basis functions

Each of the k constraint conditions, Eqn. 2.125, yields a row vector, \mathbf{h}_j ; these are vertically concatenated to form the matrix \mathbf{H} . Consequently, $\mathbf{H}\boldsymbol{\alpha} = \mathbf{w}$ defines the complete set of constraints. With this we obtain, an algebraic formulation of the constrained minimization problem.

$$\min_{\boldsymbol{\alpha}} \|\mathbf{y}_m - \mathbf{E}_m \mathbf{B} \boldsymbol{\alpha}\|_2^2 \quad (2.135)$$

$$\text{given} \quad \mathbf{H} \boldsymbol{\alpha} = \mathbf{w}. \quad (2.136)$$

Such constrained problems are most commonly approached with Lagrangian multipliers. This has the disadvantage of increasing the dimensionality of the minimization, since both the coefficients and the Lagrange multipliers must be determined.

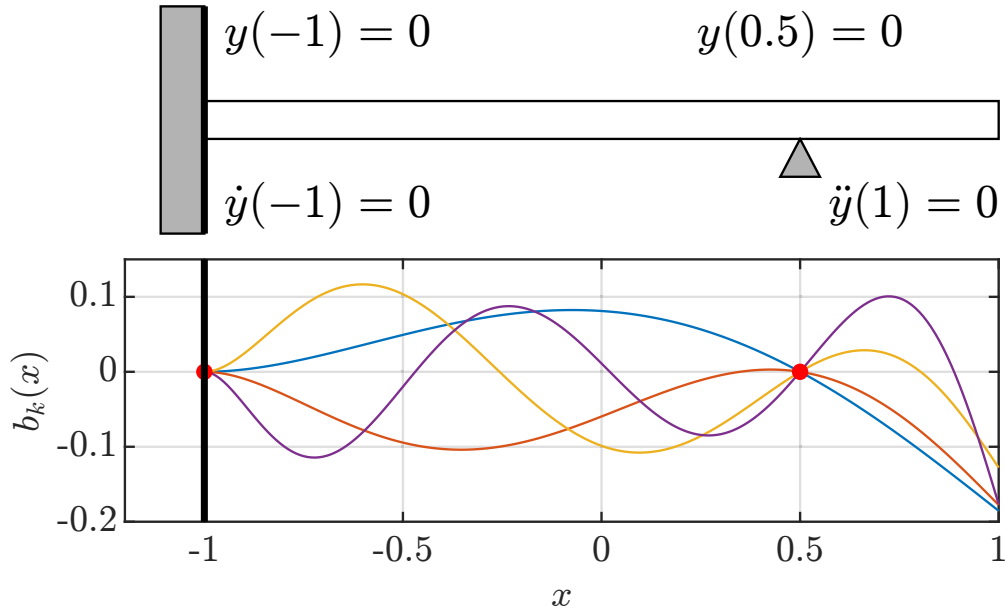


Figure 2: Top: Example of a cantilever with additional support: This is a generic example, since arbitrary constraints associated with a specific structure are possible. Bottom: The first four homogeneously constrained discrete polynomials from \mathbf{B}_c corresponding to the constraints defined above.

The alternative is to split this problem into a particular and a homogeneous portion. Starting from Eqn. 2.136 to obtain,

$$\alpha = \mathbf{H}^+ \mathbf{w} + \mathbf{N} \gamma, \quad (2.137)$$

whereby, \mathbf{N} is a unitary vector basis set for the null space of \mathbf{H} , and γ is the vector of coefficients for the remaining degrees of freedom. The coefficients α can be considered in the particular $\alpha_p = \mathbf{H}^+ \mathbf{w}$ and homogeneous $\alpha_h = \mathbf{N} \gamma$ portions. The corresponding portions of \mathbf{y} are,

$$\mathbf{y}_p = \mathbf{B} \alpha_p = \mathbf{B} \mathbf{H}^+ \mathbf{w}, \quad (2.138)$$

$$\mathbf{y}_h = \mathbf{B} \mathbf{N} \gamma. \quad (2.139)$$

The particular part of the solution \mathbf{y}_p can be computed in advance, since the values for the constraints are known a-priori. The homogeneous portion \mathbf{y}_h is a linear combination of homogeneously constrained basis functions $\mathbf{B}_c = \mathbf{B} \mathbf{N}$ such that $\mathbf{y}_h = \mathbf{B}_c \gamma$. The constrained basis functions \mathbf{B}_c are also orthonormal, since both \mathbf{B} and \mathbf{N} are orthonormal.

Residualizing the original measurement data \mathbf{y}_m wrt. the particular solutions yields, $\mathbf{y}_r = \mathbf{y}_m - \mathbf{E}_m \mathbf{y}_p$. Now the minimization can be formulated as an unconstrained problem, with the cost function $E(\gamma)$,

$$E(\gamma) = \|\mathbf{y}_r - \mathbf{E}_m \mathbf{B}_c \gamma\|_2^2. \quad (2.140)$$

Given an initial polynomial of degree d and k independent constraints, the subspace \mathbf{B}_c has the dimensionality $d_c = d - k + 1$, This is $2k$ lower dimensionality than the Lagrangian approach. The solution yields γ ,

$$\gamma = \{\mathbf{E}_m \mathbf{B}_c\}^+ \mathbf{y}_r. \quad (2.141)$$

Given γ we can calculate $\alpha_h = \mathbf{N} \gamma$. Finally, the solution \mathbf{y}_i is evaluated for the interpolative points \mathbf{x}_i ,

$$\mathbf{y}_i = \mathbf{E}_i \{\mathbf{B} \alpha_p + \mathbf{B}_c \gamma\}. \quad (2.142)$$

3.5 Covariance propagation

The continuous nature of the polynomials, together with the constraints lead to a positional dependence in the propagation of errors, i.e., an error at the input will propagate differently depending on its location. Assuming independent identically distributed (iid) noise at the input, with standard deviation σ_y and given Eqn. 2.139; the covariance propagation Λ_r for the value of $y(x)$ can be computed as,

$$\Lambda_r = \sigma_y^2 \mathbf{B}_c \mathbf{B}_c^T. \quad (2.143)$$

An estimate for the standard deviation σ_r at each point in the reconstruction can be computed as⁴,

$$\sigma_r = \sqrt{\text{diag}\{\Lambda_r\}}. \quad (2.144)$$

This can be considered to be a local variation in the confidence interval along the reconstruction curve, given iid noise at the input, see Fig. 4.

4 Numerical testing

An example problem, generic to mechanical engineering and structural monitoring, is defined in Figure 2. It has been selected since it has constraints on values as well as on first and second derivatives; furthermore, it is a BVP with an additional inner constraint. The four constraints defined in this problem are: two value constraints $y(-1) = 0$ and $y(0.5) = 0$; a constraint on the first derivative $y'(-1) = 0$ and the constraint of the second derivative $y''(1) = 0$. The corresponding constrained basis functions are also shown in Figure 2.

A Monte Carlo simulation was performed for the same generic example. There were $n = 1000$ synthetic data sets, each with i.i.d. Gaussian noise with $\sigma = 5e - 3$, yielding a signal to noise ratio $SNR = 1.91$. Both a constrained and unconstrained reconstructions were performed to permit a comparison. The reconstructions in terms of the values, first and second order derivatives are

⁴Similar computation can be performed for the propagation of uncertainty to the derivatives; however, these are not presented here due to the lack of available space.

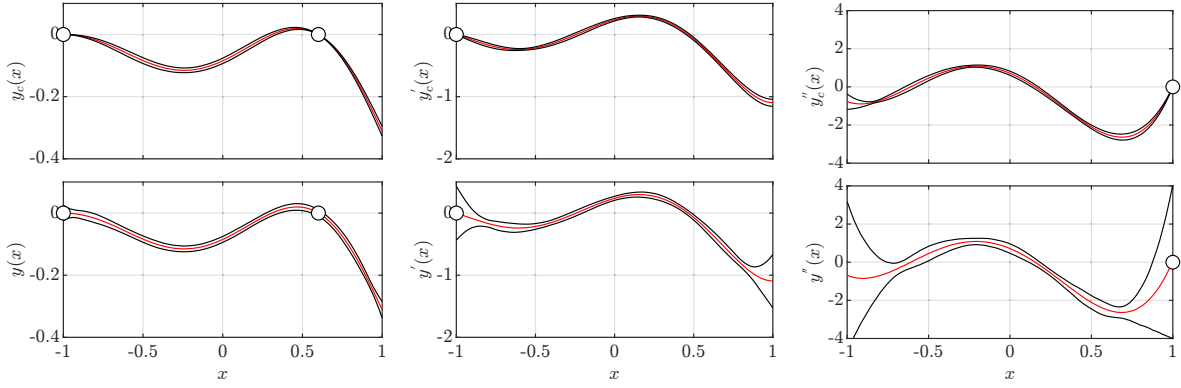


Figure 3: Results of the Monte Carlo test of the reconstruction algorithm for the cantilever shown in Figure 2. There were $n = 1000$ synthetic data sets, each with i.i.d. Gaussian noise with $\sigma = 5e - 3$, yielding a $SNR = 1.91$. The results of the reconstructions of the function $y(x)$, the first $y'(x)$ and second derivatives $y''(x)$ are organized from left to right. Top: result of the constrained approximation. Bottom: an unconstrained approximation. Red: Is the median value of the reconstructions; black: the upper and lower statistical bounds for outlier detection, i.e., $q_{75} + 1.5IQR$ and $q_{25} - 1.5IQR$. The circles indicate the constraint values in the respective domains, i.e., in the function values and its derivatives. The constrained reconstruction has zero error and no propagation of noise at the respective constraints.

shown in Figure 3. Additionally, the statistical upper b_u and lower b_l bounds,

$$b_u = q_{75} + 1.5IQR, \quad (2.145)$$

$$b_l = q_{25} - 1.5IQR. \quad (2.146)$$

for outlier detection were determined from the Monte Carlo simulation; these are also shown in Figure 3.

As can be seen in Figure 3 the proposed method has fulfilled the constraints in each of the three required domains. The median reconstruction shows that both the constrained and unconstrained reconstruction converge statistically to their expected values. However, the uncertainty of the reconstruction is significantly better for the constrained algorithm, see Table.2.1 where the 2-norm of the inter-quantile range (IQR) is given for the function, its first and second derivatives. In particular, for higher order derivatives the unconstrained approximation yields results with a very high uncertainty, see e.g. the case with the second derivative shown in Figure 3. This is particularly important for structural monitoring, since the bending of the structure is directly related to the second derivative.

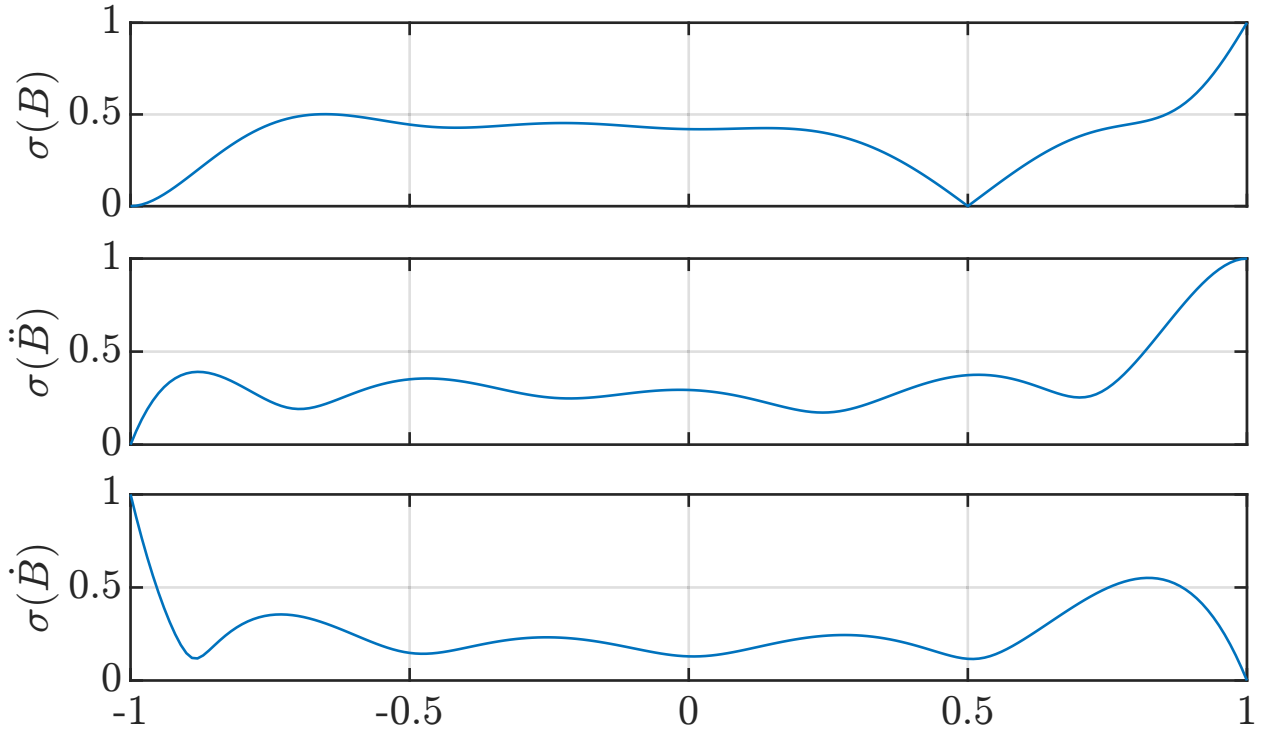


Figure 4: Relative local uncertainty along the curve for: $y(x)$ (top), $y'(x)$ (middle) and for $y''(x)$ (bottom), for iid noise at the input. These results are consistent with those obtained by the Monte-Carlo simulation, see Fig. 3.

5 Experimental verification

A laboratory test bench was assembled to enable the experimental verification of the new method. It yields measurement data for the deflection of a real beam, as shown in Figure 5. The setup was designed to correspond as closely as possible to the synthetic case presented in Figure 2. It consisted of a bending beam, clamped at one end and with an additional support and a light sectioning camera mounted on a linear drive. This permits the scanning and measurement of the actual beam bending profile.

In the first test, all the experimental measurement data were used during the reconstruction, see Figure 6. As expected the method fulfils the constraints and delivers the interpolated result at a regular set of locations, whereas the measurements and constraints are irregularly spaced. To verify the claim that a reliable reconstruction can be achieved with a low number of measurements, a subset $n = 10$ points were selected from the original measurement data. The points have been selected equal spaced along the support x with an additional random perturbation. This subset was then used to reconstruct the bending curve, see Figure 7. The constrained approximation has reliably reconstructed the deflection curve, while maintaining the desired constraints.

Table 2.1: Comparison of the constrained and unconstrained reconstruction wrt. 2-norm of the IQR of the uncertainty obtained from the Monte Carlo simulation.

	Constrained	Unconstrained
$y(x)$	0.037	0.058
$\frac{dy(x)}{dx}$	0.141	0.572
$\frac{d^2y(x)}{dx^2}$	0.633	5.873

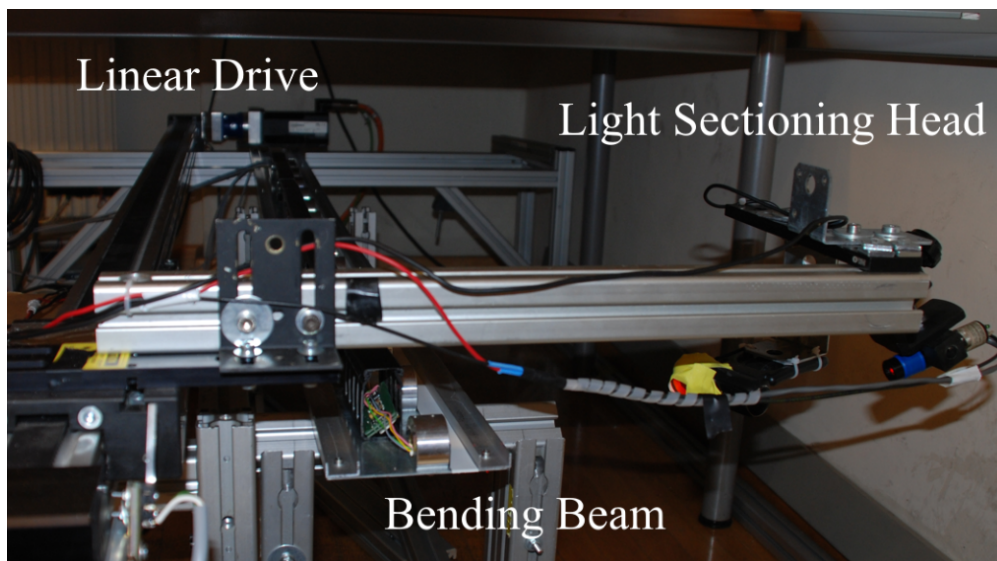


Figure 5: Laboratory test setup used to verify the method, consisting of a bending beam with an additional support. A light sectioning head on a linear drive to measure the actual deflection of the beam.

6 Conclusions

This paper has extended the theory of constrained approximations. The new method is based on matrix algebra to identify embedded subspaces that fulfill constraints. Homogeneous approximations in these subspaces are of lower dimensionality than the corresponding Lagrangian approach. This yields numerically more efficient computations and more stable results. The new formulation has functioned well with both synthetic and real measurement data. It provides a more general framework for defining and performing constrained approximations.

Future plans to extend this research include:

1. In the case of linear differential operators, iid noise is mapped to the eigenfunctions inversely proportional to their corresponding eigenvalues. This will enable a weighted approximation

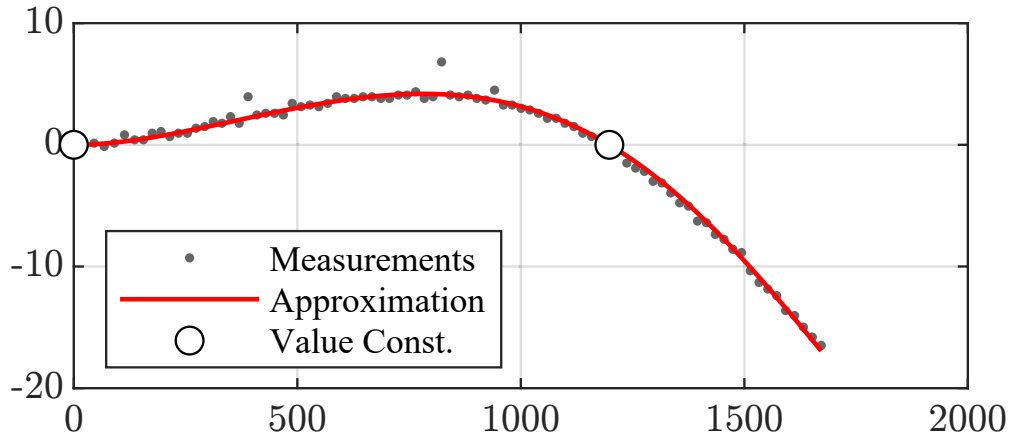


Figure 6: Reconstruction of the bending curve form the experimental data. Note: there are some measurements with larger errors, this is to be expected and desirable when testing the stability of the measurement algorithm.

based on the Rayleigh quotient.

2. To extend the current spectra regularization, with a Tikhonov regularization term relating to the second derivative. This is simple to implement, since, defining $\mathbf{B}_c^{(2)} \triangleq \mathbf{B}_c \mathbf{P}^2$ then the regularization term can be calculated as $\mathcal{L} = \lambda (\mathbf{B}_c^{(2)} \boldsymbol{\alpha})^T \mathbf{B}_c^{(2)} \boldsymbol{\alpha}$. This will penalize curves with high rates of curvature.
3. It is common to monitor ground subsidence using inclinometers, see Machan and Bennett [28] for a good overview. Consequently, it would be valuable to implement constrained curve reconstruction from measured gradients; given the basis functions and derivatives this can be achieved.

Acknowledgment

The authors gratefully acknowledge the financial support under the scope of the COMET program within the K2 Center “Integrated Computational Material, Process and Product Engineering (IC-MPPE)” (Project No 859480). This program is supported by the Austrian Federal Ministries for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) and for Digital and Economic Affairs (BMDW), represented by the Austrian research funding association (FFG), and the federal states of Styria, Upper Austria and Tyrol.

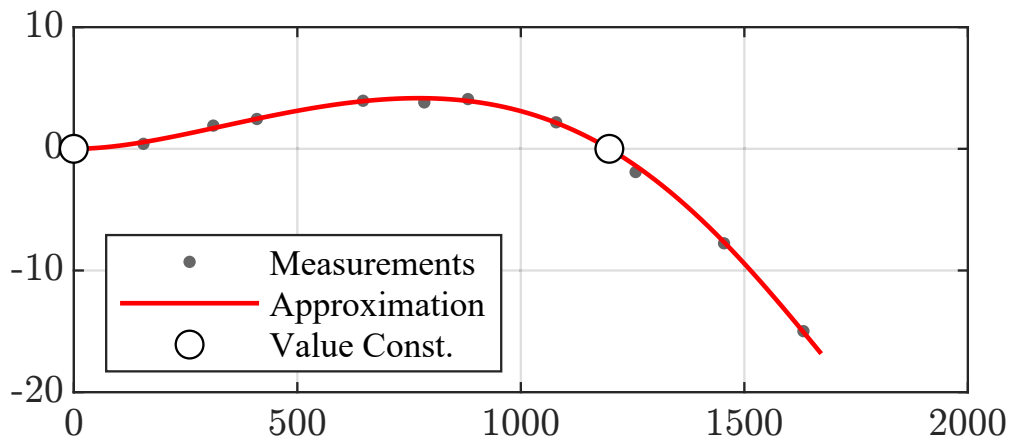


Figure 7: Reconstruction of the bending curve using a subset $n = 10$ of the measurements selected from the data shown in Figure 6. Note the irregular spacing of the measurement data. This result verifies the stability of the algorithm with a low number of observations.

Bibliography

- [1] R. Baheti and H. Gill, “Cyber-physical systems,” *The Impact of Control Technology*, pp. 161–166, 2011.
- [2] H. Owhadi, “Bayesian numerical homogenization,” *Multiscale Modeling & Simulation*, vol. 13, no. 3, pp. 812–828, 2015.
- [3] M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686 – 707, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999118307125>
- [4] H. Saxena, O. Aponte, and K. T. McConky, “A hybrid machine learning model for forecasting a billing period’s peak electric load days,” *International Journal of Forecasting*, vol. 35, no. 4, pp. 1288 – 1303, 2019.
- [5] B. Yaman, S. A. H. Hosseini, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya, “Self-supervised physics-based deep learning mri reconstruction without fully-sampled data,” 2019.
- [6] M. Last, A. Kandel, and H. Bunke, *Data Mining in Time Series Databases*, ser. Series in machine perception and artificial intelligence. World Scientific, 2004. [Online]. Available: <http://books.google.at/books?id=f38wqKjyBm4C>
- [7] C. C. Aggarwal, Ed., *Managing and Mining Sensor Data*. Springer, 2013.

- [8] P. Esling and C. Agon, “Time-series data mining,” *ACM Comput. Surv.*, vol. 45, no. 1, pp. 12:1–12:34, Dec. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2379776.2379788>
- [9] F. Müller, G. Rath, T. Lucyshyn, C. Kukla, M. Burgsteiner, and C. Holzer, “Presentation of a novel sensor based on acoustic emission in injection molding,” *Journal of Applied Polymer Science*, 2012.
- [10] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Prentice-hall Englewood Cliffs, 1999.
- [11] P. O’Leary and M. Harker, “Direct discrete variational curve reconstruction from derivatives and its application to track subsidence measurements,” in *International Instrumentation and Measurement Technology Conference (I2MTC 2011)*, 5 2011.
- [12] X. Hou, X. Yang, and Q. Huang, “Using inclinometers to measure bridge deflection,” *Journal of Bridge Engineering*, vol. 10, no. 5, pp. 564–569, 2005. [Online]. Available: <http://link.aip.org/link/?QBE/10/564/1>
- [13] J. Van Cranenbroeck, “Continuous beam deflection monitoring using precise inclinometers,” in *FIG Working Week 2007*, Hong Kong, SAR, 13..17 May, 2007.
- [14] O. Burdet and L. Zanella, “Automatic Monitoring of the Riddes Bridges using Electronic Inclinometers,” in *IABMAS, First International Conference on Bridge Maintenance, Safety and Management*, 2002.
- [15] P. O’Leary and M. Harker, “A framework for the evaluation of inclinometer data in the measurement of structures,” *IEEE T. Instrumentation and Measurement*, vol. 61, no. 5, pp. 1237–1251, 2012.
- [16] J. Pryce and L. Pryce, *Numerical Solution of Sturm-Liouville Problems*, ser. Monographs on numerical analysis. Clarendon Press, 1993. [Online]. Available: <https://books.google.at/books?id=bTDvAAAAMAAJ>
- [17] M. Sezer and M. Kaynak, “Chebyshev polynomial solutions of linear differential equations,” *International Journal of Mathematical Education in Science and Technology*, vol. 27, no. 4, pp. 607–618, 1996. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/0020739960270414>
- [18] B. Welfert. (18 Jul 2004 (Updated 21 Jul 2004)) Pseudospectral differentiation on an arbitrary grid. Matlab File Exchange. [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/5515>

- [19] J. A. Weideman and S. C. Reddy, “A matlab differentiation matrix suite,” *ACM Trans. Math. Softw.*, vol. 26, no. 4, pp. 465–519, Dec. 2000. [Online]. Available: <http://doi.acm.org/10.1145/365723.365727>
- [20] T. A. Driscoll, F. Bornemann, and L. N. Trefethen, “The chebop system for automatic solution of differential equations,” *BIT*, vol. 48, pp. 701–723, 2008.
- [21] N. Jewell. (29 Mar 2013 (Updated 01 Apr 2013)) Collocation-based spectral-element toolbox. Matlab File Exchange. [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/41011>
- [22] U. Ascher, J. Christiansen, and R. D. Russell, “Collocation software for boundary value ODE’s,” *ACM Transactions on Mathematical Software*, vol. 7, no. 2, pp. 209–222, Jun. 1981. [Online]. Available: <http://doi.acm.org/10.1145/355945.355950>
- [23] J. Douglas, “Solution of the inverse problem of the calculus of variations,” *Transactions of the American Mathematical Society*, vol. 50, no. 1, pp. 71–128, 1941. [Online]. Available: <http://www.jstor.org/stable/1989912>
- [24] P. Fuhrmann, *Linear Systems and Operators in Hilbert Space*, ser. Advanced book program. McGraw-Hill International Book Company, 1981. [Online]. Available: <https://books.google.at/books?id=cvNQAAAAMAAJ>
- [25] H. Stahl and V. Totik, *General Orthogonal Polynomials*, ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1992.
- [26] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.
- [27] R. W. Klopfenstein, “Conditional least squares polynomial approximation,” *Mathematics of Computation*, vol. 18, no. 88, pp. 659–662, 1964. [Online]. Available: <http://www.jstor.org/stable/2002954>
- [28] G. Machan and V. G. Bennett, “Use of inclinometers for geotechnical instrumentation on transportation projects,” *Transportation Research E-Circular*, vol. E-C129, 2008. [Online]. Available: <http://worldcat.org/issn/00978515>

Chapter 3

Optimal Control

Optimal control deals with problems of controlling a dynamical system that in an optimal manner, by optimizing some objective function. In this thesis, the systems dealt with can be described by a linear system of ordinary differential equations (ODEs), thus only such cases will be introduced in the text. This chapter addresses the mathematical foundations necessary for solving optimal control problems for linear systems numerically.

1 Collocative and Interstitial Numerical Derivatives

In chapter 2 it was described how to numerically approximate the derivatives of a discrete dataset. The result was an equation of the form

$$\begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \\ \vdots \\ y'_{n-1} \\ y'_n \end{bmatrix} \approx \frac{1}{2h} \begin{bmatrix} -3 & 4 & -1 & 0 & \dots & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 0 & 1 \\ 0 & \dots & 0 & 1 & -4 & 3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix}. \quad (3.1)$$

Note that in this case, the approximation is performed using polynomials of degree 2 and the locations of the derivative approximations coincides with the locations of the available data points. So, for n available datapoints y_1, \dots, y_n , one gets n derivative values y'_1, \dots, y'_n . In some applications however, calculating the derivatives between the sampling points is also beneficial. So if one has n data points, one would need to calculate $n - 1$ interstitial derivatives, since there are $n - 1$ gaps between the n points.

1.1 Interstitial Derivatives

In the case at hand, the data is assumed to be uniformly spaced and the interstitial derivatives are always located in the middle between two data points. As in the case considered in 2.2.1, assume there are three equally spaced data points $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ such that $x_1 = a, x_2 = a + h$ and $x_3 = a + 2h$. The interpolating quadratic polynomial is denoted by $p(x)$. The interstitial derivatives of interest are located at the points $x_a = a + \frac{h}{2}$ and $x_b = a + \frac{3h}{2}$. Consequently

$$y'_a = p'(x_a) = -\frac{1}{h}y_1 + \frac{1}{h}y_2 \quad (3.2)$$

$$y'_b = p'(x_b) = -\frac{1}{h}y_2 + \frac{1}{h}y_3 \quad (3.3)$$

hence one gets the following matrix equation

$$\begin{bmatrix} y'_a \\ y'_b \end{bmatrix} = \frac{1}{h} \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}. \quad (3.4)$$

Using higher order polynomials and increasing the number of points can provide more accurate approximations of the derivatives.

2 Singular Value Decomposition

The Singular Value Decomposition (SVD) is a decomposition of a matrix A such that

$$A = USV^T \quad (3.5)$$

where the matrices U and V are orthonormal, meaning

$$\begin{aligned} U^T U &= U U^T = I \\ V^T V &= V V^T = I \end{aligned} \quad (3.6)$$

and the matrix S is diagonal, i.e.,

$$S = \begin{pmatrix} \sigma_1 & & & \mathbf{0} \\ & \sigma_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \sigma_n \end{pmatrix}. \quad (3.7)$$

The diagonal elements, $\sigma_k, 1 \leq k \leq n$ are called the *singular values* and are ordered in decreasing order

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0. \quad (3.8)$$

The matrix S can be written as

$$S = \begin{bmatrix} S_r & 0 \\ 0 & \Delta \end{bmatrix} \quad (3.9)$$

where Δ is a diagonal matrix containing the singular values which are 0, or very small when calculated numerically. This leads to a partitioning of U and V , which looks like

$$U = \begin{bmatrix} U_r & \tilde{U} \end{bmatrix}, \quad V = \begin{bmatrix} V_r & \tilde{V} \end{bmatrix} \quad (3.10)$$

so one can write

$$A = USV^T = \begin{bmatrix} U_r & \tilde{U} \end{bmatrix} \begin{bmatrix} S_r & 0 \\ 0 & \Delta \end{bmatrix} \begin{bmatrix} V_r^T \\ \tilde{V}^T \end{bmatrix} = U_r S_r V_r^T + \tilde{U} \Delta \tilde{V}^T. \quad (3.11)$$

Since Δ is practically a zero matrix, the matrix A can be approximated as

$$A \approx U_r S_r V_r^T \quad (3.12)$$

2.1 Range and Rank of A

The partitioning of the matrix in such a way can provide useful properties of the matrix A . As an example, one can take the range of A .

$$\underbrace{\mathbf{y}}_{\text{vector in range of } A} = A \underbrace{\mathbf{x}}_{\text{coefficients}} \quad (3.13)$$

$$\mathbf{y} = USV^T \mathbf{x} = U_r S_r V_r^T \mathbf{x} + \tilde{U} \Delta \tilde{V}^T \mathbf{x} \approx U_r \underbrace{S_r V_r^T \mathbf{x}}_z = U_r \mathbf{z}. \quad (3.14)$$

This shows that A and U_r have the same range. On the other hand, the columns of U_r are orthonormal, so

$$\text{rank}\{A\} = \text{rank}\{U_r\} = r. \quad (3.15)$$

2.2 Nullspace of A

Let \mathbf{x} be a vector in the nullspace of A , meaning

$$A\mathbf{x} = \mathbf{0}. \quad (3.16)$$

It follows that

$$\mathbf{0} = USV^T \mathbf{x} = U_r S_r V_r^T \mathbf{x} + \tilde{U} \Delta \tilde{V}^T \mathbf{x} \approx U_r S_r V_r^T \mathbf{x}. \quad (3.17)$$

Let's assume that

$$\mathbf{x} = \tilde{V} \mathbf{z}. \quad (3.18)$$

One now gets

$$\mathbf{0} = \mathbf{U}_r \mathbf{S}_r \mathbf{V}_r^T \mathbf{x} = \mathbf{U}_r \mathbf{S}_r \underbrace{\mathbf{V}_r^T \tilde{\mathbf{V}}}_{\mathbf{0}} \mathbf{z} \quad (3.19)$$

which means that the above equality is true for all vectors \mathbf{z} , so that

$$\text{nullity}\{\mathbf{A}\} = \text{rank}\{\tilde{\mathbf{V}}\} = n - r. \quad (3.20)$$

3 SVD and the Least Squares Problem

In 2.1.2 it was shown that, when minimizing the following cost function

$$C(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\| \quad (3.21)$$

using vertical distance least squares, the solution is

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}. \quad (3.22)$$

If $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$, one gets

$$C(\mathbf{x}) = \|\mathbf{U}\mathbf{S}\mathbf{V}^T \mathbf{x} - \mathbf{b}\| \quad (3.23)$$

Now, note that in the case when \mathbf{U} is orthonormal,

$$\|\mathbf{U}^T \mathbf{r}\|_2^2 = (\mathbf{U}^T \mathbf{r})^T (\mathbf{U}^T \mathbf{r}) = \mathbf{r}^T \underbrace{\mathbf{U}\mathbf{U}^T}_{\mathbf{I}} \mathbf{r} = \|\mathbf{r}\|_2^2. \quad (3.24)$$

This means that

$$C(\mathbf{x}) = \|\mathbf{U}^T (\mathbf{U}\mathbf{S}\mathbf{V}^T \mathbf{x} - \mathbf{b})\| = \|\mathbf{U}^T \mathbf{U}\mathbf{S}\mathbf{V}^T \mathbf{x} - \mathbf{U}^T \mathbf{b}\| = \|\mathbf{S}\mathbf{V}^T \mathbf{x} - \mathbf{U}^T \mathbf{b}\| \quad (3.25)$$

Substituting

$$\mathbf{V}^T \mathbf{x} = \mathbf{z} \quad (3.26)$$

would mean that

$$\mathbf{x} = \mathbf{V}\mathbf{z}, \quad (3.27)$$

thus one obtains

$$\begin{aligned} C(\mathbf{z}) = \|\mathbf{S}\mathbf{z} - \mathbf{U}^T \mathbf{b}\|_2^2 &= \left\| \begin{bmatrix} \mathbf{S}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{\Delta} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} - \begin{bmatrix} \mathbf{U}_r^T \\ \tilde{\mathbf{U}}^T \end{bmatrix} \mathbf{b} \right\|_2^2 = \left\| \begin{bmatrix} \mathbf{S}_r \boldsymbol{\alpha} \\ \mathbf{\Delta} \boldsymbol{\beta} \end{bmatrix} - \begin{bmatrix} \mathbf{U}_r^T \\ \tilde{\mathbf{U}}^T \end{bmatrix} \mathbf{b} \right\|_2^2 \\ &= \left\| \begin{bmatrix} \mathbf{S}_r \boldsymbol{\alpha} - \mathbf{U}_r^T \mathbf{b} \\ \mathbf{\Delta} \boldsymbol{\beta} - \tilde{\mathbf{U}}^T \mathbf{b} \end{bmatrix} \right\|_2^2 = \|\mathbf{S}_r \boldsymbol{\alpha} - \mathbf{U}_r^T \mathbf{b}\|_2^2 + \|\mathbf{\Delta} \boldsymbol{\beta} - \tilde{\mathbf{U}}^T \mathbf{b}\|_2^2, \end{aligned} \quad (3.28)$$

so the cost function can be written as

$$C(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \|\mathbf{S}_r \boldsymbol{\alpha} - \mathbf{U}_r^T \mathbf{b}\|_2^2. \quad (3.29)$$

The solution to this is

$$\boldsymbol{\alpha} = (\mathbf{S}_r^T \mathbf{S}_r)^{-1} \mathbf{S}_r^T \mathbf{U}_r^T \mathbf{b} = \mathbf{S}_r^{-1} \mathbf{U}_r^T \mathbf{b} \quad (3.30)$$

Note that $\boldsymbol{\beta}$ will not change anything in the cost function, since $\Delta \boldsymbol{\beta} \approx \mathbf{0}$. Once \mathbf{z} is calculated, \mathbf{x} can be computed as

$$\mathbf{x} = \mathbf{V} \mathbf{z} = \begin{bmatrix} \mathbf{V}_r & \tilde{\mathbf{V}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \mathbf{V}_r \boldsymbol{\alpha} + \tilde{\mathbf{V}} \boldsymbol{\beta} = \underbrace{\mathbf{V}_r \mathbf{S}_r^{-1} \mathbf{U}_r^T \mathbf{b}}_{\text{minimizes } C} + \underbrace{\tilde{\mathbf{V}} \boldsymbol{\beta}}_{\text{doesn't influence } C}. \quad (3.31)$$

When looking at the solution in equation 3.22, one sees that this required \mathbf{A} to be full rank. If \mathbf{A} isn't full rank, the inverse of $\mathbf{A}^T \mathbf{A}$ wouldn't exist and then the solution is given by equation 3.31. The matrix

$$\mathbf{A}^+ = \mathbf{V}_r \mathbf{S}_r^{-1} \mathbf{U}_r^T \quad (3.32)$$

is called the Moore-Penrose¹ pseudo inverse. Additionally, since the columns of the matrix $\tilde{\mathbf{V}}$ form a basis for the nullspace of \mathbf{A} , $\boldsymbol{\beta}$ is a vector of parameters in order to find all solutions. If a unique solution is necessary, some constraints will be needed.

4 Numerical Solutions to Linear Systems of ODEs

A linear system of ordinary differential equations (ODEs) has the form

$$\begin{aligned} y_1'(x) &= a_{11}y_1(x) + a_{12}y_2(x) + \dots + a_{1p}y_p(x) + b_1u(x) \\ y_2'(x) &= a_{21}y_1(x) + a_{22}y_2(x) + \dots + a_{2p}y_p(x) + b_2u(x) \\ &\vdots \\ y_p'(x) &= a_{p1}y_1(x) + a_{p2}y_2(x) + \dots + a_{pp}y_p(x) + b_pu(x) \end{aligned} \quad (3.33)$$

or, it can be written compactly in matrix form as

$$\mathbf{y}' = \mathbf{A} \mathbf{y} + \mathbf{u} b. \quad (3.34)$$

Systems of ODEs are very often used in engineering. In control theory, the functions $y_i(x)$ represent the states of the system, such as position, velocity, and the function $u(x)$ represents an input to the system. Systems of this form can be solved numerically using the derivative matrix methods discussed in 2.1.2.

The first step is to discretize the system on some interval $[a, b]$ wrt. the variable x , namely $a = x_0 < x_1 < x_2 < \dots < x_n = b$. The transpose of equation 3.34 can be written as

$$\mathbf{y}'^T = \mathbf{y}^T \mathbf{A}^T + \mathbf{u} b^T. \quad (3.35)$$

¹This is one way of calculating it.

Discretizing the vector of functions \mathbf{y} one gets

$$\mathbf{Y} \triangleq \begin{bmatrix} \mathbf{y}_1^T \\ \mathbf{y}_2^T \\ \vdots \\ \mathbf{y}_p^T \end{bmatrix} = \begin{bmatrix} y_1(x_1) & y_1(x_2) & \dots & y_1(x_n) \\ y_2(x_1) & y_2(x_2) & \dots & y_2(x_n) \\ \vdots & \vdots & \ddots & \vdots \\ y_p(x_1) & y_p(x_2) & \dots & y_p(x_n) \end{bmatrix}. \quad (3.36)$$

Using a derivative matrix \mathbf{D} , one can write

$$\mathbf{y}'_i = \mathbf{D}\mathbf{y}_i, \quad \text{for } 1 \leq i \leq p \quad (3.37)$$

and discretizing the vector of derivatives \mathbf{y}' one gets

$$\mathbf{Y}' \triangleq \begin{bmatrix} \mathbf{y}'_1^T \\ \mathbf{y}'_2^T \\ \vdots \\ \mathbf{y}'_p^T \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1^T \mathbf{D}^T \\ \mathbf{y}_2^T \mathbf{D}^T \\ \vdots \\ \mathbf{y}_p^T \mathbf{D}^T \end{bmatrix} = \mathbf{Y} \mathbf{D}^T. \quad (3.38)$$

The last function to discretize is $u(x)$ which is done as

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}. \quad (3.39)$$

Thus equation 3.35 can be discretized as

$$\mathbf{Y}'^T = \mathbf{Y}^T \mathbf{A}^T + \mathbf{u} \mathbf{b}^T \quad (3.40)$$

and using equation 3.38 one obtains

$$\mathbf{D} \mathbf{Y}^T - \mathbf{Y}^T \mathbf{A}^T - \mathbf{u} \mathbf{b}^T = \mathbf{0}. \quad (3.41)$$

By vectorizing the left hand side of the last equation one obtains

$$\text{vec}(\mathbf{D} \mathbf{Y}^T) - \text{vec}(\mathbf{Y}^T \mathbf{A}^T) - \text{vec}(\mathbf{u} \mathbf{b}^T) \quad (3.42)$$

and using the equality [6]

$$\text{vec}(\mathbf{A} \mathbf{B} \mathbf{C}) = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B}) \quad (3.43)$$

one finally gets

$$(\mathbf{I} \otimes \mathbf{D} - \mathbf{A} \otimes \mathbf{I}) \text{vec}(\mathbf{Y}^T) = \text{vec}(\mathbf{u} \mathbf{b}^T). \quad (3.44)$$

Note that the equation 3.44 represents an ordinary system of linear equations which can be solved using any standard method for systems of equations, such as SVD.

5 Calculus of Variations

Definition 1. A *functional* is a real valued function on a vector space V , usually of functions. In other words, it is a correspondence which assigns real numbers to each function (or curve) in a given class.

Calculus of variations is a part of mathematics dealing with optimizing functionals. Normal calculus looks for points which maximize or minimize certain functions, whereas calculus of variations looks for *curves* which optimize certain cost functions. A typical example of a functional is the length of a curve

$$J(y(x)) = \int_a^b \sqrt{1 + y'(x)^2} dx. \quad (3.45)$$

So the functional J assigns to each curve $y(x)$ its length. In order to find the function which minimizes a given functional, the Euler-Lagrange equations need to be introduced. What follows will be two Lemmas, necessary for the derivation of the Euler-Lagrange equations. They will be given without proof and for more details and proofs, the reader is referred to [7].

Lemma 1. If $\alpha(x)$ is continuous on the interval $[a, b]$ and if

$$\int_a^b \alpha(x) h'(x) dx = 0 \quad (3.46)$$

for every continuously differentiable function $h(x)$ for which $h(a) = h(b) = 0$, then $\alpha(x)$ is a constant, i.e. $\alpha(x) = c$.

Lemma 2. If $\alpha(x)$ and $\beta(x)$ are continuous on the interval $[a, b]$ and if

$$\int_a^b [\alpha(x)h(x) + \beta(x)h'(x)] dx = 0 \quad (3.47)$$

for every continuously differentiable function $h(x)$ for which $h(a) = h(b) = 0$, then $\alpha(x) = \beta'(x)$.
 $\alpha(x) = c_1x + c_0$.

5.1 Euler-Lagrange Equations

The most fundamental problem in calculus of variations is the minimization of the following functional

$$J = \int_a^b F(x, y, y') dx \quad (3.48)$$

subject to the boundary conditions

$$y(a) = A, y(b) = B. \quad (3.49)$$

The following theorem is one of the most fundamental theorems of calculus of variations.

Theorem 3. Let J be a functional like in equation 3.48, defined for all continuously differentiable functions satisfying the boundary conditions defined in equation 3.49. Then a necessary condition for J to have an optimum for a given function $y(x)$ is that $y(x)$ satisfy the *Euler-Lagrange* equation

$$F_y - \frac{d}{dx}F_{y'} = 0. \quad (3.50)$$

Proof. Consider a small increment $h(x)$ is added to the function $y(x)$, such that $h(a) = h(b) = 0$. Then the function $y(x) + h(x)$ also satisfies the boundary conditions. Consider now the following difference

$$\Delta J = J(y(x) + h(x)) - J(y(x)) = \int_a^b F(x, y + h, y' + h') dx - \int_a^b F(x, y, y') dx. \quad (3.51)$$

Using Taylor's theorem, one gets

$$\Delta J = \int_a^b [F_y(x, y, y')h + F_{y'}(x, y, y')h'] dx + \dots \quad (3.52)$$

where the dots represent the higher order terms in h and h' . It is easy to check, that a necessary (but not sufficient) condition that $y(x)$ is an optimum for the functional J is that

$$\int_a^b [F_y(x, y, y')h + F_{y'}(x, y, y')h'] dx = 0. \quad (3.53)$$

In other words,

$$\int_a^b [F_y h + F_{y'} h'] dx = 0 \quad (3.54)$$

for all h with the property $h(a) = h(b) = 0$ and according to Lemma 2, it follows that

$$F_y - \frac{d}{dx}F_{y'} = 0. \quad (3.55)$$

□

Thus, in order to minimize a functional of the form from equation 3.48, one has to solve the Euler-Lagrange equation 3.50.

5.2 Calculus of Variations in Optimal Control

The state space representation of a physical system has the form

$$\mathbf{y}'(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{B}\mathbf{u}(t) \quad (3.56)$$

where \mathbf{y} is the vector of states of the system, $\mathbf{u}(t)$ is the control variable vector, or input vector of the system, and \mathbf{A} and \mathbf{B} are matrices of coefficients. The optimal control problem is to determine

the input which would bring a system from one state to another in an optimal manner. Hence, calculus of variations methods are particularly useful when solving such problems. A simple cost function in optimal control is

$$C(\mathbf{u}(t)) = \int_{t_0}^{t_f} \mathbf{u}^T \mathbf{u}(t) dt. \quad (3.57)$$

However, it is worth noting that if any control is applied to the system, the changes in the state of the system will be governed by the system of ODEs defining the system. In other words, the final solution should satisfy the defining equation of the system. Thus, the optimization problem becomes a constrained optimization problem

$$\min_{\mathbf{u}} \int_{t_0}^{t_f} \mathbf{u}^T \mathbf{u}(t) dt \quad \text{given the constraints} \quad \mathbf{y}'(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{B}\mathbf{u}(t). \quad (3.58)$$

This can be solved using Lagrange multipliers. Thus, the following functional is formulated

$$J(\mathbf{y}(t), \mathbf{y}'(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)) = \frac{1}{2} \int_{t_0}^{t_f} \mathbf{u}^T \mathbf{u}(t) dt - \int_{t_0}^{t_f} \boldsymbol{\lambda}^T(t) (\mathbf{y}'(t) - \mathbf{A}\mathbf{y}(t) - \mathbf{B}\mathbf{u}(t)) dt \quad (3.59)$$

where the $\frac{1}{2}$ is added for algebraic simplicity later on in the calculations and doesn't affect the end result. The integrand of the functional has the form

$$F(\mathbf{y}(t), \mathbf{y}'(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)) = \frac{1}{2} \mathbf{u}^T \mathbf{u}(t) - \boldsymbol{\lambda}^T(t) (\mathbf{y}'(t) - \mathbf{A}\mathbf{y}(t) - \mathbf{B}\mathbf{u}(t)). \quad (3.60)$$

The Euler-Lagrange equations for this functional are

$$\begin{aligned} F_y - \frac{d}{dt} F_{y'} &= 0 \\ F_u - \frac{d}{dt} F_{u'} &= 0 \\ F_{\lambda} - \frac{d}{dt} F_{\lambda'} &= 0. \end{aligned} \quad (3.61)$$

Calculating the partial derivatives, one gets

$$F_y = \mathbf{A}^T \boldsymbol{\lambda}(t), \quad F_{y'} = -\boldsymbol{\lambda}(t) \quad (3.62)$$

$$F_u = \mathbf{u}(t) + \mathbf{B}^T \boldsymbol{\lambda}(t), \quad F_{u'} = 0 \quad (3.63)$$

$$F_{\lambda} = -\mathbf{y}'(t) + \mathbf{A}\mathbf{y}(t) + \mathbf{B}\mathbf{u}(t), \quad F_{\lambda'} = 0 \quad (3.64)$$

and thus

$$\mathbf{A}^T \boldsymbol{\lambda}(t) + \boldsymbol{\lambda}'(t) = 0 \quad (3.65)$$

$$\mathbf{u}(t) + \mathbf{B}^T \boldsymbol{\lambda}(t) = 0 \quad (3.66)$$

$$\mathbf{y}'(t) - \mathbf{A}\mathbf{y}(t) - \mathbf{B}\mathbf{u}(t) = 0. \quad (3.67)$$

Substituting $\mathbf{u}(t) = -\mathbf{B}^T \boldsymbol{\lambda}(t)$ from the second equation into the other two, one obtains a system of differential equations,

$$\begin{aligned}\mathbf{y}'(t) &= \mathbf{A}\mathbf{y}(t) - \mathbf{B}\mathbf{B}^T \boldsymbol{\lambda}(t) \\ \boldsymbol{\lambda}'(t) &= -\mathbf{A}^T \boldsymbol{\lambda}(t)\end{aligned}\tag{3.68}$$

which can now be solved using the techniques described earlier in the chapter.

Bibliography

- [1] A. Eisinberg, G. Franzé, and N. Salerno, “Rectangular vandermonde matrices on chebyshev nodes,” *Linear Algebra and its Applications*, vol. 338, no. 1, pp. 27–36, 2001. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S002437950100355X>
- [2] H. Wertz, “On the numerical inversion of a recurrent problem: The vandermonde matrix,” *IEEE Transactions on Automatic Control*, vol. 10, no. 4, pp. 492–492, 1965.
- [3] J. Baik, T. Kriecherbauer, K. D.-R. McLaughlin, and P. D. Miller, *Discrete Orthogonal Polynomials. (AM-164): Asymptotics and Applications (AM-164): Asymptotics and Applications (AM-164)*. Princeton University Press, 2007. [Online]. Available: <https://doi.org/10.1515/9781400837137>
- [4] G. Szegő, S. G. Szego, and A. M. Society, *Orthogonal Polynomials*, ser. American Math. Soc: Colloquium publ. American Mathematical Society, 1939. [Online]. Available: <https://books.google.at/books?id=ZOhmnsXlcY0C>
- [5] M. Harker, *Fractional Differential Equations: Numerical Methods for Applications*, ser. Studies in Systems, Decision and Control. Springer Cham.
- [6] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- [7] I. Gelfand and S. Fomin, *Calculus of Variations*, ser. Dover Books on Mathematics. Dover Publications, 2012. [Online]. Available: <https://books.google.at/books?id=CeC7AQAQBAJ>
- [8] G. H. Golub and V. Pereyra, “The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate,” *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 413–432, 1973.
- [9] G. Golub and V. Pereyra, “Separable nonlinear least squares: the variable projection method and its applications,” *Inverse Problems*, vol. 19, pp. R1–R26(1), 01 2003.

A Novel Method for Solving an Optimal Control Problem for a Numerically Stiff Independent Metering System

Goran Stojanoski¹, Dimitar Ninevski¹, Gerhard Rath¹ and
Matthew Harker²

¹University of Leoben, A8700 Leoben, Austria
{goran.stojanoski, dimitar.ninevski, gerhard.rath}@unileoben.ac.at
<http://automatiom.unileoben.ac.at>

²Faculty of Engineering and Applied Science,
Ontario Tech University, Ontario, Canada
matthew.harker@ontariotechu.ca

Abstract

This paper describes a new approach for solving an optimal control problem for a numerically stiff system. The objective is to move a load from its initial states to its final states in an optimal way. This was achieved by means of Lagrange multipliers. The resulting Euler-Lagrange equations lead to a system of differential equations, which was solved using a mass matrix method and discretized interstitial derivatives to obtain a numerically stable solution. In addition, the performance of the method was compared with both LQR and PID controllers. As a result, it can be seen for the example that is given, that the proposed method has an improved energy efficiency. The new method always attains the desired boundary values exactly with high precision.

1 Introduction

Independent metering valves today offer more flexible strategies for handling high inertia loads [1]. They can control the flow and the pressure simultaneously and accurately [2]. Increasing the pressure of the hydraulic system on one side will increase the overall mechanical stiffness of the system

which will lead to less oscillations [3]. This is very valuable for systems with high inertia load. Such machines include the tunnel boring machines, whose simplified system is described in Fig. 1. However, servo valves usually react much faster than the natural frequency of the load system. This leads to numerically stiff systems [4], [5], [6]. The solution of these systems cannot be computed with conventional solvers [7]. Furthermore, Rath in [8] shows that the exponential matrix method cannot be used to solve numerically stiff systems.

Several approaches are used for finding the optimal control of such systems. In [9] an optimization based on the Hamiltonian is used to find the optimal flows for a hydraulic system. Even though the system's linear friction parameter is set to zero, the system achieves the desired values without oscillations. Dupree in [10] uses the implicit learning capabilities of the RISE control to solve the Hamilton-Jacobi-Bellman (HJB) equation. In [11] an improvement in the control of different hydraulic systems is achieved using the Lagrangian force method. Pourebrahim in [12] uses a linear quadratic regulator (LQR) for trajectory control of a clutch actuated by a servo valve mechanism. In [13] the LQR control is used to design a digital autopilot for a reusable launch vehicle. Optimal non-linear feedback control based on the state-dependent Riccati equation (SDRE) is proposed in [14]. Snuegucki, in [15] an optimal control method based on mixed integer quadratic programming in a model predictive control framework is presented. In [16] an optimization scheme based on discretization of the distribution of the parameters into a fixed number of points and finding their optimal locations by methods of constrained non-linear programming is proposed.

This paper describes a new matrix based approach for solving the optimal control problem for a numerically stiff system, using the Euler-Lagrange equations and discretized interstitial derivatives. Additionally, the mass matrix method [17] is used to reduce the numerical stiffness of the system. The new method is then simulated for a position-controlled hydraulic system and compared with both LQR and PID controllers. The results show that the new method precisely attains the set target values for the position and pressure of the system. In addition, the Euler-Lagrange method shows improved energy-efficient performance compared to an LQR [18] and PID [19] controller in the example discussed in this paper.

2 System Model

2.1 Mechanical System

The mechanical system consists of a rotating load that performs a cutting operation. In this case, Fig. 1 shows a simplified model of the mechanical system in which the cylinder exerts a force on a large mass moving in a positive or negative direction. The position of the mass in Fig. 1 is described as y_1 . Furthermore, b is the viscous friction coefficient term and P_A and P_B represent the pressures on both piston and rod side respectively. Following this the equation of motion one obtains is:

$$m\ddot{y}_1 = P_A A_A - P_B A_B - b\dot{y}_1. \quad (3.69)$$

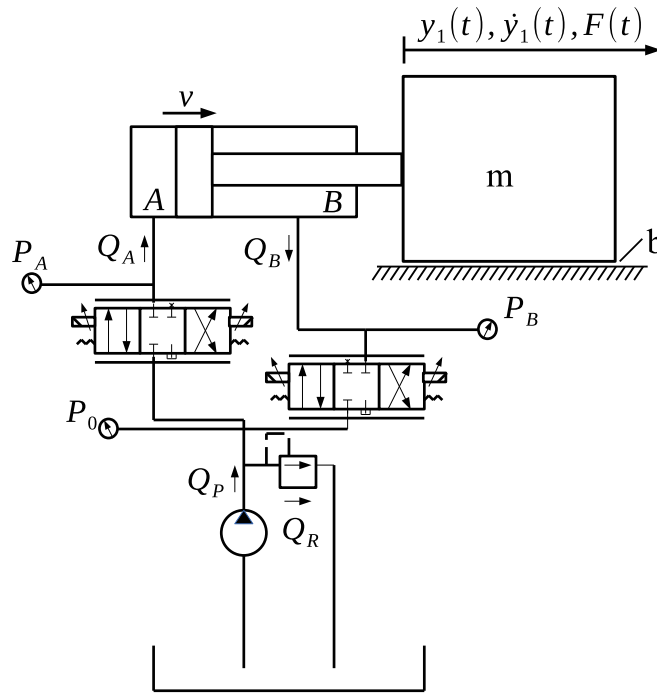


Figure 1: A hydraulic system with two independent metering valves, including a flow controller on the actuating (A) and a pressure controller on the back side (B). The load (m) of the system is position controlled. Auxiliary components, like pressure compensators and load sensing are not shown in this figure.

2.2 Hydraulic System

The hydraulic system consists of several parts: a hydraulic pump, two independent metering valves and a hydraulic actuator. In such systems all of the components can be the source of friction. The oil flow through the valve orifice, the movement of the load and the steel structure of the system can cause damping. However the hydraulic cylinder has highly non linear friction behaviour. A complete mathematical model of the friction includes the Stribeck and stiction effect as well as viscous and Coulomb friction [20]. In our case the assumption is made that the greatest contribution comes from the rotating load that performs the cutting operation. Therefore in (3.100) the friction is assumed to be viscous with coefficient b .

2.3 System dynamics

The position of the mass in Fig. 1 is controlled by a position controller. The position controller then actuates the independent metering valves, which have the embedded flow and pressure controllers, accordingly. The valves offer different operating modes for different load scenarios [2]. In our case shown in Fig. 1, the load will be passive for all directions as gravity does not affect the movement

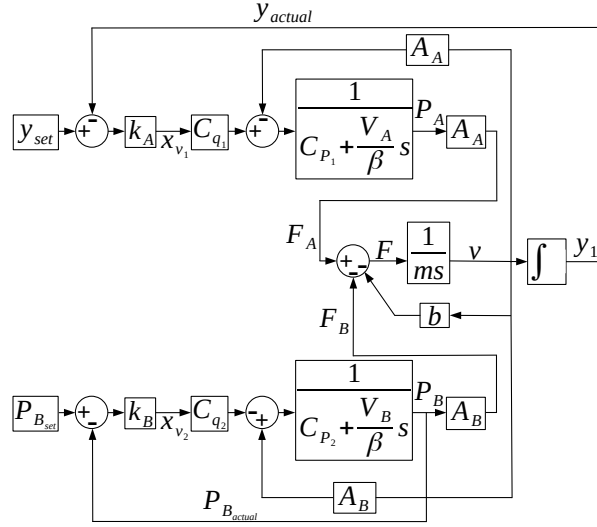


Figure 2: Block diagram for the system presented in Fig. 1. The system is controlled on both sides, where y_{set} and $P_{B_{set}}$ are the desired position and pressure values. Here k_A and k_B are the proportional parts of the PID controllers for both sides respectively.

of the system. For this scenario, the piston side (A) of the system is always flow controlled and the rod side (B) is pressure controlled. The hydraulic system was linearized using the method of small perturbations [21]. The linearized equations for the valve flows are a function of the spool positions of the valves and the pressures

$$Q_A = C_{q1}x_{v1} - C_{p1}P_A, \quad Q_B = C_{q2}x_{v2} + C_{p2}P_B. \quad (3.70)$$

In (3.101), x_{v1}, x_{v2} are spool positions of the valves and $C_{q1}, C_{q2}, C_{p1}, C_{p2}$ are the linearized terms for the valve flows. If (3.101) is now combined with the equations for the flow through the cylinder [22] one obtains

$$P_A = \frac{C_{q1}x_{v1} - A_A v}{\frac{V_A}{\beta}s + C_{p1}}, \quad P_B = \frac{A_B v - C_{q2}x_{v2}}{\frac{V_B}{\beta}s + C_{p2}}, \quad (3.71)$$

where V_A and V_B are the volumes and A_A and A_B are the areas of the piston and rod side of the cylinder respectively. Additionally, β is the bulk modulus of the oil. In Fig. 2 the system is position controlled on the piston side with a set value of y_{set} . In practice, the independent metering valves shown in Fig. 1 react much faster when compared to the natural frequency of the mechanical system. For this reason no flow controller was implemented on the piston side in Fig. 2. The position is controlled with a proportional P-controller with coefficient k_A . On the rod side a pressure controller was implemented with set value of $P_{B_{set}}$. The pressure is controlled with a proportional P-controller with a coefficient k_B .

As a result, the state space form for the LTI system is

$$\mathbf{\Pi} \dot{\mathbf{x}}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t) \quad (3.72)$$

where $\mathbf{\Pi}$ is the mass matrix [17]. Furthermore,

$$\mathbf{\Pi} = \begin{bmatrix} \frac{1}{\beta} & 0 & 0 & 0 \\ 0 & \frac{1}{\beta} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & m \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} P_A \\ P_B \\ y_1 \\ y_2 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} y_{set} \\ P_{Bset} \end{bmatrix}, \quad (3.73)$$

$$\mathbf{A} = \begin{bmatrix} \frac{-C_{p1}}{V_A} & 0 & \frac{-k_A C_{q1}}{V_A} & \frac{-A_A}{V_A} \\ 0 & \frac{-(k_B C_{q2} + C_{p2})}{V_B} & 0 & \frac{-A_B}{V_B} \\ 0 & 0 & 0 & 1 \\ \frac{A_A}{m} & \frac{-A_B}{m} & 0 & \frac{-b}{m} \end{bmatrix}, \quad (3.74)$$

$$\mathbf{B} = \begin{bmatrix} \frac{k_A C_{q1}}{V_A} & 0 \\ 0 & \frac{-k_B C_{q2}}{V_B} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \quad (3.75)$$

Note that the value of the parameter β , the bulk modulus, appears only in the mass matrix. On the other hand only a portion of the value of m , the mass, was stored in the mass matrix in order to have a numerically stable solution. This was done to improve the computational stability of the solving algorithm. When all parameters are on the right hand side and no mass matrix is used, the coefficients on the right hand side become too large and the computation too unstable to obtain a solution. The initial and final conditions of the system are defined as

$$\begin{aligned} \mathbf{I}_C &= [P_A(t_0) \quad P_B(t_0) \quad y_1(t_0) \quad y_2(t_0)]^T \\ \mathbf{B}_C &= [P_A(t_f) \quad P_B(t_f) \quad y_1(t_f) \quad y_2(t_f)]^T \end{aligned} \quad (3.76)$$

where t_0 and t_f are the start and final value of the time vector. The values in both \mathbf{I}_C and \mathbf{B}_C in (3.76) represent the real physical values of the system start and end values.

3 Solution of the problem

The task at hand is to bring the mass in Fig. 1 from its initial state $\mathbf{x}(t_0)$ to a final state $\mathbf{x}(t_f)$ within the desired time t_f in an optimal manner. This is an optimal control problem, which can be solved by minimizing the norm of the control variable defined as,

$$\int_{t_0}^{t_f} \mathbf{u}^T(t) \mathbf{u}(t) dt \quad (3.77)$$

Minimizing this integral, one has to consider that the system must satisfy the set of differential equations. This leads to a constrained optimization problem which can be solved using methods

from the Calculus of Variations, namely the method of Lagrange multipliers. Following this, the functional which needs to be minimized will have the following form:

$$J(\mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)) = \frac{1}{2} \int_{t_0}^{t_f} \mathbf{u}^T(t) \mathbf{u}(t) dt - \int_{t_0}^{t_f} \boldsymbol{\lambda}^T(t) (\boldsymbol{\Pi} \dot{\mathbf{x}}(t) - \mathbf{A} \mathbf{x}(t) - \mathbf{B} \mathbf{u}(t)) dt \quad (3.78)$$

The Euler-Lagrange equations [23] for this variational problem are as follows:

$$\begin{aligned} \mathbf{A}^T \boldsymbol{\lambda}(t) + \boldsymbol{\Pi}^T \dot{\boldsymbol{\lambda}}(t) &= \mathbf{0}, \\ \mathbf{B}^T \boldsymbol{\lambda}(t) + \mathbf{u}(t) &= \mathbf{0} \\ \boldsymbol{\Pi} \dot{\mathbf{x}}(t) + \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t) &= \mathbf{0} \end{aligned} \quad (3.79)$$

From these three equations the following system of differential equations is derived:

$$\begin{aligned} \boldsymbol{\Pi} \dot{\mathbf{x}}(t) &= \mathbf{A} \mathbf{x}(t) - \mathbf{B} \mathbf{B}^T \boldsymbol{\lambda}(t) \\ \boldsymbol{\Pi}^T \dot{\boldsymbol{\lambda}}(t) &= -\mathbf{A}^T \boldsymbol{\lambda}(t). \end{aligned} \quad (3.80)$$

These equations can be written compactly in matrix form as:

$$\boldsymbol{\Pi}_1 \dot{\boldsymbol{\gamma}}(t) - \mathbf{A}_1 \boldsymbol{\gamma}(t) = \mathbf{0} \quad (3.81)$$

where

$$\boldsymbol{\Pi}_1 = \begin{bmatrix} \boldsymbol{\Pi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Pi}^T \end{bmatrix}, \boldsymbol{\gamma}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \boldsymbol{\lambda}(t) \end{bmatrix} \quad (3.82)$$

and

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{A} & -\mathbf{B} \mathbf{B}^T \\ \mathbf{0} & -\mathbf{A}^T \end{bmatrix}. \quad (3.83)$$

This is a system of eight differential equations. A total of eight initial or boundary conditions are required to obtain a unique solution.

3.1 Numerical Solution

Transposing (3.114) one obtains,

$$\dot{\boldsymbol{\gamma}}(t)^T \boldsymbol{\Pi}_1^T - \boldsymbol{\gamma}(t)^T \mathbf{A}_1^T = \mathbf{0} \quad (3.84)$$

Now this is an ordinary system of differential equations and can be solved numerically by discretizing with the methods described in [24] and in [25]. The outline of the method is as follows:

the vector of states is discretized at the intersitial points, i.e., between the sample $t_k < s_k < t_{k+1}$ for $k = 1, \dots, n - 1$. The resulting discretized states are written in a matrix as follows:

$$\begin{bmatrix} \gamma_1^T \\ \gamma_2^T \\ \vdots \\ \gamma_n^T \end{bmatrix} = \begin{bmatrix} \gamma_1(s_0) & \gamma_1(s_1) & \dots & \gamma_1(s_f) \\ \gamma_2(s_0) & \gamma_2(s_1) & \dots & \gamma_2(s_f) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_n(s_0) & \gamma_n(s_1) & \dots & \gamma_n(s_f) \end{bmatrix} \mathbf{D}_0^T = \mathbf{\Gamma}^T \mathbf{D}_0^T \quad (3.85)$$

Additionally, the derivatives of the state vector can be discretized similarly, by taking derivatives at the interstitial points. The resulting discretized derivatives of the states are written in a matrix as:

$$\begin{bmatrix} \dot{\gamma}_1^T \\ \dot{\gamma}_2^T \\ \vdots \\ \dot{\gamma}_n^T \end{bmatrix} = \begin{bmatrix} \gamma_1(s_0) & \gamma_1(s_1) & \dots & \gamma_1(s_f) \\ \gamma_2(s_0) & \gamma_2(s_1) & \dots & \gamma_2(s_f) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_n(s_0) & \gamma_n(s_1) & \dots & \gamma_n(s_f) \end{bmatrix} \mathbf{D}_1^T = \mathbf{\Gamma}^T \mathbf{D}_1^T \quad (3.86)$$

The matrices \mathbf{D}_0 and \mathbf{D}_1 are an identity operator and a numerical first derivative operator respectively. The details about their structures as well as the discretization done can be found in [25]. Finally, the discretized form of equation (3.116) is

$$\mathbf{D}_1 \mathbf{\Gamma} \mathbf{\Pi}_1^T - \mathbf{D}_0 \mathbf{\Gamma} \mathbf{A}_1^T = \mathbf{0}. \quad (3.87)$$

where $\mathbf{\Gamma}$ and $\mathbf{\Xi}$ are the matrices derived from discretizing γ and ξ respectively. Finally, by vectorizing the last equation, one obtains

$$(\mathbf{\Pi}_1 \otimes \mathbf{D}_1 - \mathbf{A}_1 \otimes \mathbf{D}_0) \text{vec}(\mathbf{\Gamma}) = \mathbf{0} \quad (3.88)$$

which is a linear system of equations and can be solved using standard methods for linear systems of equations. In this paper, this linear system with the constraints from (3.76) was solved using SVD. The system and the constraints have the form

$$\begin{aligned} \mathbf{L} \mathbf{y} &= \mathbf{0} \\ \mathbf{C}^T \mathbf{y} &= \mathbf{d} \end{aligned} \quad (3.89)$$

where

$$\mathbf{L} = (\mathbf{\Pi}_1 \otimes \mathbf{D}_1 - \mathbf{A}_1 \otimes \mathbf{D}_0), \quad \mathbf{y} = \text{vec}(\mathbf{\Gamma}) \quad (3.90)$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{E} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{E} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}^T, \quad (3.91)$$

$$\mathbf{d} = [\mathbf{I}_C^T, \mathbf{B}_C^T]^T. \quad (3.92)$$

In (3.91) E is the $2 \times n$ matrix (where n is the number of discretization steps) of the form

$$E = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 1 \end{bmatrix}. \quad (3.93)$$

Using singular value decomposition on the matrix L one gets

$$L = USV^T = \begin{bmatrix} U_r & \tilde{U} \end{bmatrix} \begin{bmatrix} S_r & 0 \\ 0 & \Delta \end{bmatrix} \begin{bmatrix} V_r^T \\ \tilde{V}^T \end{bmatrix}, \quad (3.94)$$

where Δ is the block matrix with zero (or very small) singular values, r is the rank of the matrix L , and U and V have been partitioned accordingly [26]. It is important to note that the columns of \tilde{V} form a basis for the nullspace of L and as such the solution to the equation $Ly = 0$ will be of the form

$$y = \tilde{V}\alpha \quad (3.95)$$

where α is a vector of parameters. This represents the set of solutions to the homogeneous differential equation. To determine the values of the parameters in α one can use the constraints. Namely

$$\begin{aligned} C^T y &= C^T \tilde{V}\alpha = d, \quad \text{and thus} \\ \alpha &= (C\tilde{V})^{-1} d, \end{aligned} \quad (3.96)$$

which finally leads to the solution of the equations in the form

$$y = \tilde{V} (C\tilde{V})^{-1} d \quad (3.97)$$

4 Computation and Results

4.1 Computation

The canonical system in (3.114) was solved using (3.97). The system parameters are cylinder areas of $A_A = A_B = 6 \cdot 10^{-2} \text{ m}^2$, friction parameter $b = 1.4 \cdot 10^7 \frac{\text{Ns}}{\text{m}}$, mass $m = 4 \cdot 10^5 \text{ kg}$ and a bulk modulus of $\beta = 1.5 \cdot 10^9 \text{ Pa}$. The value for the friction parameter b was identified during a cutting experiment. Additionally, the values for the linearized terms $C_{q_1}, C_{q_2}, C_{p_1}$ and C_{p_2} were calculated for the supply pressure of $300 \cdot 10^5 \text{ Pa}$ and it is assumed that the initial values for the pressures on the both sides will be $10 \cdot 10^5 \text{ Pa}$. The values for the proportional parts of the controllers were determined experimentally. Furthermore, the value of the weighting matrices Q and R are

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix}, R = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}. \quad (3.98)$$

The optimal control solution was then compared to both PID and LQR controllers. The main purpose was to compare the results of these two controllers with the optimal control solution. Furthermore, the performance and energy efficiency are two important factors which were observed. All algorithms were computed in MATLAB[®].

4.2 Results

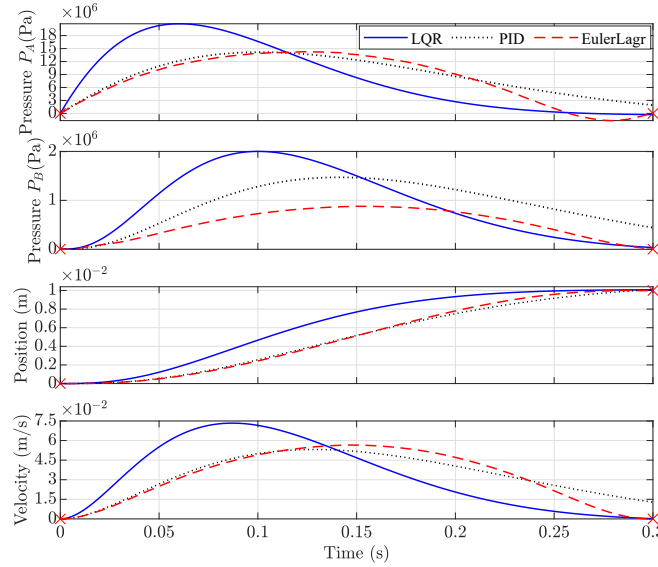


Figure 3: Results of the first simulation where all initial and final values are set to zero, except for the end position which is set to $y_1 = 0.01 m$. For the position and the pressure P_B it can be seen that the LQR and PID controller overshoot and will not end at the desired values. On the other hand, the optimal control algorithm meets the set limits at the exact positions. It also shows improved energy performance compared to PID and LQR, which can be seen in the curves of the velocity and the pressure P_A .

Two different experiments were performed for the same system. First, for both cases the initial values for the system are defined as $I_C = [0 \ 0 \ 0 \ 0]^T$. In the first case the system was simulated for a time of $t \in [0, 0.3] s$. The final values of the system are $B_C = [0 \ 0 \ 0.01 \ 0]^T$. For the PID and LQR controllers, step functions with an input values of $y_1 = 0.01 m$ for position and $P_B = 0 Pa$ for pressure have been defined.

In Fig. 3 the optimal control algorithm achieves the set boundary values at the exact location. It is also more energy efficient when compared with the PID and LQR.

In the second case the simulation time was set to $t \in [0, 0.6] s$. The final values in this case are set to $B_C = [0 \ 10 \cdot 10^5 \ 0.01 \ 0]^T$. As in the first case, step functions with set values of $y_1 = 0.01 m$ for position and $P_B = 10 \cdot 10^5 Pa$ for pressure are defined as the input for the PID and LQR

controllers. Fig. 4 shows that in this case both the LQR and PID controller have a steady-state error for both the position and the pressure. On the other hand the optimal control algorithm successfully attains the set values accurately. The system in both cases Fig. 3 and Fig. 4 achieves its maximum velocity near the mid point of the path, demonstrating the energy efficient performance of this method.

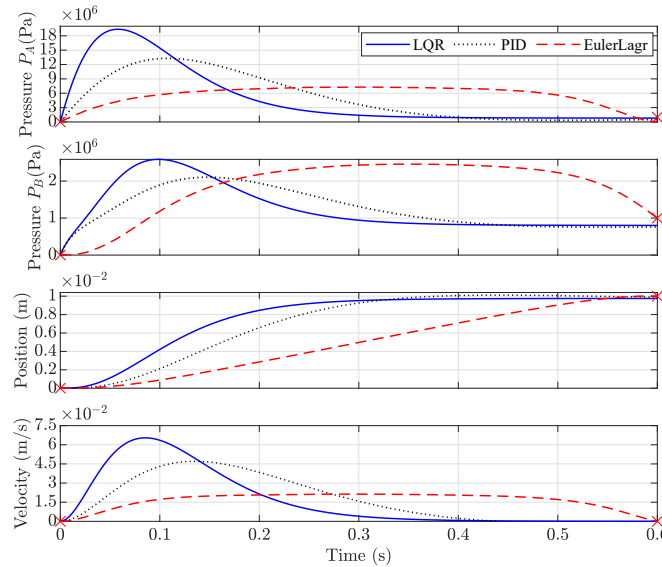


Figure 4: Results of the second simulation where the $t \in [0, 0.6]s$. All initial and final values are set to zero, except for the final position which is set to $y_1 = 0.01 m$ and the pressure which is set to $P_B = 10 \cdot 10^5 Pa$. For the position and the pressure P_B it can be seen that the LQR and PID controller have a steady state error and will not end at the desired values. On the other hand, the optimal control algorithm meets the set limits at the exact positions. It also shows improved energy performance compared to PID and LQR, which can be seen in the curves of the velocity and the pressure P_A

5 Conclusions

The results in this paper show that the new matrix-based approach, which uses the mass matrix method and interstitial derivatives, can be used to solve systems of stiff differential equations. The optimal control method was tested for two different time spans and boundary conditions. In both cases, the proposed method shows improved overall and energy efficiency compared to a linear quadratic regulator (LQR) and a PID controller. It reaches the final values at exact locations without overshoot or steady state error.

Acknowledgements

This work was partially funded by the European Institute of Innovation and Technology (EIT), a body of the European Union which receives support from the European Union's Horizon 2020 research and innovation programme. This was carried out under Framework Partnership Agreement No. 17031 (MaMMa - Maintained Mine & Machine).

Bibliography

- [1] Q. Yuan, E. Corporation, J. Y. Lew, and E. Corporation, "Electronic Flow Control Valve (EFCV) with Pressure Compensation Capability."
- [2] B. Eriksson, "Control Strategy for Energy Efficient Fluid Power Actuators - Utilizing Individual Metering," *Science And Technology*, no. 1341, p. 70, 2007.
- [3] E. ZaeV, G. Rath, and H. Kargl, "Energy Efficient Active Vibration Damping," in *13th Scandinavian International Conference on Fluid Power*, sep 2013, pp. 355–364.
- [4] C. Ciftci, H. S. S. Cayci, M. T. Atay, B. Toker, B. Guncan, and A. T. Yildirim, "The numerical solutions for stiff ordinary differential equations by using interpolated variational iteration method with comparison to exact solutions," *AIP Conference Proceedings*, vol. 1978, 2018.
- [5] S. A. Yatim, Z. B. Ibrahim, K. I. Othman, and M. B. Suleiman, "A numerical algorithm for solving stiff ordinary differential equations," *Mathematical Problems in Engineering*, vol. 2013, 2013.
- [6] C. Curtis and J. Hirschenfelder, "Integration of Stiff Equations," pp. 235–243, 1952.
- [7] L. F. Shampine and M. W. Reichelt, *The MATLAB ode suite*, 1997, vol. 18, no. 1.
- [8] G. Rath, M. Harker, and E. ZaeV, "Direct numerical solution of stiff ODE systems in optimal control," in *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, 2017, pp. 1–5.
- [9] G. Rath and E. ZaeV, "Optimal Control for Hydraulic System With Separate Meter-in and Separate Meter-Out," *The 15th Scandinavian International Conference on Fluid Power, SICFP'17, June 7-9, 2017, Linköping, Sweden*, pp. 125–134, 2017.
- [10] K. Dupree, P. M. Patre, Z. D. Wilcox, and W. E. Dixon, "Asymptotic optimal control of uncertain nonlinear Euler-Lagrange systems," *Automatica*, vol. 47, no. 1, pp. 99–107, 2011. [Online]. Available: <http://dx.doi.org/10.1016/j.automatica.2010.10.007>

- [11] T. N. Ta, C. S. Tran, and Y. L. Hwang, "The Kinematic and Dynamic Analysis of Hydraulic Control System Based on the Lagrangian Force Method," *International Journal of Computational Methods*, vol. 15, no. 5, pp. 1–18, 2018.
- [12] M. Pourebrahim, A. S. Ghafari, and M. Pourebrahim, "Designing a LQR controller for an electro-hydraulic-actuated-clutch model," in *2016 2nd International Conference on Control Science and Systems Engineering (ICCSSE)*. IEEE, jul 2016, pp. 82–87.
- [13] S. E. Skariya, B. Sebastian, and M. Namboodiripad, "Integrated Optimal Control of Reusable launch Vehicle and Actuation system using Linear Quadratic Regulator," *IFAC Proceedings Volumes*, vol. 47, no. 1, pp. 840–846, 2014.
- [14] F. Liccardo, S. Strano, and M. Terzo, "Real-time nonlinear optimal control of a hydraulic actuator," *Engineering Letters*, vol. 21, no. 4, pp. 241–246, 2013.
- [15] M. Sniegucki, M. Gottfried, and U. Klingauf, "Optimal Control of Digital Hydraulic Drives using Mixed-Integer Quadratic Programming," *IFAC Proceedings Volumes*, vol. 46, no. 23, pp. 827–832, 2013.
- [16] P. Hołobut, "Time-optimal control of hydraulic manipulators with path constraints," *Journal of Theoretical and Applied Mechanics*, vol. 43, no. January, pp. 523–538, 2005.
- [17] I. Gladwell, L. Shampine, and S. Thompson, *Solving ODEs with MATLAB*. USA: Cambridge University Press, 2003.
- [18] B. N. Datta, *Numerical Methods for Linear Control Systems Design and Analysis*. San Diego, Calif. :: Elsevier Academic Press., 2004.
- [19] K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*. ISA - The Instrumentation, Systems and Automation Society, 1995.
- [20] Y. F. Liu, J. Li, Z. M. Zhang, X. H. Hu, and W. J. Zhang, "Experimental comparison of five friction models on the same test-bed of the micro stick-slip motion system," pp. 15–28, 2015.
- [21] P. Chapple, *Principles of Hydraulic Systems Design*, vol. 2 ed., 2015.
- [22] H. E. Merrit, "Hydraulic Control Systems," pp. 76 – 131, 1967.
- [23] I. M. Gelfand and S. V. Fomin, *Calculus of Variations*, ser. Dover Books on Mathematics. Dover Publications, 2012.
- [24] M. Harker and P. O’Leary, "Approximation of physical measurement data by discrete orthogonal eigenfunctions of linear differential operators," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 41, no. 5, pp. 804–824, 2017.

- [25] G. Stojanoski, D. Ninevski, G. Rath, and M. Harker, "Multidimensional Trajectory Tracking for Numerically Stiff Independent Metering System," no. Submitted to: 17th Scandinavian International Conference on Fluid Power SICFP21, 2021.
- [26] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

Multidimensional Trajectory Tracking for Numerically Stiff Independent Metering System

Goran Stojanoski¹, Dimitar Ninevski¹, Gerhard Rath¹ and
Matthew Harker²

¹University of Leoben, A8700 Leoben, Austria
{goran.stojanoski, dimitar.ninevski, gerhard.rath}@unileoben.ac.at
<http://automatiom.unileoben.ac.at>

²Faculty of Engineering and Applied Science,
Ontario Tech University, Ontario, Canada
matthew.harker@ontariotechu.ca

Abstract

This paper presents a new approach for solving an optimal control problem in a hydraulic system, using a variational calculus method. It uses a path tracking method of two different states with different units and of different magnitude. To ensure the uniqueness of the solution, two regularization terms were introduced, whose influence is regulated by regularization parameters. The system of differential equations, obtained from the Euler-Lagrange equations of the variational problem, was solved by a mass matrix method and discretized with linear differential operators at the interstitial points for numerical stability. This enabled the calculation of the control variables, despite the stiffness of the numerical problem. The results obtained show an energy-efficient performance and no oscillations. Finally, a Simulink model of the hydraulic system was created in which the calculated control variables were inserted as feed-forward inputs, to verify the results.

Keywords: Hydraulics, Optimal control, Independent metering, Euler-Lagrange.

1 Introduction

The theory of optimal control has existed for several centuries [1]. When applied to a hydraulic system, it dampens all vibrations at the end of the motion and still achieves the desired values

[2]. However, due to the extremely non-linear dynamics its use in hydraulics is a challenge [3]. With the increasing demands for accuracy and performance in mining, especially for tunnel boring machines, the demand for new methods is also increasing. In addition, these new methods must deliver energy efficient control results and reduce the operating costs of the entire process. The new state of the art independent metering valves, which are an essential part of these systems, offer very flexible control strategies [4]. Since the valves move much faster than the natural frequency of the load system, the ODEs to be solved become numerically very stiff [5]. For this reason these systems cannot be solved with conventional solvers [6]. Rath in [5] shows that the use of exponential matrix results in unstable solutions for these type of systems. In this paper, the mass matrix method and interstitial derivatives are used to compute a stable numerical solution. Since tunnel boring machines have a given profile which has to be cut, optimal path tracking methods are suitable for this type of systems. Compared to conventional methods, optimal path tracking is often used for the navigation of mining equipment in mines. It increases efficiency and reduces working time, which in turn increases the safety of the process [7] [8]. In most cases, there is a given path that the system must follow [9] [10]. One of the methods that is frequently used is model predictive control (MPC) [11] [12] because it can take the constraints of the system into account [13]. Wang in [14] improves the efficiency of the ant colony algorithm to find the optimal path for barrier environments of varying complexity. Path tracking is also frequently used for control of multi-dimensional hydraulic manipulators. In [15], the trajectory of a one-armed hydraulic manipulator is tracked by a digital hydraulic system. Kalaiarassan also shows that the 5-bit digital flow control unit performs much better than the 4-bit system. Rudolfsen [16] solves the kinematics of a crane for operation in the vertical plane. He uses the inverse to compensate and identify the non-linearity of the static dead zone input signals. A global least squares method is used in [17] to determine the optimal control input for multi-dimensional path tracking of a hydraulic crane driven by electric drives. In [18], a new control law based on the sliding perturbation observer (SPO) is designed. Here the SPO is used to eliminate all disturbances that come from the environment, dynamic uncertainties and modeling errors. Chin in [19] presents a new type of contour tracking control which uses the force calculation for hydraulic parallel manipulators. Zhang in [20] uses the D-H (Denavit - Hartenberg) method to set the coordinates for the path of a hydraulic excavator. On the other hand, Kang [21] uses a PD controller with dead zone compensation to track the three-dimensional path. However, all of the above methods focus on tracking one or more parameters which have the same unit and magnitude and, in most cases, the position of the given system is tracked. Tunnel boring machines have very complex hydraulic circuits. In the case presented here, the valve is equipped with pressure and flow controllers that actuate the system. Since the external forces are very high, the set pressure value at the pressure controller is usually very high. This will increase the mechanical stiffness of the hydraulic system which in turn increases the mechanical stiffness of the overall system. However, the PID controllers normally used in these valves show a very oscillating performance when the value of the pressure changes. In [22] it is shown that different pressure values during a motion can lead to an improved energy performance of these

2 System Model

2.1 Mechanical System

The mechanical system consists of a large mass rotating around a point O (Fig. 1) with a lever of length L . The hydraulic cylinder exerts the force F on a smaller lever of length l . If we assume that the system makes small movements around the fulcrum O, the effective mass of the system is

$$m = \left(\frac{l}{L}\right)^2 M. \quad (3.99)$$

If the position of the mass m in Fig. 1 is labeled as y_1 , the equation of motion will have the form

$$m\ddot{y}_1 = P_A A_A - P_B A_B - b\dot{y}_1. \quad (3.100)$$

2.2 Hydraulic System

The hydraulic part of the system consists of a hydraulic pump, two independent metering valves and a hydraulic actuator. In this system the friction consists of several components. The oil flow through the valve orifice depends on the pressure which can cause damping. In addition, the movement of the load and the steel structure also contribute to the overall friction. The hydraulic cylinder has a highly non-linear friction behavior. A complete mathematical model of friction includes the Stribeck and stiction effect as well as Coulomb and viscous friction [23]. In the actual machine however, the greatest contribution comes from the rotating load that performs a cutting operation. Therefore, in eq. (3.100) the friction is assumed to be viscous with coefficient b .

2.3 Dynamics of the system

The displacement of the mechanical system is controlled by a position controller which actuates the two independent metering valves. The valves offer different operating modes for different load scenarios [4]. In this case it is assumed that the load on the system is passive when the movement is positive. This means that the actuating side of the system is flow controlled and the rear side is pressure controlled. The flows supplied by the valves are a function of the spool movement x_v and the pressures P_A, P_B [24],

$$Q_A = C_{q1} x_{v1} - C_{p1} P_A, \quad Q_B = C_{q2} x_{v2} + C_{p2} P_B \quad (3.101)$$

where $C_{q1}, C_{q2}, C_{p1}, C_{p2}$ are the valve linearization coefficients and x_{v1}, x_{v2} are the valves' spool positions. If eq. (3.101) is combined with the equations for the flow through the cylinder [25] the pressure equations are obtained:

$$P_A = \frac{C_{q1} x_{v1} - A_A v}{\frac{V_A}{\beta} s + C_{p1}}, \quad P_B = \frac{A_B v - C_{q2} x_{v2}}{\frac{V_B}{\beta} s + C_{p2}}, \quad (3.102)$$

$$\mathbf{B} = \begin{bmatrix} k_A C_{q1} \frac{\beta}{V_A} & 0 \\ 0 & -k_B C_{q2} \frac{\beta}{V_B} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.106)$$

In eq. (3.104, eq. (3.105) and eq. (3.106) the value of the bulk modulus β and the mass m are on both sides of the equations. Since β and m have very high values, a fraction of these values was stored in the mass matrix to ensure a numerically stable solution.

3 Solution of the problem

The path tracking problem can be formulated as determining the input vector \mathbf{u} , so that a desired output of the system is achieved. This problem can be formulated in terms of variational calculus, as finding the optimal input vector \mathbf{u} for which the cost function

$$\frac{\mu_1^2}{2} \int_{t_0}^{t_f} (x_2(t) - \xi_2(t))^2 dt + \frac{\mu_2^2}{2} \int_{t_0}^{t_f} (x_3(t) - \xi_1(t))^2 dt \quad (3.107)$$

is minimized. Note that because of the way the vector \mathbf{u} is defined, x_2 tracks ξ_2 and x_3 tracks ξ_1 . The integrals quantify the least-squares differences between the states $x_2(t), x_3(t)$ and the desired paths $\xi_2(t), \xi_1(t)$ respectively. Since the states $x_2(t), x_3(t)$ are measured in different units of different magnitude, Pa and m respectively, a normalization of the least squares differences is necessary. This is done with the parameters μ_1, μ_2 . The previous equation can be rewritten as

$$\frac{\mu_1^2}{2} \int_{t_0}^{t_f} (\mathbf{e}_2^T \mathbf{x}(t) - \xi_2(t))^2 dt + \frac{\mu_2^2}{2} \int_{t_0}^{t_f} (\mathbf{e}_3^T \mathbf{x}(t) - \xi_1(t))^2 dt, \quad (3.108)$$

where $\mathbf{e}_2 = [0 \ 1 \ 0 \ 0]^T$ and $\mathbf{e}_3 = [0 \ 0 \ 1 \ 0]^T$ are coordinate unit vectors. In order to get a unique solution, a regularization parameter is also necessary [26], which in the case discussed here looks as follows:

$$\frac{\mu_3^2}{2} \int_{t_0}^{t_f} \dot{\mathbf{u}}^T(t) \dot{\mathbf{u}}(t) dt + \frac{\mu_4^2}{2} \int_{t_0}^{t_f} \ddot{\mathbf{u}}^T(t) \ddot{\mathbf{u}}(t) dt. \quad (3.109)$$

Finally, the functional which needs to be minimized will have the following form:

$$\begin{aligned} J(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)) = & \\ & \frac{\mu_1^2}{2} \int_{t_0}^{t_f} (\mathbf{e}_2^T \mathbf{x}(t) - \xi_2(t))^2 dt + \frac{\mu_2^2}{2} \int_{t_0}^{t_f} (\mathbf{e}_3^T \mathbf{x}(t) - \xi_1(t))^2 dt \\ & + \frac{\mu_3^2}{2} \int_{t_0}^{t_f} \dot{\mathbf{u}}^T(t) \dot{\mathbf{u}}(t) dt + \frac{\mu_4^2}{2} \int_{t_0}^{t_f} \ddot{\mathbf{u}}^T(t) \ddot{\mathbf{u}}(t) dt \\ & - \int_{t_0}^{t_f} \boldsymbol{\lambda}^T(t) (\boldsymbol{\Pi} \dot{\mathbf{x}}(t) - \mathbf{A} \mathbf{x}(t) - \mathbf{B} \mathbf{u}(t)) dt \end{aligned} \quad (3.110)$$

In eq. (3.110) only the ratio $\mu_1 : \mu_2 : \mu_3 : \mu_4$ is relevant, thus one of the parameters can be set arbitrarily. That is why μ_1 is set to be 1. The Euler-Lagrange equations [27] for this variational problem are as follows:

$$\begin{aligned} \mathbf{e}_2 \mathbf{e}_2^T \mathbf{x} - \mathbf{e}_2 \xi_2 + \mu_2^2 \mathbf{e}_3 \mathbf{e}_3^T \mathbf{x} - \mu_2^2 \mathbf{e}_3 \xi_1 + \mathbf{A}^T \boldsymbol{\lambda} + \boldsymbol{\Pi}^T \dot{\boldsymbol{\lambda}} &= \mathbf{0}. \\ \mathbf{B}^T \boldsymbol{\lambda} - \mu_3^2 \ddot{\mathbf{u}} + \mu_4^2 \mathbf{u}^{(4)} &= \mathbf{0} \\ \boldsymbol{\Pi} \dot{\mathbf{x}}(t) + \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t) &= \mathbf{0}. \end{aligned} \quad (3.111)$$

From these three equations the following system of differential equations is derived:

$$\begin{aligned} \boldsymbol{\Pi}^T \dot{\boldsymbol{\lambda}} &= -\mathbf{E}_{23} \mathbf{x} + \mathbf{e}_2 \xi_2 + \mu_2^2 \mathbf{e}_3 \xi_1 - \mathbf{A}^T \boldsymbol{\lambda} \\ \mathbf{u}^{(4)} &= -\frac{1}{\mu_4^2} \mathbf{B}^T \boldsymbol{\lambda} + \frac{\mu_3^2}{\mu_4^2} \ddot{\mathbf{u}} \\ \boldsymbol{\Pi} \dot{\mathbf{x}}(t) &= \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t) \end{aligned} \quad (3.112)$$

where

$$\mathbf{E}_{23} = \mathbf{e}_2 \mathbf{e}_2^T + \mu_2^2 \mathbf{e}_3 \mathbf{e}_3^T. \quad (3.113)$$

These equations can be written compactly in matrix form as:

$$\boldsymbol{\Pi}_1 \dot{\boldsymbol{\gamma}}(t) - \mathbf{V} \boldsymbol{\gamma}(t) - \mathbf{W} \boldsymbol{\xi}(t) = \mathbf{0} \quad (3.114)$$

where

$$\begin{aligned} \boldsymbol{\Pi}_1 &= \begin{bmatrix} \boldsymbol{\Pi} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Pi}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \boldsymbol{\gamma}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \boldsymbol{\lambda}(t) \\ \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \\ \ddot{\mathbf{u}}(t) \\ \mathbf{u}^{(3)}(t) \end{bmatrix}, \boldsymbol{\xi} = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}, \\ \mathbf{V} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{E}_{23} & -\mathbf{A}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & -\frac{1}{\mu_4^2} \mathbf{B}^T & \mathbf{0} & \mathbf{0} & \frac{\mu_3^2}{\mu_4^2} \mathbf{I} & \mathbf{0} \end{bmatrix}, \mathbf{W} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{e}_2 & \mu_2^2 \mathbf{e}_3 \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \end{aligned} \quad (3.115)$$

3.1 Numerical Solution

Transposing eq. (3.114) one gets,

$$\dot{\boldsymbol{\gamma}}^T(t) \boldsymbol{\Pi}_1^T - \boldsymbol{\gamma}^T(t) \mathbf{V}^T - \boldsymbol{\xi}^T(t) \mathbf{W}^T = \mathbf{0} \quad (3.116)$$

Now, for this system, the input vector is known (because it consists of the desired output) and can be solved numerically by discretizing with the methods described in [28]. The methods can be described as follows: note first that any state can be discretized directly as a vector,

$$\boldsymbol{\gamma}_k = [\gamma_k(s_0) \quad \gamma_k(s_1) \quad \dots \quad \gamma_k(s_f)]^T \quad (3.117)$$

or, if the discretization is done at the interstitial points (the points t_i , between the samples s_i), one gets

$$\begin{bmatrix} \gamma_k(t_1) \\ \gamma_k(t_2) \\ \vdots \\ \gamma_k(t_f) \end{bmatrix} = \mathbf{J}_0 \begin{bmatrix} \gamma_k(s_0) \\ \gamma_k(s_1) \\ \vdots \\ \gamma_k(s_f) \end{bmatrix} \quad (3.118)$$

$$\mathbf{J}_0 = \frac{1}{16} \begin{bmatrix} 5 & 15 & -5 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ -1 & 9 & 9 & -1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & -1 & 9 & 9 & -1 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & -1 & 9 & 9 & -1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & -5 & 15 & 5 \end{bmatrix} \quad (3.119)$$

Using this, any vector of states $\boldsymbol{\gamma}$ can be discretized as a matrix in the following way

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\gamma}_1^T \\ \boldsymbol{\gamma}_2^T \\ \vdots \\ \boldsymbol{\gamma}_n^T \end{bmatrix} = \begin{bmatrix} \gamma_1(s_0) & \gamma_1(s_1) & \dots & \gamma_1(s_f) \\ \gamma_2(s_0) & \gamma_2(s_1) & \dots & \gamma_2(s_f) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_n(s_0) & \gamma_n(s_1) & \dots & \gamma_n(s_f) \end{bmatrix} \mathbf{D}_0^T \quad (3.120)$$

where

$$\mathbf{D}_0 = \begin{bmatrix} \mathbf{J}_0 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_0 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}_0 \end{bmatrix} \quad (3.121)$$

Additionally, the derivative of a state can be discretized as

$$\dot{\boldsymbol{\gamma}}_k \approx \mathbf{D}\boldsymbol{\gamma}_k, \quad (3.122)$$

where \mathbf{D} is a differentiation matrix, with the following form

$$\frac{1}{24h} \begin{bmatrix} -23 & 21 & 3 & -1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 1 & -27 & 27 & -1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & -27 & 27 & -1 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & -27 & 27 & -1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & -3 & -21 & 23 \end{bmatrix} \quad (3.123)$$

where it is assumed that the discretization is done uniformly with step size h . Hence, a vector of first derivatives of states can be discretized as the following matrix

$$\mathbf{\Gamma} = \begin{bmatrix} \dot{\gamma}_1^T \\ \dot{\gamma}_2^T \\ \vdots \\ \dot{\gamma}_n^T \end{bmatrix} = \begin{bmatrix} \gamma_1(s_0) & \gamma_1(s_1) & \dots & \gamma_1(s_f) \\ \gamma_2(s_0) & \gamma_2(s_1) & \dots & \gamma_2(s_f) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_n(s_0) & \gamma_n(s_1) & \dots & \gamma_n(s_f) \end{bmatrix} \mathbf{D}_1^T \quad (3.124)$$

$$\mathbf{D}_1 = \begin{bmatrix} \mathbf{D} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{D} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{D} \end{bmatrix} \quad (3.125)$$

Finally, the discretized form of eq. (3.116) is

$$\mathbf{D}_1 \mathbf{\Gamma} \mathbf{\Pi}_1^T - \mathbf{D}_0 \mathbf{\Gamma} \mathbf{V}^T - \mathbf{D}_0 \mathbf{\Xi} \mathbf{W}^T = \mathbf{0}. \quad (3.126)$$

where $\mathbf{\Gamma}$ and $\mathbf{\Xi}$ are the matrices derived from the discretization of γ and ξ respectively. For more details, see [28, 29]. Finally vectorizing the last equation, one gets

$$(\mathbf{\Pi}_1 \otimes \mathbf{D}_1 - \mathbf{V} \otimes \mathbf{D}_0) \text{vec}(\mathbf{\Gamma}) = (\mathbf{W} \otimes \mathbf{D}_0) \text{vec}(\mathbf{\Xi}) \quad (3.127)$$

which is a linear system of equations and can be solved using standard methods for linear systems of equations, along with appropriate constraints (SVD, QR decomposition). In order to get a unique solution, an appropriate number of initial and final conditions need to be defined. From the system of 16 differential equations eq. (3.116), a total of 16 initial or boundary conditions are given.

$$\begin{aligned} \gamma_1(t_0) &= 0 \\ \gamma_1(t_f) &= 0 & \gamma_{4-8}(t_0) &= 0 & \gamma_{10}(t_0) &= \xi_1(t_0) \\ \gamma_2(t_0) &= \xi_2(t_0) & \gamma_{4-8}(t_f) &= 0 & \gamma_{10}(t_f) &= \xi_1(t_f) \\ \gamma_2(t_f) &= \xi_2(t_f) & \gamma_9(t_0) &= \xi_2(t_0) & \gamma_{11-16}(t_0) &= 0 \\ \gamma_3(t_0) &= \xi_1(t_0) & \gamma_9(t_f) &= \xi_2(t_f) & \gamma_{11-16}(t_f) &= 0 \\ \gamma_3(t_f) &= \xi_1(t_f) \end{aligned} \quad (3.128)$$

The values in eq. (3.128) represent the real physical values of the tracked paths.

4 Computation and Results

4.1 Computation

The results of the path tracking algorithm were computed for an effective mass of $m = 4 \cdot 10^5$ kg, cylinder areas $A_A = A_B = 6 \cdot 10^{-2} \text{ m}^2$, friction parameter $b = 1.4 \cdot 10^7 \frac{\text{NS}}{\text{m}}$ and a bulk modulus of

$\beta = 1.4 \cdot 10^9$ Pa. The value for the friction parameter b was identified during a cutting experiment. Furthermore, the values for the linearized terms C_{q1}, C_{q2}, C_{p1} and C_{p2} were calculated for the supply pressure of $300 \cdot 10^5$ Pa and it is assumed that the initial values for the pressures on the both sides will be $10 \cdot 10^5$ Pa. An oblique rectangular shape was chosen as the reference path for the pressure. This increased the overall stiffness of the system during the motion. For the position we have chosen a path that is followed by one of the cutting arms of the real machine. Due to the different dimensions and magnitudes of the two tracked states, the values of the normalization parameter μ_2 was assumed to be $\mu_2 = 10^4$. The values of μ_3 and μ_4 were experimentally determined to be $5 \cdot 10^{-5}$. The path tracking algorithm was firstly computed in MATLAB, where the control variable u_1 and u_2 were calculated. Then the system presented in Fig. 2 was simulated in

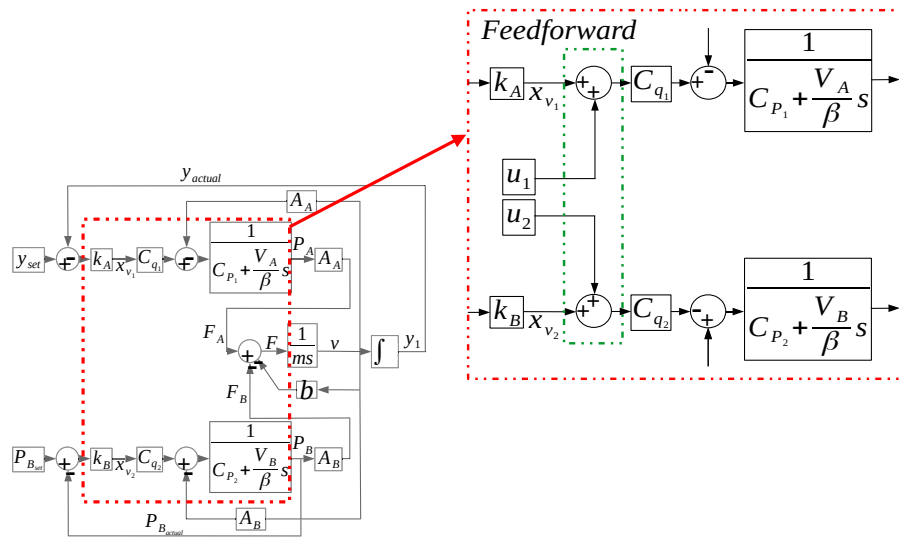


Figure 3: Implementation of the calculated control variables u_1 and u_2 into the simulation model of the system as a feed-forward.

MATLAB Simulink where the control variables were inserted as feed forward inputs to the system (see Fig. 3).

4.2 Results

The system presented in Fig. 1 was simulated for two different time intervals, namely for $t \in [0, 4]s$ and for $t \in [0, 8]s$. In Fig. 4 it can be seen that the system is following the given trajectories smoothly and precisely. The maximum value of the pressure on the rod side is set to $20 \cdot 10^5$ Pa. This increases the mechanical stiffness of the hydraulic system which will increase the mechanical stiffness of the overall system. Although the pressure changes frequently during the time interval, there are no oscillations in the solution. This is not the case when using conventional controllers. The same applies for the position. The system reaches its maximum speed near the middle point of the path, demonstrating the energy efficient performance of this method. To be able to observe

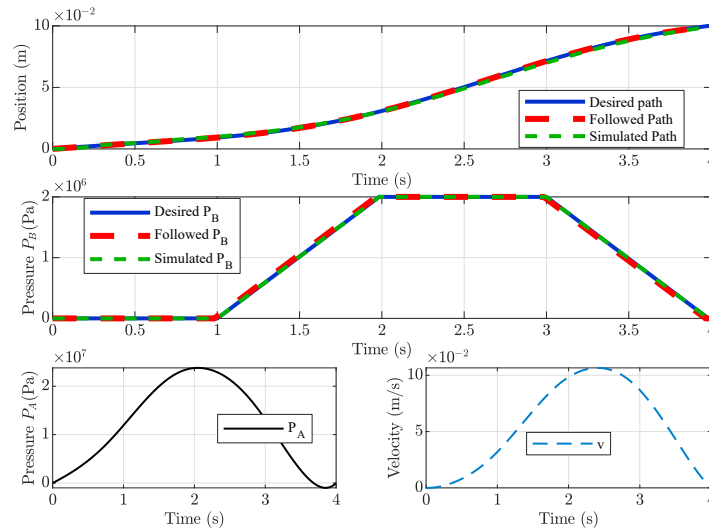
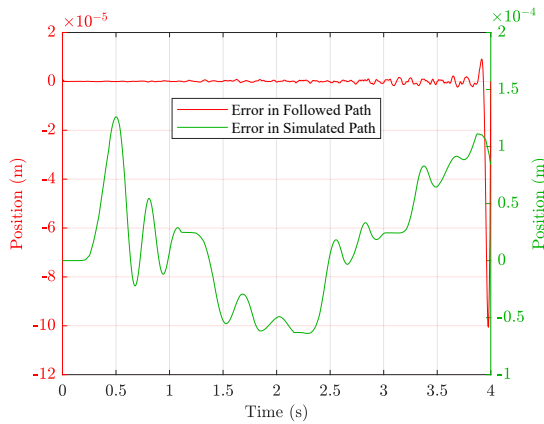
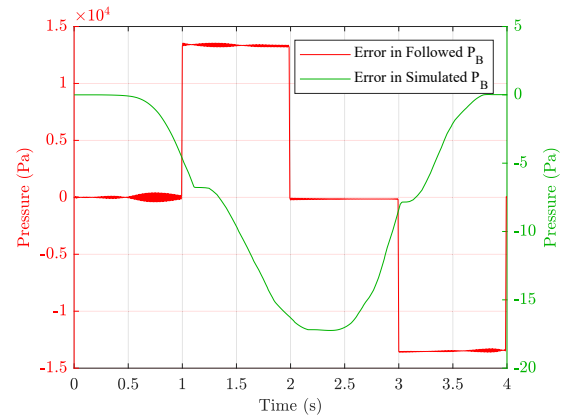


Figure 4: (Top and middle graph) - Results of the path tracking algorithm for time $t \in [0, 4] s$. The states x_3 and x_2 (position and pressure respectively) are being tracked. The followed values and the simulated values deviate only slightly from the desired values. (Bottom graphs) - The behavior of the remaining two states of the system.



(a) Error curves for the followed and simulated positions in Fig. 4 .



(b) Error curves for the followed and simulated pressures in Fig. 4 .

Figure 5: Error curves for the followed and simulated positions and pressures in Fig. 4 accordingly. It is clear that the inaccuracy for both graphs is smaller than 1 %.

the accuracy of the new method the inaccuracy of the path tracking method was observed for both the position and the pressure in Fig. 5. It can be seen that the inaccuracy for the positions and the pressures is smaller 0.01 % and 1 % accordingly. This shows that the path tracking method has a very high accuracy.

In Fig. 6 on the other hand, the system follows the same path for position and pressure but in

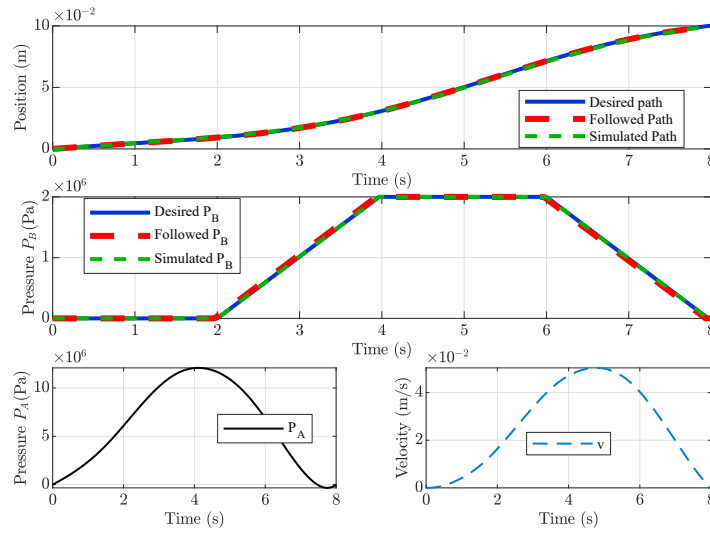
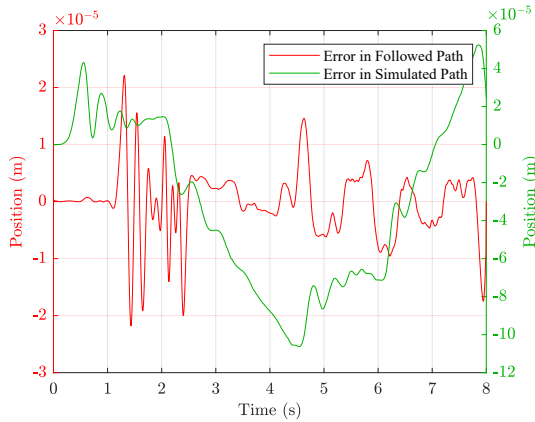
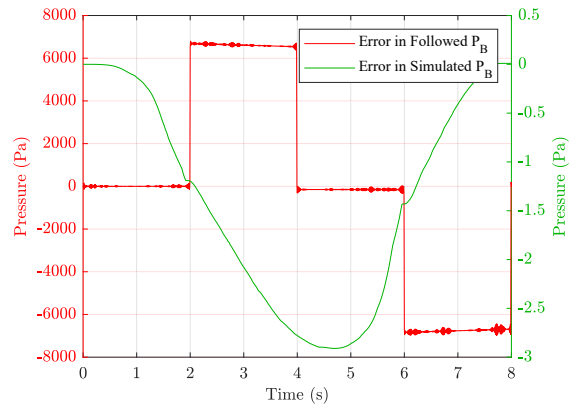


Figure 6: (Top and middle graph) - Results of the path tracking algorithm for time $t \in [0, 8]$ s. The states x_3 and x_2 (position and pressure respectively) are being tracked. (Bottom graphs) - The system is tracking the same distance over a longer period of time, which results in smaller pressure values on the piston side and a decreased velocity.



(a) Error curves for the followed and simulated positions in Fig. 6



(b) Error curves for the followed and simulated pressures in Fig. 6

Figure 7: Error curves for the followed and simulated positions and pressures in Fig. 6 accordingly. Here the path tracking method shows maximum offset of 0.005 %.

double the time. Accordingly, the pressure on the piston side P_A are much smaller than in Fig. 4. The same applies to the velocity. From Fig. 7 it can be seen that the path tracking method shows even higher accuracy for longer movements. The maximum errors are in the magnitude of 0.005 %.

5 Conclusions and Future Works

5.1 Conclusions

From the results obtained it can be concluded that the path tracking method can be used successfully to track states in different units and of different magnitudes. The calculated control variables, which were later used as feed forward inputs for the simulated system, provided paths that were very similar to the desired and to the followed ones. Furthermore, the behavior of the other two states shows that the method is energy efficient for both tested scenarios.

5.2 Future Works

In this paper we use the mass matrix method and the interstitial derivatives to find stable numerical solution to a stiff numerical problem. Further we have implemented only P (proportional part of PID) controllers on the pressure and flow controlled side. In practice however, the valves have also I (integral part of PID) controllers. The numerical solution of the integro-differential equations for these systems could be a question of future work. Additionally, the further implementation of this method on the real system is of great interest.

Bibliography

- [1] H.J. Sussmann and J.C. Willems. 300 years of optimal control: from the brachystochrone to the maximum principle. *IEEE Control Systems*, 17(3):32–44, 1997.
- [2] Gerhard Rath and Emil Zaev. Optimal Control for Hydraulic System With Separate Meter-in and Separate Meter-Out. *The 15th Scandinavian International Conference on Fluid Power, SICFP'17, June 7-9, 2017, Linköping, Sweden*, pages 125–134, 2017.
- [3] Randal W. Bea. Successive Galerkin approximation algorithms for nonlinear optimal and robust control. *International Journal of Control*, 71(5):717–743, 1998.
- [4] B Eriksson and J-O Palmberg. Individual metering fluid power systems: challenges and opportunities. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 225(2):196–211, 2011.
- [5] G Rath, M Harker, and E Zaev. Direct numerical solution of stiff ODE systems in optimal control. In *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–5, 2017.
- [6] Lawrence F. Shampine and Mark W. Reichelt. *The MATLAB ode suite*, volume 18. 1997.

- [7] B.J. Alshaer, T.T. Darabseh, and M.A. Alhanouti. Path planning, modeling and simulation of an autonomous articulated heavy construction machine performing a loading cycle. *Applied Mathematical Modelling*, 37(7):5315–5325, 2013.
- [8] B.J. Alshaer, T.T. Darabseh, and A.Q. Momani. Modelling and control of an autonomous articulated mining vehicle navigating a predefined path. *International Journal of Heavy Vehicle Systems*, 21(2):152, 2014.
- [9] Xuewu Ji, Yulong Liu, Xiangkun He, Kaiming Yang, Xiaoxiang Na, Chen Lv, and Yahui Liu. Interactive Control Paradigm-Based Robust Lateral Stability Controller Design for Autonomous Automobile Path Tracking With Uncertain Disturbance: A Dynamic Game Approach. *IEEE Transactions on Vehicular Technology*, 67(8):6906–6920, 2018.
- [10] Hans Andersen, Zhuang Jie Chong, You Hong Eng, Scott Pendleton, and Marcelo H. Ang. Geometric path tracking algorithm for autonomous driving in pedestrian environment. In *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1669–1674. IEEE, 2016.
- [11] Guoxing Bai, Li Liu, Yu Meng, Weidong Luo, Qing Gu, and Baoquan Ma. Path tracking of mining vehicles based on nonlinear model predictive control. *Applied Sciences (Switzerland)*, 9(7), 2019.
- [12] Jie Ji, Amir Khajepour, Wael William Melek, and Yanjun Huang. Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control With Multiconstraints. *IEEE Transactions on Vehicular Technology*, 66(2):952–964, 2017.
- [13] J M Maciejowski. *Predictive Control with Constraints*. Prentice Hall, England., 2002.
- [14] Tao Wang, Lianyu Zhao, Yunhui Jia, and Jutao Wang. Robot Path Planning Based on Improved Ant Colony Algorithm. In *2018 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, pages 70–76. IEEE, 2018.
- [15] G. Kalaiarassan and K. Krishnamurthy. Digital hydraulic single-link trajectory tracking control through flow-based control. *Measurement and Control (United Kingdom)*, 52(7-8):775–787, 2019.
- [16] Morten H. Rudolfson, Teodor N. Aune, Oddgeir Auklend, Leif Tore Aarland, and Michael Ruderman. Identification and Control Design for Path Tracking of Hydraulic Loader Crane. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pages 565–570, 2017.
- [17] Johannes Handler, Matthew Harker, and Gerhard Rath. Multidimensional Path Tracking With Global Least Squares Solution. *21st IFAC World Congress*, 21, 2020.

- [18] Jie Wang, Min Cheol Lee, Karam Dad Kallu, Saad Jamshed Abbasi, and Seokyoung Ahn. Trajectory tracking control of a hydraulic system using TSMCSPO based on sliding perturbation observer. *Applied Sciences (Switzerland)*, 9(7):1–17, 2019.
- [19] Jih Hua Chin, Yen His Sun, and Yuan Ming Cheng. Force computation and continuous path tracking for hydraulic parallel manipulators. *Control Engineering Practice*, 16(6):697–709, 2008.
- [20] Bin Zhang, Shuang Wang, Yuting Liu, and Huayong Yang. Research on Trajectory Planning and Autodig of Hydraulic Excavator. *Mathematical Problems in Engineering*, 2017:1–10, 2017.
- [21] Seonhyeok Kang, Jaemann Park, Seunghyun Kim, Bongju Lee, Youngbum Kim, Panyoung Kim, and H. Jin Kim. Path tracking for a hydraulic excavator utilizing proportional-derivative and linear quadratic control. In *2014 IEEE Conference on Control Applications (CCA)*, pages 808–813. IEEE, 2014.
- [22] Goran Stojanoski, Gerhard Rath, and Martin Gimpel. The Effects of Bulk Modulus on the Dynamics of Hydraulic Independent Metering Systems. *Sixteenth Scandinavian International Conference on Fluid Power*, pages 276 – 290, 2019.
- [23] Y F Liu, J Li, Z M Zhang, X H Hu, and W J Zhang. Experimental comparison of five friction models on the same test-bed of the micro stick-slip motion system. pages 15–28, 2015.
- [24] Herbert E. Merrit. *Hydraulic Control Systems*. John Wiley & Sons, Inc., 1967.
- [25] Peter Chapple. *Principles of Hydraulic Systems Design*. Momentum Press, vol. 2 edition, 2015.
- [26] Richard Bellman. *Mathematical Optimization Techniques*. 1963.
- [27] I M Gelfand and S V Fomin. *Calculus of Variations*. Dover Books on Mathematics. Dover Publications, 2012.
- [28] Matthew Harker and Gerhard Rath. Discrete Inverse Problem Approach to Path Tracking in State Space Form. In *2018 International Conference on Applied Electronics (AE)*, pages 1–4. IEEE, 2018.
- [29] Matthew Harker. *Fractional Differential Equations: Numerical Methods for Applications*. Springer International Publishing, 2020.

Chapter 4

The Method of Variable Projections

The variable projection method [8,9] is a method used for non-linear approximation; more specifically for problems where the model equation is a linear combination of non-linear functions, such as:

$$y = \alpha_1 \Phi_1(x, \beta_1) + \alpha_2 \Phi_2(x, \beta_2) + \dots + \alpha_m \Phi_m(x, \beta_m). \quad (4.1)$$

The main advantage of this method is that it separates the problem into a linear and a non-linear portion. If the nonlinear coefficients β are known, then solving for the linear coefficients α is a least-squares problem. And in order to determine the non-linear coefficients β , one needs to solve a non-linear problem of smaller dimension than the original problem. In other words, the benefit is that the non-linear minimization problem works in a reduced space, and as such less initial guesses are necessary. Additionally, it always converges in fewer iterations than the minimization of the full functional. Like with linear optimization problems discussed in Chapter 2, the following residual is of interest:

$$r_k = y_k - \alpha_1 \Phi_1(x, \beta_1) - \alpha_2 \Phi_2(x, \beta_2) - \dots + \alpha_m \Phi_m(x, \beta_m). \quad (4.2)$$

Written in matrix form this becomes:

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} \Phi_1(x_1, \beta_1) & \dots & \Phi_m(x_1, \beta_m) \\ \Phi_1(x_2, \beta_1) & \dots & \Phi_m(x_2, \beta_m) \\ \vdots & \ddots & \vdots \\ \Phi_1(x_n, \beta_1) & \dots & \Phi_m(x_n, \beta_m) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix}, \quad (4.3)$$

or written more compactly

$$\mathbf{r} = \mathbf{y} - \mathbf{A}(\beta) \boldsymbol{\alpha} \quad (4.4)$$

The least squares solution to this problem is

$$\boldsymbol{\alpha} = (\mathbf{A}^T(\beta) \mathbf{A}(\beta))^{-1} \mathbf{A}^T(\beta) \mathbf{y}. \quad (4.5)$$

The cost function now has the form

$$\begin{aligned} C(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{A}(\boldsymbol{\beta}) \boldsymbol{\alpha}\| &= \|\mathbf{y} - \mathbf{A}(\boldsymbol{\beta}) (\mathbf{A}^T(\boldsymbol{\beta}) \mathbf{A}(\boldsymbol{\beta}))^{-1} \mathbf{A}^T(\boldsymbol{\beta}) \mathbf{y}\| = \\ &= \underbrace{\|(\mathbf{I} - \mathbf{A}(\boldsymbol{\beta}) (\mathbf{A}^T(\boldsymbol{\beta}) \mathbf{A}(\boldsymbol{\beta}))^{-1} \mathbf{A}^T(\boldsymbol{\beta})) \mathbf{y}\|}_{\text{projection matrix}}. \end{aligned} \quad (4.6)$$

The matrix $\mathbf{I} - \mathbf{A}(\boldsymbol{\beta}) (\mathbf{A}^T(\boldsymbol{\beta}) \mathbf{A}(\boldsymbol{\beta}))^{-1} \mathbf{A}^T(\boldsymbol{\beta})$ is a projection matrix, which depends (varies) on the variable $\boldsymbol{\beta}$, which is where the name *variable projections* comes from. Note that the cost function now depends only on the parameters $\boldsymbol{\beta}$ and is non-linear, so a standard non-linear approach can be used to minimize and solve for $\boldsymbol{\beta}$. And once $\boldsymbol{\beta}$ is obtained, one can compute $\boldsymbol{\alpha}$ using equation 4.5.

Bibliography

- [1] A. Eisinberg, G. Franzé, and N. Salerno, “Rectangular vandermonde matrices on chebyshev nodes,” *Linear Algebra and its Applications*, vol. 338, no. 1, pp. 27–36, 2001. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S002437950100355X>
- [2] H. Wertz, “On the numerical inversion of a recurrent problem: The vandermonde matrix,” *IEEE Transactions on Automatic Control*, vol. 10, no. 4, pp. 492–492, 1965.
- [3] J. Baik, T. Kriecherbauer, K. D.-R. McLaughlin, and P. D. Miller, *Discrete Orthogonal Polynomials. (AM-164): Asymptotics and Applications (AM-164): Asymptotics and Applications (AM-164)*. Princeton University Press, 2007. [Online]. Available: <https://doi.org/10.1515/9781400837137>
- [4] G. Szegő, S. G. Szego, and A. M. Society, *Orthogonal Polynomials*, ser. American Math. Soc: Colloquium publ. American Mathematical Society, 1939. [Online]. Available: <https://books.google.at/books?id=ZOhmnsXlcY0C>
- [5] M. Harker, *Fractional Differential Equations: Numerical Methods for Applications*, ser. Studies in Systems, Decision and Control. Springer Cham.
- [6] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- [7] I. Gelfand and S. Fomin, *Calculus of Variations*, ser. Dover Books on Mathematics. Dover Publications, 2012. [Online]. Available: <https://books.google.at/books?id=CeC7AQAQBAJ>
- [8] G. H. Golub and V. Pereyra, “The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate,” *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 413–432, 1973.
- [9] G. Golub and V. Pereyra, “Separable nonlinear least squares: the variable projection method and its applications,” *Inverse Problems*, vol. 19, pp. R1–R26(1), 01 2003.

Estimating Parameters of a Sine Wave by the Method of Variable Projection

Dimitar Ninevski and Paul O'Leary
University of Leoben, A8700 Leoben, Austria
automation@unileoben.ac.at
<http://automatiom.unileoben.ac.at>

Abstract

This paper presents a new derivation and implementation of the method of variable projection for the estimation of the parameters of a sine wave model. The new approach yields insights into the cost function associated with the model and how this behaves when there is less than one cycle of the signal in the measurement period. Additionally, the new method delivers the covariance matrix for the linear parameters; permitting computations of prediction and confidence intervals for the approximation. The method is thoroughly tested using Monte Carlo simulations and the results compared with those obtained by Chen.

1 Introduction

The availability of low cost high performance MEMS accelerometers and gyroscopes [1], primarily driven by the mobile telephone development, is opening new areas of application, e.g., in medicine [2]; these are in addition to the already common use of accelerometers [3, 4]. With this, there is the an increasing demand for methods and approaches to determine the exact frequency of motion and the parameters for this motion. The four-parameter model for a sine wave is defined in the IEEE-standard 1057 [5]. It is commonly used to determine the frequency of a signal, characterize noise and signal to noise ratio in waveform digitizers, e.g. digital oscilloscopes, analog to digital converters etc [6, 7]. However, it is also relevant in any application where there is a dominant frequency of oscillation.

The ability of the MEMS devices to measure low frequencies, even down to DC, a task which is very difficult with classical piezoelectric sensors [8], enables the automatic registration of the coordinate system with respect to gravity. That is, the MEMS accelerometer can be simultaneously

used as: an accelerometer for oscillations and as an inclinometer to determine the orientation of the machine. However, this requires the reliable and exact separation of DC and oscillatory components. This is the reason to select the four parameter sine wave model [5, 9].

A key issue for the computations to be performed here is that the sine wave, in general, will not be perfectly periodic in the measurement time T ; consequently, direct Fourier methods such as the FFT will be subject to Gibbs error [10]. This error makes it difficult to estimate the parameters of a single frequency exactly and to reliably separate the DC and oscillatory components. Consequently, several efficient algorithms [6, 11–13], primarily non-linear, have been developed to estimate the desired parameters.

The paper by Chen [14] is the closest and most relevant paper for the work presented here. He derives the equations required to perform separable nonlinear least squares analytically¹ and solves them approximately using Newton’s method. However, at low cycles in the record, c_r , the matrices involved become poorly conditioned and convergence to a minimum in cost is not guaranteed.

The value of this paper lies in: a through matrix algebraic derivation for the method of variable projection applied to parameter estimation of sine waves. The availability of an explicit equation for the *variable projection functional* VPF also permits the calculation of the region of convergence for the algorithm. Given the algebraic formulation, it is then possible to calculate the covariance matrix for the linear coefficients of the model in a simple manner. Via the condition number of the projection matrix, it is possible to detect where the model becomes poorly conditioned. Numerical results are presented to permit a thorough comparison with the work of Chen [14], showing that the new implementation is superior in almost all respects. The salient m-code is provided so that the reader can implement the method and independently verify the work.

2 The four parameter sine-model in algebraic form

Let us assume that there are n measurements $\mathbf{y} = [y_1, \dots, y_n]^T$, performed at the time points $\mathbf{t} = [t_1, \dots, t_n]^T$. In this manner both \mathbf{y} and \mathbf{t} are $n \times 1$ column vectors. The signal model \mathbf{y}_m , also an $n \times 1$ vector, can be formulated as,

$$\mathbf{y}_m = a \sin(\omega \mathbf{t} + \phi) + d. \quad (4.7)$$

The four parameters we wish to estimate are amplitude a , frequency ω , phase ϕ and the DC component d . The above model can be equivalently formulated as,

$$\mathbf{y}_m = a_c \cos(\omega \mathbf{t}) + a_s \sin(\omega \mathbf{t}) + d. \quad (4.8)$$

This can be written as a linear combination of basis functions,

$$\mathbf{y}_m = \begin{bmatrix} 1 & \cos(\omega \mathbf{t}) & \sin(\omega \mathbf{t}) \end{bmatrix} \begin{bmatrix} d \\ a_c \\ a_s \end{bmatrix}. \quad (4.9)$$

¹In his paper he refers to variable projection; however, his implementation is clearly based on separability.

Note that $\mathbf{1}$ in this equation is an $n \times 1$ vector of ones.

3 Method of variable projection

The method of variable projection [15, 16], is characterized by the model being a linear combination of nonlinear functions; this is exactly how the sine wave model has been formulated in Equation 4.9. Defining the basis functions as

$$\mathbf{b}_1 \triangleq \mathbf{1}, \quad \mathbf{b}_2(\omega) \triangleq \cos(\omega t), \quad \mathbf{b}_3(\omega) \triangleq \sin(\omega t) \quad (4.10)$$

and concatenating the individual basis functions into the matrix of basis functions, one gets

$$\mathbf{B}(\omega) = [\mathbf{b}_1, \mathbf{b}_2(\omega), \mathbf{b}_3(\omega)]. \quad (4.11)$$

The notation $\mathbf{B}(\omega)$ indicates that the contents of the matrix \mathbf{B} is dependent on ω . Now defining the coefficient vector

$$\mathbf{c} \triangleq [d, a_c, a_s]^T \quad (4.12)$$

leads directly to the matrix vector equation,

$$\mathbf{y}_m = \mathbf{B}(\omega) \mathbf{c}. \quad (4.13)$$

Given an estimate for ω , a least squares estimate for \mathbf{c} as is obtained from,

$$\mathbf{c} = \mathbf{B}^+(\omega) \mathbf{y}, \quad (4.14)$$

whereby, $\mathbf{B}^+(\omega)$ denotes the Moore-Penrose pseudo inverse of $\mathbf{B}(\omega)$. Now substitution Equation 4.14 into 4.13 one obtains,

$$\mathbf{y}_m = \mathbf{B}(\omega) \mathbf{B}^+(\omega) \mathbf{y}. \quad (4.15)$$

Note that $\mathbf{P}(\omega) = \mathbf{B}(\omega) \mathbf{B}^+(\omega)$ is the projection onto the basis functions contained in $\mathbf{B}(\omega)$. This projection varies with ω , hence the name, method of variable projection. The residual vector is computed as

$$\mathbf{r} = \mathbf{y} - \mathbf{y}_m \quad (4.16)$$

leading to the cost function $E(\omega)$, defined as the sum of squares of the residual \mathbf{r} , to be calculated as

$$E(\omega) = \|\mathbf{r}\|^2 \quad (4.17)$$

$$= \|\mathbf{y} - \mathbf{B}(\omega) \mathbf{B}^+(\omega) \mathbf{y}\|^2, \quad (4.18)$$

$$= \|\{\mathbf{I} - \mathbf{B}(\omega) \mathbf{B}^+(\omega)\} \mathbf{y}\|^2. \quad (4.19)$$

This is called the *variable projection functional* (VPF). Note that the estimation of ω is now a non-linear least squares problem in one variable. Equation 4.17 permits us to explicitly calculate the cost function $E(\omega)$ as a function of ω for a given measurement \mathbf{y} . In Section 4 of [15], Golub provides the formal proofs required to determine that the Fréchet derivatives over the pseudo-inverse and projection yield the gradient of the cost function. This implies that gradient based non-linear solvers can be used to find the value of ω which minimizes the cost function $E(\omega)$.

4 Range of convergence

The range of convergence of a non-linear algorithm is dependent of the shape of the cost function. A well implemented gradient minimization procedure will converge to the optimal solution from any initial value, e.g., ω_i , in as far as the functional $E(\omega)$ has monotonic derivatives toward the optimum. Since by using Equation 4.17, the functional $E(\omega)$ can be explicitly calculated for a given measurement \mathbf{y} , the range of convergence for this algorithm can be determined.

The discrete Fourier transform (DFT), or its faster implementation the FFT, is commonly used to determine frequencies; for this reason, the cost functions for the method of variable projection $E(\omega)$ and for the DFT $E_d(\omega)$ are compared in the following:

$$E(\omega) = (\mathbf{I} - \mathbf{B}\mathbf{B}^+) \mathbf{y}, \quad (4.20)$$

$$E_d(\omega) = (\mathbf{I} - \mathbf{B}\mathbf{W}\mathbf{B}^T) \mathbf{y}. \quad (4.21)$$

whereby, scaling of the DFT with respect to the number of points and for a real signal, only half the spectrum is required.

$$\mathbf{W} = \frac{1}{n} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}. \quad (4.22)$$

For an integer number of cycles in the record, the two cost functions should agree exactly; indeed they do, which can be seen in Figure 1. It is important to note that this cost function has a C^1 discontinuity at the optimum. Such discontinuities can cause convergence difficulties with simple Newton type algorithms as used in [14].

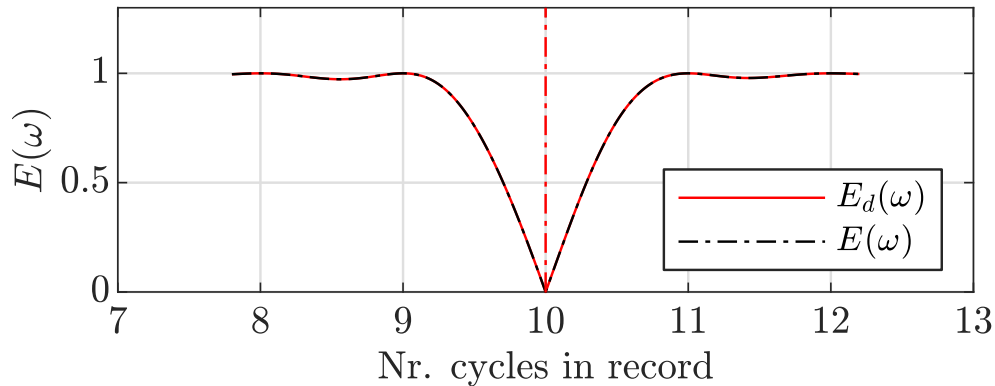


Figure 1: Normalized cost functions $E(\omega)$ and $E_d(\omega)$ for integer $c_r = 10$. Note the C^1 discontinuity at the optimum.

In the case of non-integer c_r , the two cost functions diverge at the optimum slightly, see Figure 2. The finite curvature in $E_d(\omega)$ at the optimum means the location of the minimum is not as well defined for the DFT as it is for the variable projection.

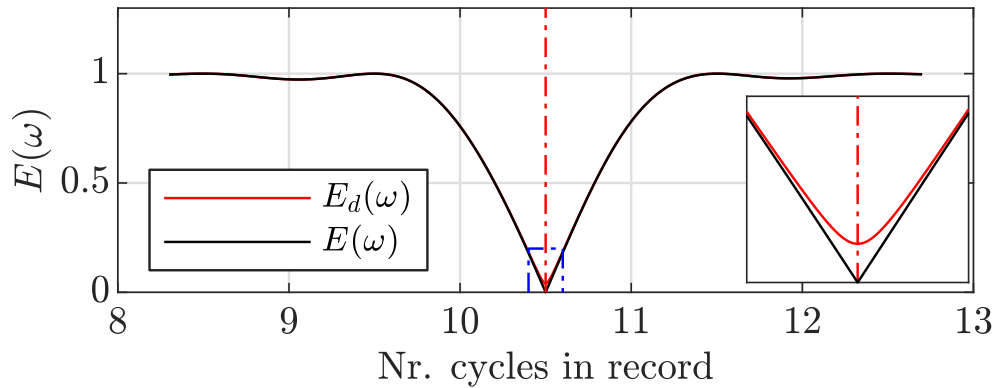


Figure 2: Normalized cost functions $E(\omega)$ and $E_d(\omega)$ for non-integer $c_r = 10.5$. Note how the cost functions diverge slightly at the optimum.

At lower values of $c_r < 1$ the two cost functions diverge strongly, see Figure 3; to the extent that the minimum of $E_d(\omega)$, no longer corresponds to the c_r value.

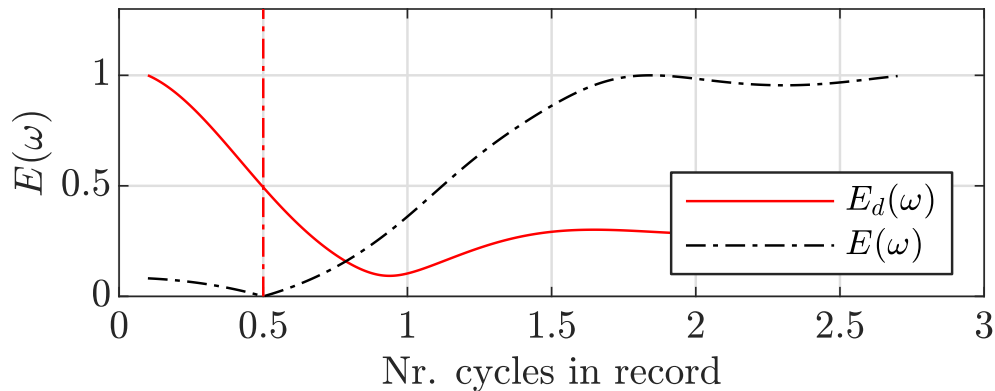


Figure 3: Normalized cost functions $E(\omega)$ and $E_d(\omega)$ for low value of c_r , in this example $c_r = 0.5$. Note the minimum of $E_d(\omega)$ does not correspond to the c_r values.

At low values of $c_r < 1$ the four term sine wave model may become inappropriate. For example, a sine wave close to the origin is very well approximated by a straight line; consequently, fewer parameters are required than the four foreseen. This is reflected in the condition number of the matrix of basis functions \mathbf{B} as seen in Figure 4. A high condition number indicates that a matrix is approaching degeneracy; which in this case, would imply that for $c_r < 0.5$ the basis functions are becoming linearly dependent.

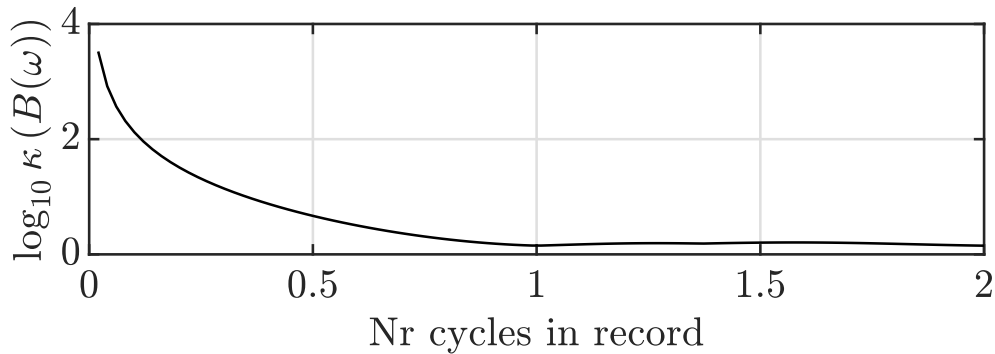


Figure 4: Condition number $\kappa(\mathbf{B}(\omega))$ in a log scale as function of ω .

5 Covariance propagation and noise bandwidth

A further advantage of the method of variable projection is that it yields a direct method of calculating the covariance propagation for the linear coefficients. Given $\mathbf{c} = \mathbf{B}^+ \mathbf{y}$, a linear mapping, then Λ_c , the covariance matrix of \mathbf{c} , can be computed as [17],

$$\Lambda_c = \mathbf{B}^+ \Lambda_y \mathbf{B}^{+\text{T}}. \quad (4.23)$$

whereby, Λ_y is the covariance matrix of the data vector \mathbf{y} . If \mathbf{y} is perturbed by i.i.d. Gaussian noise with the standard deviation σ_y , then the equation becomes

$$\Lambda_c = \sigma_y^2 \mathbf{B}^+ \mathbf{B}^{+\text{T}}. \quad (4.24)$$

An estimate for σ_y^2 can be computed from the residual vector \mathbf{r} as follows:

$$\sigma_y^2 = \frac{1}{n - n_{df}} \|\mathbf{r}\|_2^2, \quad (4.25)$$

where by n_{df} denotes the number of degrees of freedom, in this case $n_{df} = 4$, since four parameters are being determined. In this manner, the covariance of the linear coefficients can be directly computed.

An example for the approximation of a noisy sine wave is shown in Figure 5 and the corresponding covariances of the linear coefficients are given in Table 4.1.

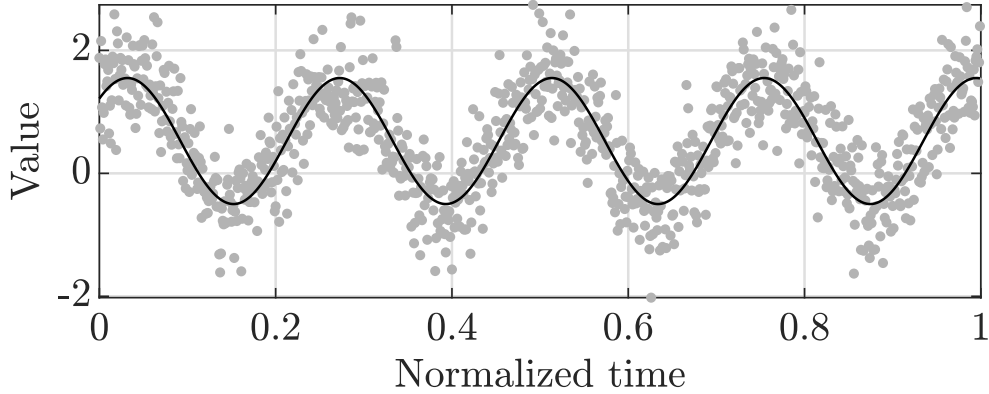


Figure 5: Synthetic sine wave used to demonstrate the computation of covariances. The model parameters were: $\omega = 4.15$, $d = 0.5$, $a_c = a_s = 1/\sqrt{2}$ with iid noise with a standard deviation of $\sigma = 0.5$ and $n = 1000$ samples were used. The estimation results are: $\omega = 4.1530$, $d = 0.4837$, $a_c = 0.7064$ and $a_s = 0.7058$. The corresponding covariance of the linear coefficients is shown in Table 4.1.

Cov	d, d	d, a_c	d, a_s
d, d	0.241	-0.015	-0.007
d, a_c	-0.015	0.474	-0.012
d, a_s	-0.007	-0.012	0.491

Table 4.1: Table of covariances for the test case shown in Figure 5. The results are scaled by 10^3 since they were very small.

6 Numerical testing

The following presents a thorough numerical testing of the new method and its comparison to the results obtained by Chen [14]. The results are obtained using Monte Carlo simulations, each run consisting of $m = 500$ independent simulations; each simulation as $n = 1000$ samples; a sine wave amplitude of $a = 1$ and independent identically distributed (iid) Gaussian noise with a standard deviation of $\sigma = 0.1$ was chosen. Based on RMS signal values this corresponds to 17.1 db. This was repeated for $k = 20$ values of c_r , such that $0 < c_r \leq 2$. Then for each value of c_r and each performance indication, a box plot is created indicating the median, IQR, upper bound, lower bound and outliers. The results are shown in figures 6, 7 and 8.

6.1 Residual error

The value of the cost functions after convergence of the algorithm is shown in Figure 6. The new algorithm produces a consistent IQR for all the values of c_r , this is exactly what one would expect, since the noise level added to the signal is independent of ω . Note the implementation proposed by Chen² fails to converge for $c_r < 0.7$.

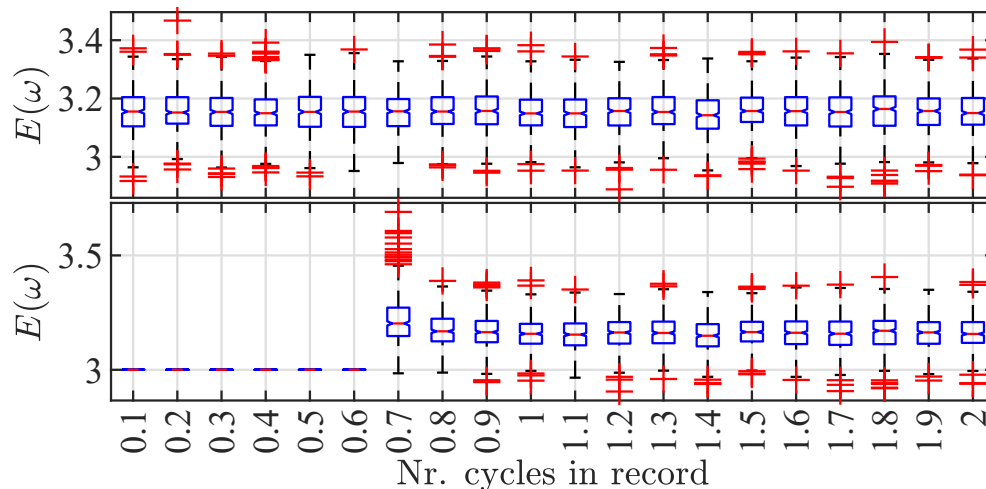


Figure 6: Box plots of cost functions obtained from Monte Carlo simulation with $m = 500$ repetitions: (top) the new solution, (bottom) Chen's implementation [14]; note, that where no data is shown, Chen's method failed to converge at least once during the m simulations. The signals over the given frequency range were all generated with $n = 1000$ samples, and the parameters $d = 0.1$, $a_c = 0$, $a_s = 1$.

6.2 Estimated frequency

The errors in the frequency estimates, ε_ω , obtained by applying the method of variable projection, as implemented in listings 1 and 2, are shown in Figure 7, together with the results from Chen's solution. Both methods yield comparable results for $c_r \geq 1$. Below this value the new method performs better and the median estimate for the frequency is consistent for values as low as $c_r = 0.2$.

²We have strictly adhered to the convergence conditions proposed by Chen in his paper [14].

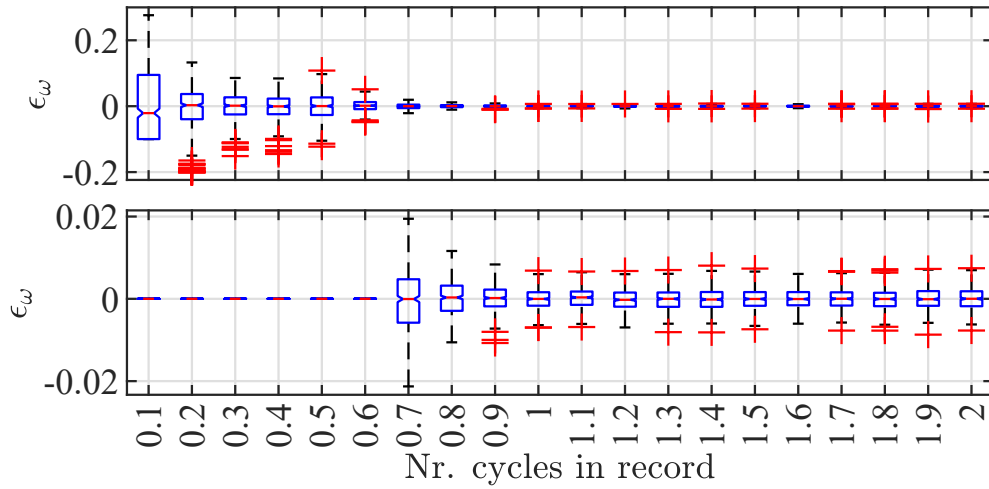


Figure 7: Box plots for the error in the estimated frequency, results are from the same Monte Carlo simulations shown in Figure 6. No results are shown where Chen’s method failed to converge.

6.3 Computational time

The CPU elapse time for each algorithm is shown in Figure 8. Once again they are comparable for $c_r \geq 1$. However, Chen’s implementation requires significant CPU time when it does not converge.

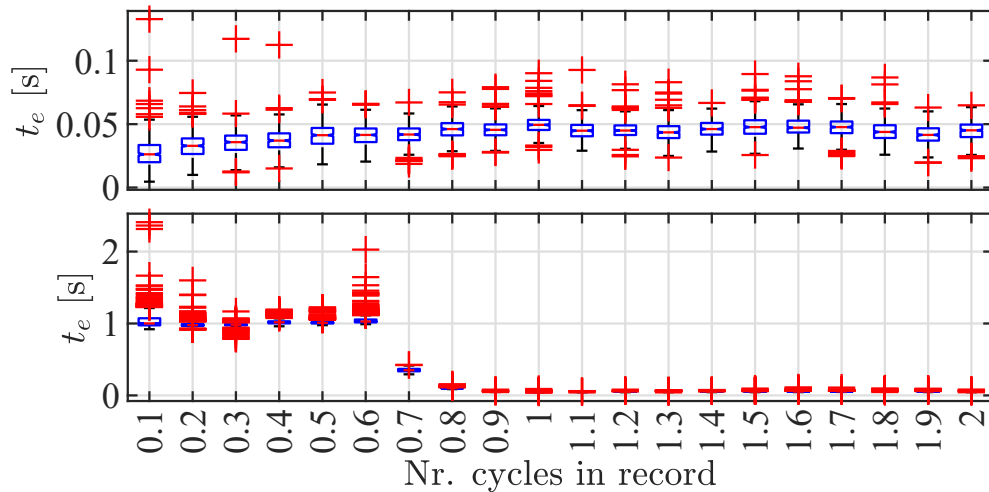


Figure 8: Box plots for the execution times t_e for each algorithm: (top) new solution, (bottom) Chen. Note, the times for Chen’s algorithm, although it did not converge, are shown, since the algorithm must run to this extend to be able to detect non-convergence. The results are from the same Monte Carlo simulations shown in Figure 6.

6.4 Residual error vs phase

The residual error as a function of phase for low number of cycles in the record $c_r = 0.5$ is shown in Figure 9. Only the results for the new approach presented in this paper are shown here, since the implementation of Chen does not converge for low values of c_r . The results demonstrate, as one would expect, are statistically independent of phase. This is due to the nature of the projection $P = B(\omega)B^+(\omega)$; the 2-norms of both, all columns and rows, have a single stationary point. Consequently, there is non tendency for the iterative minimization to converge to a local minimum.

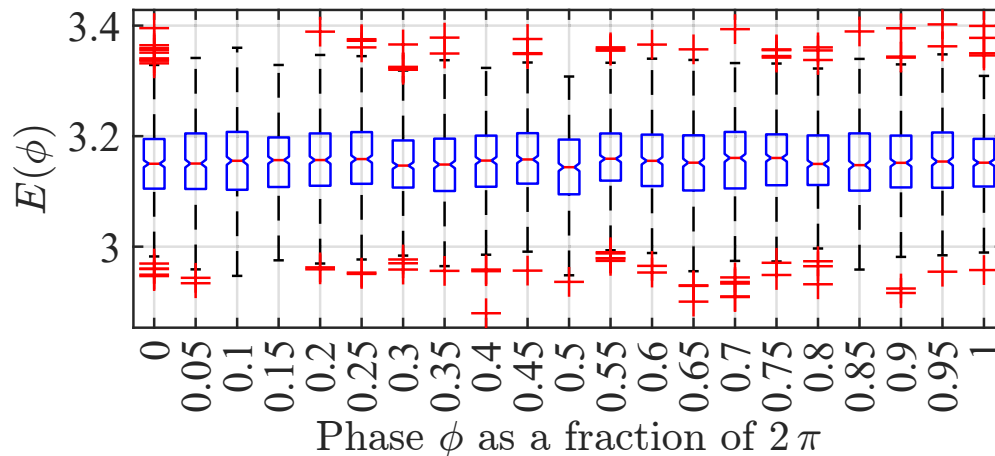


Figure 9: Box plots for the residual error $\varepsilon(\phi)$ with $c_r = 0.5$ for phase shifts in the range $0 \leq \phi \leq 2\pi$. Only the results for the new algorithm are shown, since this value of c_r the Chen approach does not converge.

7 Code implementation

Here the salient snippets of code required to implement the method of variable projection for the four parameter sine wave model are presented. The first piece of m-code is a function that is used to compute the cost function $E(\omega)$ for the method of variable projection, see Listing 1.

```
1 function cost = sineCost( omega, t, y )
2 %
3 % Setup the basis functions
4 phi = omega * t ;
5 B = [cos(phi), sin(phi), ones(size(y)) ] ;
6 % Compute the projection
7 ys = B * ( B \ y ) ;
8 % Evaluate the residual
9 r = y - ys ;
10 % compute the corresponding cost
```

```
11 cost = norm( r );
```

Listing 4.1: Variable projection cost function for the four parameter sine wave model.

As in [18], here a standard high quality generic nonlinear iteration process³ is availed of to perform the optimization. The code required for this portion is shown in Listing 2. An autonomous function `fun` is defined with one calling parameter, `omega` and one return parameter `cost`. Additionally, there are two passively passed parameters `t` and `y`: the time vector and measurement vector, respectively.

```
1 % Define the anonymous for the cost
2 fun = @(omega) sineCost( omega, t, y ) ;
3 % Wrap with the nonlinear iteration.
4 omega = lsqnonlin( fun, omega_Init );
```

Listing 4.2: Wrapping with a nonlinear solver

At the end of running the code in Listing 2, the optimal value for ω is obtained. The remaining linear coefficients c , since ω is available, can be computed according to Equation 4.14, this is implemented in Listing 3.

```
1 phi = omega * t ;
2 B = [cos(phi), sin(phi), ones(size(y)) ] ;
3 % Compute the projection
4 cfs = B \ y ;
```

Listing 4.3: Compute the linear parameters.

Finally, if required the covariances of the linear coefficients can be computed according to the m-code in Listing 4.

```
1 nrParams = 4 ;
2 % Number of degrees of freedom
3 df = length(y) - nrParams ;
4 % Estimate the standard deviation of y
5 stdY = norm(r) / sqrt(df) ;
6 % Evaluate the covariance
7 Bp = pinv( B ) ;
8 Cov = stdY^2 * ( Bp * Bp' ) ;
```

Listing 4.4: Code required to compute the covariances.

8 Further improvements

Possible improvements of the algorithm and areas of future research are as follows:

³This is a common approach, since such solvers are available in most numerical computation packages.

1. The matrix $\mathbf{B}(\boldsymbol{\omega})$ has a special structure with a constant column of 1's. This column is invariant with respect to $\boldsymbol{\omega}$; consequently, orthogonal residualization can be applied to remove this portion from the matrix. This is done prior to initiating the nonlinear iteration, it reduces the dimension of $\mathbf{B}(\boldsymbol{\omega})$ from $n \times 3$ to $n \times 2$. This makes the computation of the pseudo-inverse $\mathbf{B}^+(\boldsymbol{\omega})$ numerically more efficient. This is important since this computation is performed in the iteration loop.
2. When \mathbf{B} has a large condition number, a regularized variable projection method [19] can be used to improve the results obtained.
3. The discontinuity in the cost function, $E(\boldsymbol{\omega})$, see Figure 1, suggests that an optimization based on a simplex method [20] may perform better than a gradient method.

9 Conclusions

This paper has presented a new derivation and implementation of the method of variable projection for the estimation of the four parameter sine wave model. The new derivations give new insights into the nature of the cost function and how it changes for low values of c_r . Additionally, this effect can be detected during evaluation by computing the condition number for the matrix of basis functions. The new set of equations also provides a means of calculating the covariance of the linear parameters; these can be used to compute confidence and prediction intervals. A thorough numerical testing also revealed that the new implementation is more stable and faster than past solutions.

Acknowledgements

This work was partially funded by:

1. The COMET program within the K2 Center “Integrated Computational Material, Process and Product Engineering (IC-MPPE)” (Project No 859480). This program is supported by the Austrian Federal Ministries for Transport, Innovation and Technology (BMVIT) and for Digital and Economic Affairs (BMDW), represented by the Austrian research funding association (FFG), and the federal states of Styria, Upper Austria and Tyrol.
2. The European Institute of Innovation and Technology (EIT), a body of the European Union which receives support from the European Union’s Horizon 2020 research and innovation programme. This was carried out under Framework Partnership Agreement No. 17031 (MaMMa - Maintained Mine & Machine).

The authors gratefully acknowledge this financial support.

Bibliography

- [1] A. Albarbar, S. Mekid, A. Starr, and R. Pietruszkiewicz, “Suitability of mems accelerometers for condition monitoring: An experimental study,” *Sensors*, vol. 8, no. 2, pp. 2192–2196, 2008.
- [2] M. Preeti, Koushik Guha, K. Baishnab, K. Dusarlapudi, and K. Narasimha Raju, “Low frequency mems accelerometers in health monitoring – a review based on material and design aspects,” *Materials Today: Proceedings*, vol. 18, pp. 2152 – 2157, 2019, 2nd International Conference on Applied Sciences and Technology (ICAST-2019): Material Science. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2214785319320310>
- [3] J. S. Lee, S. Choi, S. Kim, C. Park, and Y. G. Kim, “A mixed filtering approach for track condition monitoring using accelerometers on the axle box and bogie,” *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 3, pp. 749–758, 2012.
- [4] H. Wang, Z. Liu, A. Núñez, and R. Dollevoet, “Identification of the catenary structure wavelength using pantograph head acceleration measurements,” in *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2017, pp. 1–6.
- [5] IEEE, “IEEE standard for digitizing waveform recorders,” *IEEE Std 1057-2017 (Revision of IEEE Std 1057-2007)*, pp. 1–0, 2018.
- [6] K. Hejn and A. Pacut, “Effective resolution of analog to digital converters,” *IEEE Instrumentation Measurement Magazine*, vol. 6, no. 3, pp. 48–55, 2003.
- [7] A. Baccigalupi, M. D’Arco, and A. Liccardo, “Parameters and methods for adcs testing compliant with the guide to the expression of uncertainty in measurements,” *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 3, pp. 424–431, 2017.
- [8] R. Koyabagi and J. Pollard, “Piezoelectric accelerometer low-frequency response by signal insertion methods,” NIST Report: NBSIR 74-597, Tech. Rep., May 1975.
- [9] P. Arpaia and H. Schumny, “International standardization of adc-based measuring systems—state of the art,” *Computer Standards and Interfaces*, vol. 19, no. 3, pp. 173 – 188, 1998. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0920548998000154>
- [10] P. O’Leary and M. Harker, “Polynomial approximation: An alternative to windowing in fourier analysis,” in *2011 IEEE International Instrumentation and Measurement Technology Conference*, 2011, pp. 1–6.

- [11] M. Fonseca da Silva, P. M. Ramos, and A. Serra, "A new four parameter sine fitting technique," *Measurement*, vol. 35, no. 2, pp. 131 – 137, 2004, 2 special issues: Vibration measurement by Laser techniques Advances and Applications and 7th workshop on ADC modelling and testing. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0263224103000812>
- [12] I. Kollar and J. J. Blair, "Improved determination of the best fitting sine wave in adc testing," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 5, pp. 1978–1983, 2005.
- [13] K. Hejn and D. Morling, "A semi-fixed frequency method for evaluating the effective resolution of a/d converters," in *IEEE Instrumentation and Measurement Technology Conference*, 1991, pp. 51–54.
- [14] K. F. Chen, "Estimating parameters of a sine wave by separable nonlinear least squares fitting," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 12, pp. 3214–3217, 2010.
- [15] G. H. Golub and V. Pereyra, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 413–432, 1973.
- [16] G. Golub and V. Pereyra, "Separable nonlinear least squares: the variable projection method and its applications," *Inverse Problems*, vol. 19, pp. R1–R26(1), 01 2003.
- [17] G. Gowan and S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*, ser. Ohlin Lectures; 7. Springer New York, 1998.
- [18] D. O’Leary and B. Rust, "Variable projection for nonlinear least squares problems," *Computational Optimization and Applications*, vol. 54, no. 3, pp. 579–593, 2013.
- [19] L. Guo and Z. Fu, "Tikhonov regularized variable projection algorithms for separable nonlinear least squares problems," *Complexity, Hindawi*, vol. November, pp. 1–9, 2019.
- [20] H. Karloff, *The Simplex Algorithm: Chapter 2 in Linear Programming*. Birkhäuser Boston., 2009.

Decomposition of a Periodic Perturbed Signal with Unknown Perturbation Frequency by the Method of Variable Projection

Johannes Handler, Dimitar Ninevski and Paul O'Leary

Chair of Automation

Department of Product Engineering

University of Leoben

8700 Leoben, Austria

Email: automation@unileoben.ac.at

Abstract

This paper presents a new approach to separate a signal into its periodic and aperiodic components; whereby, the exact frequency of the periodic component is unknown. In other words, it is shown how to determine the underlying trend of a periodic perturbed signal and simultaneously identify the shape of the periodic perturbation. Therefore the signal is modeled by a nonlinear design matrix containing periodic basis functions, which depend on the unknown frequency, and aperiodic basis functions, more precisely discrete orthogonal polynomials (DOP). The nonlinear least squares problem of computing the model coefficients is solved by the method of variable projection. A well chosen partitioning of the design matrix enables an orthogonal residualization corresponding to a generalized Eckart-Young-Mirsky matrix approximation, which yields an efficient implementation of the variable projection method. This implementation is thoroughly tested using Monte Carlo simulations and the results are compared with those obtained by the classical implementation of the method of variable projection.

Keywords: Signal separation, Nonlinear least squares; Variable projection; Eckart-Young-Mirsky matrix approximation

1 Introduction

The need to separate a signal into its periodic and aperiodic components is a well known problem in many different fields of study, e.g. in medicine [1] or in economics, where it is usually known as seasonal adjustment [2] [3]. Another area where this problem arises is in engineering. Here a process which inherits a rotational motion, naturally can become the source of a periodic pattern, e.g. the dynamic bulging process during steel casting [4] or the flow ripples in fluid gear pumps [5]. An eccentric mounting of rotational components leads to the so called runout, which is a common problem of gear wheels [6] or in milling [7] [8]. It also occurs when measuring the roundness and sphericity of a sphere [9]. For the active control of the lubrication gap of plain bearings [10] [11] it is necessary to directly measure the lubrication gap or to measure the eccentricity of the journal with respect to the bearing bushing. However, if the journal itself has a deviation of roundness the eccentricity measurement will yield the actual eccentricity superimposed with a periodic pattern originating from the deviation of roundness. So, in order to control the underlying process or quantity, in this case the lubrication gap, it is necessary to separate the periodic perturbation from the signal of interest.

For clarification it has to be mentioned, that the task of decomposing a signal into its periodic and aperiodic components is also looked at in speech and audio processing [12] [13] when analyzing and describing the characteristics of voice sources. However, in audio processing a stochastic (or noise-like) signal is meant when talking about the aperiodic portion. This is not the task dealt with here. This work proposes a method to separate a periodic perturbation superimposed on an aperiodic (trend-like) signal.

With regard to a process that inherits a rotational motion, depending on the method of measuring the rotational speed and the possible occurrence of slip effects the measured rotational speed might not be exactly the same as the actual rotational speed. Consequently, this method considers the frequency of the periodic perturbation as unknown. Due to this unknown frequency it has to be assumed that the periodic pattern might not be perfectly periodic in the measurement interval. Hence, direct Fourier methods such as the FFT will be subject to Gibbs error [14]. This error makes it difficult to estimate the exact frequency of the periodic component. Therefore, the method of variable projection [15–18] is used to overcome this problem.

The main contribution of this paper is a thorough matrix algebraic derivation of a new approach to separate a signal into its periodic and aperiodic components. The proposed method is based on a nonlinear least squares approximation of the measurement signal. The approximation model consists of a combination of periodic and aperiodic basis functions. Once the model coefficients are determined the periodic basis functions are used to reconstruct the periodic signal component and the aperiodic basis functions are used to compute the aperiodic signal portion. A special partitioning of the resulting matrix of basis functions yields a numerically efficient solution to the signal separation task.

2 Separation Model

The goal is to decompose a signal y consisting of n measurements, so $\mathbf{y} = [y_1, \dots, y_n]^T$ into its periodic and aperiodic components and the measurements are taken at the time points $\mathbf{t} = [t_1, \dots, t_n]^T$. In this manner both \mathbf{y} and \mathbf{t} are $n \times 1$ column vectors.

2.1 Periodic Component

Since the periodic component \mathbf{y}_p of the signal, also a $n \times 1$ column vector, can have a shape different to a perfect sine wave, it will be approximated by a sum of sine and cosine functions and their k harmonics with the fundamental frequency ω_0 ,

$$\begin{aligned} \mathbf{y}_p = & a_1 \cos(\omega_0 \mathbf{t}) + a_2 \cos(2\omega_0 \mathbf{t}) + \dots + a_k \cos(k\omega_0 \mathbf{t}) + \\ & + b_1 \sin(\omega_0 \mathbf{t}) + b_2 \sin(2\omega_0 \mathbf{t}) + \dots + b_k \sin(k\omega_0 \mathbf{t}) . \end{aligned} \quad (4.26)$$

This can be written as a linear combination of basis functions,

$$\begin{aligned} \mathbf{y}_p = & [\cos(\omega_0 \mathbf{t}) \dots \cos(k\omega_0 \mathbf{t})] [a_1 \dots a_k]^T + \\ & + [\sin(\omega_0 \mathbf{t}) \dots \sin(k\omega_0 \mathbf{t})] [b_1 \dots b_k]^T . \end{aligned} \quad (4.27)$$

Since the fundamental frequency ω_0 is unknown, this model can be seen as a linear combination of nonlinear functions. With the following definitions,

$$\begin{aligned} \mathbf{H}(\omega_0) = & [\cos(\omega_0 \mathbf{t}) \dots \cos(k\omega_0 \mathbf{t}) \quad \sin(\omega_0 \mathbf{t}) \dots \sin(k\omega_0 \mathbf{t})] \\ \boldsymbol{\sigma} = & [a_1 \dots a_k \quad b_1 \dots b_k]^T \end{aligned} \quad (4.28)$$

it can be written short,

$$\mathbf{y}_p = \mathbf{H}(\omega_0) \boldsymbol{\sigma} , \quad (4.29)$$

where $\mathbf{H}(\omega_0)$ is a matrix containing the harmonic basis functions and $\boldsymbol{\sigma}$ is the vector of coefficients. For all further derivations it is assumed, that the number of measurements $n \geq 2k$, this ensures that $\mathbf{H}(\omega_0)$ has full column rank.

2.2 Aperiodic Component

The aperiodic component \mathbf{y}_t , also called the trend component here, is modeled by a polynomial,

$$\mathbf{y}_t = \mathbf{B} \boldsymbol{\alpha} , \quad (4.30)$$

where \mathbf{B} is the matrix with discrete orthogonal polynomial (DOP) basis functions [19] as its columns and $\boldsymbol{\alpha}$ are the coefficients of the polynomial.

2.3 Combined Basis

Finally the complete model \mathbf{y}_m is established by adding the periodic component and the trend component,

$$\mathbf{y}_m = \mathbf{y}_t + \mathbf{y}_p = \mathbf{B}\boldsymbol{\alpha} + \mathbf{H}(\omega_0)\boldsymbol{\sigma} . \quad (4.31)$$

It becomes apparent that by concatenating the two matrices \mathbf{B} and $\mathbf{H}(\omega_0)$ this expression can be written as,

$$\mathbf{y}_m = [\mathbf{B} \ \mathbf{H}(\omega_0)] \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\sigma} \end{bmatrix} . \quad (4.32)$$

The following definitions,

$$\mathbf{M}(\omega_0) = [\mathbf{B} \ \mathbf{H}(\omega_0)] \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\sigma} \end{bmatrix} \quad (4.33)$$

lead the matrix vector notation for the final model,

$$\mathbf{y}_m = \mathbf{M}(\omega_0)\mathbf{c} , \quad (4.34)$$

where $\mathbf{M}(\omega_0)$ is the so called design matrix and \mathbf{c} is the vector of linear coefficients.

3 Separation Procedure

The idea behind the method of variable projection is to define the model as a linear combination of nonlinear functions. This is exactly how the proposed model here (Equ. 4.53) is chosen. The least squares approximation of the data by the model is defined by minimizing the cost function,

$$E(\omega_0) = \|\mathbf{r}\|^2 \quad (4.35)$$

where \mathbf{r} is the residual vector computed by,

$$\mathbf{r} = \mathbf{y} - \mathbf{y}_m . \quad (4.36)$$

As shown in [18] this leads to the so called *variable projection functional* (VPF) being,

$$E(\omega_0) = \|\{\mathbf{I} - \mathbf{M}(\omega_0)\mathbf{M}(\omega_0)^+\}\mathbf{y}\|^2 , \quad (4.37)$$

where $\mathbf{M}(\omega_0)^+$ denotes the Moore-Penrose pseudo inverse. This nonlinear least squares problem can be solved for ω_0 with a standard nonlinear iteration process. However, this would lead to the necessity of computing the pseudo inverse of $\mathbf{M}(\omega_0)$ in every iteration step, which is numerically expensive. As it can be seen in Equ. (4.33) not every column of $\mathbf{M}(\omega_0)$ depends on ω_0 . This fact

can be used to derive a more efficient solution to the problem. So the cost function (Equ. (4.35)) can be expanded to,

$$\begin{aligned} E(\omega_0) &= \|\mathbf{r}\|^2 = \mathbf{r}^T \mathbf{r} = (\mathbf{y} - \mathbf{y}_m)^T (\mathbf{y} - \mathbf{y}_m) = \\ &= \mathbf{y}^T \mathbf{y} - 2\mathbf{y}_m^T \mathbf{y} + \mathbf{y}_m^T \mathbf{y}_m . \end{aligned} \quad (4.38)$$

By using the relationship from Equ. (4.31) this can also be written as,

$$\begin{aligned} E(\omega_0) &= \mathbf{y}^T \mathbf{y} - 2\alpha^T \mathbf{B}^T \mathbf{y} - 2\sigma^T \mathbf{H}(\omega_0)^T \mathbf{y} + 2\alpha^T \mathbf{B}^T \mathbf{H}(\omega_0) \sigma + \\ &+ \alpha^T \mathbf{B}^T \mathbf{B} \alpha + \sigma^T \mathbf{H}(\omega_0)^T \mathbf{H}(\omega_0) \sigma . \end{aligned} \quad (4.39)$$

In order to get the linear coefficients α and σ which minimize the cost function $E(\omega_0)$ the partial derivatives have to be zero,

$$\frac{\partial E}{\partial \alpha} = -2\mathbf{B}^T \mathbf{y} + 2\mathbf{B}^T \mathbf{H}(\omega_0) \sigma + 2\mathbf{B}^T \mathbf{B} \alpha = 0 \quad (4.40)$$

$$\frac{\partial E}{\partial \sigma} = -2\mathbf{H}(\omega_0)^T \mathbf{y} + 2\mathbf{H}(\omega_0)^T \mathbf{B} \alpha + 2\mathbf{H}(\omega_0)^T \mathbf{H}(\omega_0) \sigma = 0. \quad (4.41)$$

Equ. (4.41) can be solved for σ yielding,

$$\begin{aligned} \sigma &= (\mathbf{H}(\omega_0)^T \mathbf{H}(\omega_0))^{-1} (\mathbf{H}(\omega_0)^T \mathbf{y} - \mathbf{H}(\omega_0)^T \mathbf{B} \alpha) \\ &= (\mathbf{H}(\omega_0)^T \mathbf{H}(\omega_0))^{-1} \mathbf{H}(\omega_0)^T (\mathbf{y} - \mathbf{B} \alpha) . \end{aligned} \quad (4.42)$$

By using the Moore-Penrose pseudo inverse Equ. (4.42) states,

$$\sigma = \mathbf{H}(\omega_0)^+ (\mathbf{y} - \mathbf{B} \alpha) . \quad (4.43)$$

This equation shows that σ and thereby the periodic signal portion depends on the difference of $\mathbf{y} - \mathbf{B} \alpha = \mathbf{y} - \mathbf{y}_t$. In order for this difference to contain as little aperiodic component as possible, α is defined as,

$$\alpha = \mathbf{B}^T \mathbf{y} . \quad (4.44)$$

Substituting this expression into Equ. (4.43) finally yields,

$$\sigma = \mathbf{H}(\omega_0)^+ (\mathbf{I} - \mathbf{B} \mathbf{B}^T) \mathbf{y} . \quad (4.45)$$

Now it becomes apparent that, by defining Equ. (4.44) the signal \mathbf{y} is first approximated solely by the aperiodic model and the residual of this approximation is then used to approximate the periodic portion. This approach corresponds to a generalized Eckart-Young-Mirsky matrix approximation [20], which will further lead to a numerical efficient computation of the VPF. It has to be mentioned though, that the consequence of defining α the way it is done in Equ. (4.44) is, that Equ. (4.40) does not hold anymore. Therefore, the result of this approximation is in general only

close to the optimum and not the true optimum. However, with the relation from Equ. (4.45), the periodic model \mathbf{y}_p from Equ. (4.29) can be expressed as,

$$\mathbf{y}_p = \mathbf{H}(\omega_0)\mathbf{H}(\omega_0)^+ (\mathbf{I} - \mathbf{B}\mathbf{B}^T) \mathbf{y} , \quad (4.46)$$

and by substituting Equ. (4.44) into Equ. (4.30) the trend component \mathbf{y}_t is defined as,

$$\mathbf{y}_t = \mathbf{B}\mathbf{B}^T \mathbf{y} . \quad (4.47)$$

Adding this two results the final model equation yields,

$$\mathbf{y}_m = \mathbf{B}^T \mathbf{B} \mathbf{y} + \mathbf{H}(\omega_0)\mathbf{H}(\omega_0)^+ (\mathbf{I} - \mathbf{B}\mathbf{B}^T) \mathbf{y} . \quad (4.48)$$

By using this model equation, the residual vector (Equ. (4.36)) can be written as,

$$\mathbf{r} = (\mathbf{I} - \mathbf{H}(\omega_0)\mathbf{H}(\omega_0)^+) (\mathbf{I} - \mathbf{B}\mathbf{B}^T) \mathbf{y} . \quad (4.49)$$

Defining $\mathbf{P}_H(\omega_0) \triangleq \mathbf{H}(\omega_0)\mathbf{H}(\omega_0)^+$, which is the projection matrix associated with $\mathbf{H}(\omega_0)$ and $\mathbf{P}_B \triangleq \mathbf{B}\mathbf{B}^T$, which is the projection matrix associated with \mathbf{B} , gives

$$\mathbf{r} = (\mathbf{I} - \mathbf{P}_H(\omega_0)) (\mathbf{I} - \mathbf{P}_B) \mathbf{y} . \quad (4.50)$$

This equation shows that \mathbf{y} is first projected onto the subspace orthogonal to the one spanned by \mathbf{B} , denoted by $\mathbf{P}_B^\perp = \mathbf{I} - \mathbf{P}_B$ and then onto the orthogonal complement of the subspace spanned by $\mathbf{H}(\omega_0)$, denoted by $\mathbf{P}_H^\perp(\omega_0) = \mathbf{I} - \mathbf{P}_H(\omega_0)$. As stated before, this corresponds to a generalized Eckart-Young-Mirsky matrix approximation. Using these definitions the VPF can now be written as,

$$E(\omega_0) = \|\mathbf{P}_H^\perp(\omega_0)\mathbf{P}_B^\perp \mathbf{y}\|^2 . \quad (4.51)$$

If this result is compared to Equ. (4.37) the numerical advantage of this derivation becomes apparent. Because rather than having to compute the pseudo inverse of $\mathbf{M}(\omega_0)$ in every iteration step, now only the pseudo inverse of $\mathbf{H}(\omega_0)$ has to be computed during the nonlinear iteration process. Once the nonlinear parameter ω_0 is determined Equ. (4.46) yields an estimation for the periodic signal portion \mathbf{y}_p and the trend component \mathbf{y}_t can be approximated with Equ. (4.47). The linear coefficients of the model are computed by equations (4.44) and (4.45).

4 Numerical Testing

The proposed method is tested on two sets of synthetic data (see Fig. 1 and Fig. 2), each of which consists of $n = 1000$ samples. The first test signal consists of an aperiodic component created by a DOP of degree four and for the second test signal a DOP of degree 13 is used. The periodic component is the same for both signals. The final test signal is the sum of those two components superimposed with i.i.d. Gaussian noise with a standard deviation of $\sigma = 0.02$.

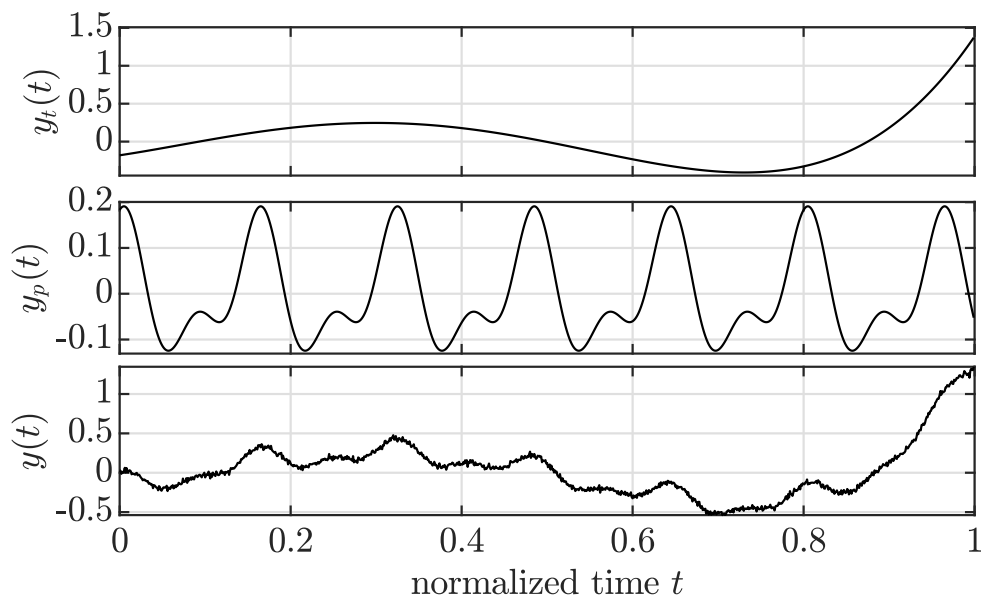


Figure 1: First synthetic test signal, consisting of the periodic component $y_p(t)$ here with the number of cycles in record $c_r = 6.25$ and the trend component $y_t(t)$ created by a DOP of degree four. The final test signal $y(t)$ is the sum of those two components superimposed with i.i.d. Gaussian noise with a standard deviation of $\sigma = 0.02$.

The proposed method is tested on both synthetic signals via Monte Carlo simulation with respect to its execution time and its frequency estimation accuracy. Therefore, each run of the Monte Carlo simulation consists of $m = 500$ independent simulations, which again were carried out for different numbers of cycles in record c_r of the periodic perturbation. These simulations are used to compare the proposed method, based on the Eckart-Young-Mirsky matrix approximation (Equ. (4.51)) with the classical variable projection approach (Equ. (4.37)). For solving the non-linear optimization problem, the MATLAB solver *lsqnonlin()*, based on a trust-region reflective algorithm [21], is used.

4.1 Estimated frequency

The deviation of the estimated frequency to the actual frequency for the first test signal is presented in Fig. 3 and Fig. 4. It can be seen that the estimated frequency deviates strongly from the actual frequency for a low number of cycles in record. Therefore, the aperiodic component can also be thought of as a periodic signal with close to one cycle in record. Now, if the periodic perturbation is also close to only one cycle in record the algorithm fails to separate the two signals correctly. However, Fig. 4 shows, that for $c_r \geq 2.75$ the classical approach already provides an accurate estimation of the frequency. Another interesting fact that becomes apparent from Fig. 4 is, that the frequency estimation of the proposed algorithm gets more accurate the larger c_r is. This can be

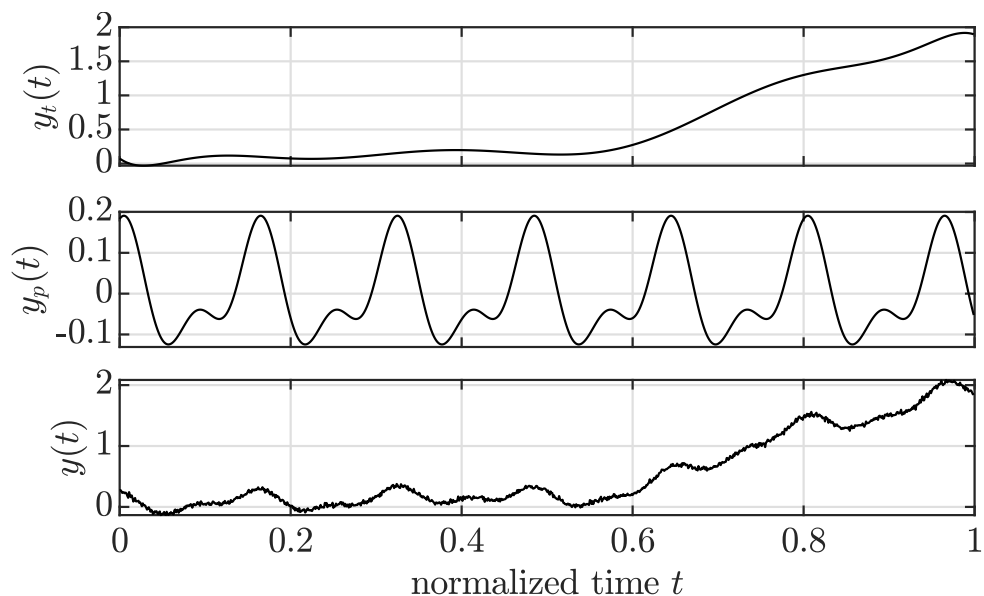


Figure 2: Second synthetic test signal, consisting of the periodic component $y_p(t)$ here with the number of cycles in record $c_r = 6.25$ and the trend component $y_t(t)$ created by a DOP of degree 13. The final test signal $y(t)$ is the sum of those two components superimposed with i.i.d. Gaussian noise with a standard deviation of $\sigma = 0.02$.

explained by looking at Equ. (4.51) more closely, because what the Eckart-Young-Mirsky matrix approximation does is a kind of cascaded least squares. The signal is first approximated by the DOP in a least squares sense and the residual of this fit is then approximated by the harmonic basis functions. So the more periodicity the signal has the harder it is for the DOP to approximate it. Hence, the remaining residual contains more of the true periodic portion which enables a more accurate estimation of the frequency. The deviation of the estimated frequency for the second test signal is presented in Fig. 5. The results show similar behavior to the first test signal.

4.2 Computational time

The result of the Monte Carlo simulations regarding the elapsed CPU time is shown in Fig. 6 for the first test signal and in Fig. 7 for the second test signal respectively. These simulations show that for the first test signal there is no noticeable difference in the computation time for either method. It follows that the Eckart-Young-Mirsky matrix approximation provides no significant numerical advantage for the DOP of degree 4. However, if the aperiodic component is approximated by a higher order DOP, as it is the case for the second test signal (see Fig. 7) the advantage concerning the numerical efficiency of the proposed algorithm becomes apparent.

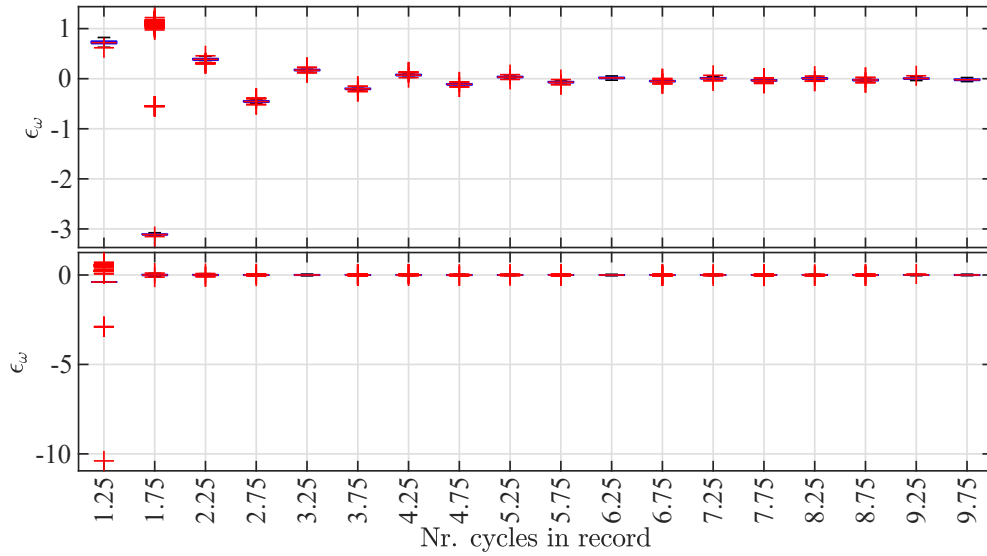


Figure 3: Box plots of the estimated frequency deviation for the first test signal, which has an aperiodic component modeled by a DOP of degree four, obtained from Monte Carlo simulations: (top) the presented approach based on the Eckart-Young-Mirsky matrix approximation (Equ. (4.51)), (bottom) the classical variable projection approach (Equ. (4.37)).

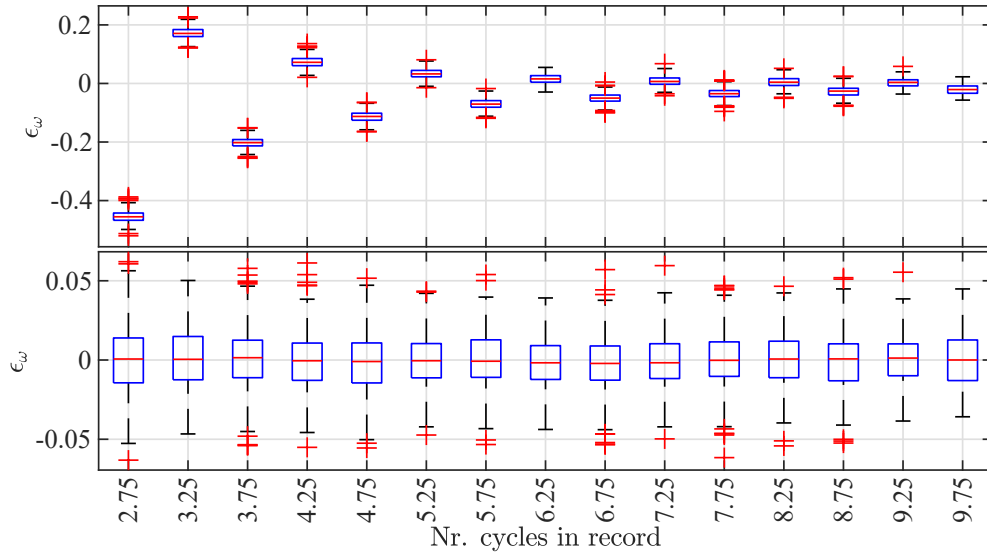


Figure 4: Detailed view of Fig. 3 for $2.75 \leq c_r \leq 9.75$

4.3 Separation task

The result of the separation task for the first test signal is shown in Fig. 8, where $c_r = 6.25$ is chosen for the illustration. It can be seen, that the proposed algorithm is able to solve the separation task

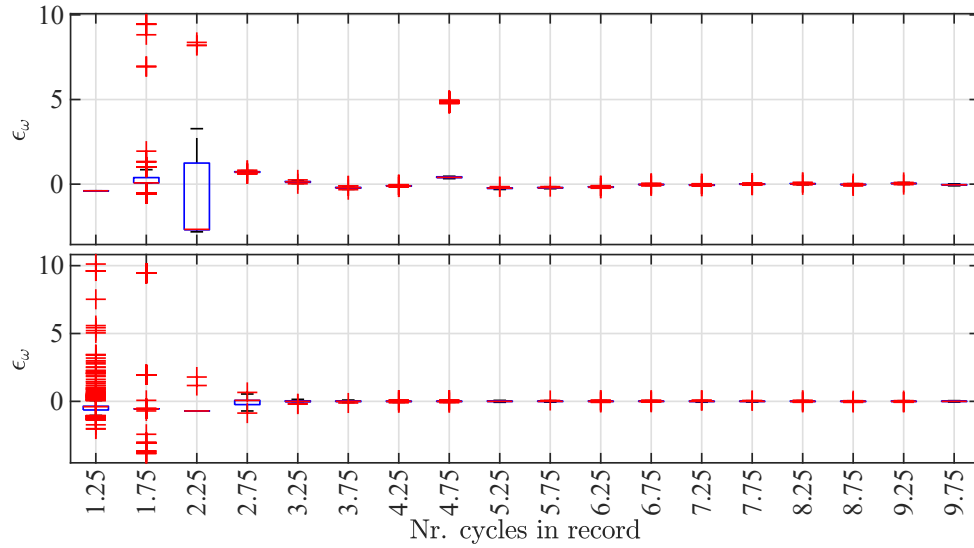


Figure 5: Box plots of the the estimated frequency deviation for the second test signal, which has an aperiodic component modeled by a DOP of degree 13, obtained from Monte Carlo simulations: (top) the presented approach based on the Eckart-Young-Mirsky matrix approximation (Equ. (4.51)), (bottom) the classical variable projection approach (Equ. (4.37)).

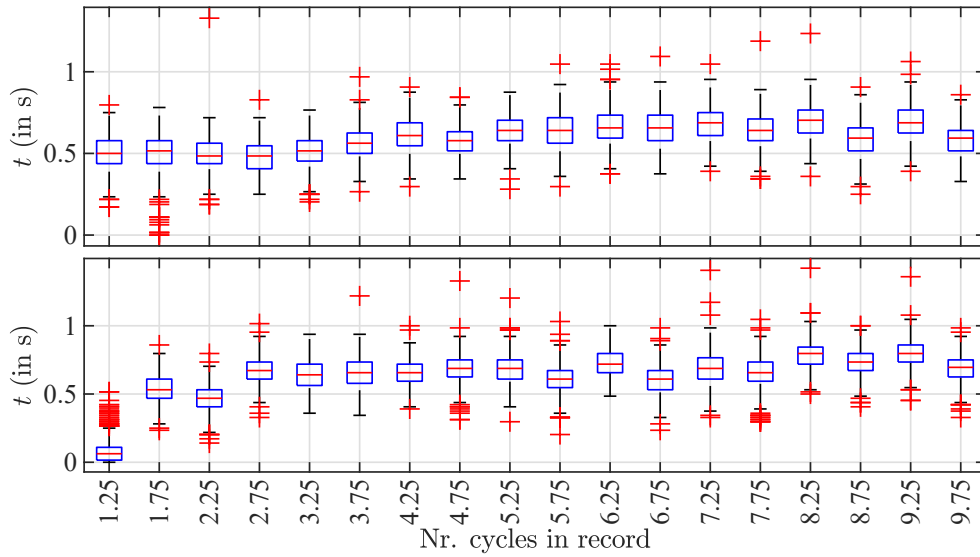


Figure 6: Box plots of the execution times for the first test signal, which has an aperiodic component modeled by a DOP of degree four, obtained from Monte Carlo simulations: (top) the presented approach based on the Eckart-Young-Mirsky matrix approximation (Equ. (4.51)), (bottom) the classical variable projection approach (Equ. (4.37)).

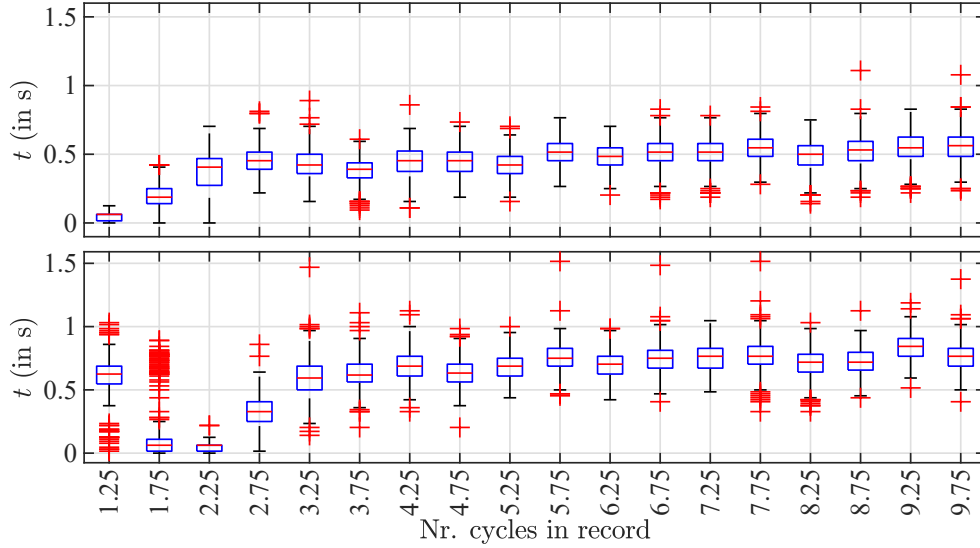


Figure 7: Box plots of the execution times for the second test signal, which has an aperiodic component modeled by a DOP of degree 13, obtained from Monte Carlo simulations: (top) the presented approach based on the Eckart-Young-Mirsky matrix approximation (Equ. (4.51)), (bottom) the classical variable projection approach (Equ. (4.37)).

very well. The result for the second test signal, also for $c_r = 6.25$ is presented in Fig. 9. The previous mentioned problem of the cascaded approximation becomes evident here. Due to the high order of the DOP, parts of the periodic portion are already included by the aperiodic model, which leads to the deviations in the periodic component, shown in the middle plot. Again, as stated before, this effect becomes smaller the more cycles in record the signal has.

5 Conclusion

This paper presented a new approach for periodic/aperiodic signal separation where the exact frequency of the periodic component is unknown. This is used to determine the underlying trend of a periodic perturbed signal or for identifying the shape of a periodic pattern. The task is formulated as a nonlinear least squares problem which is solved by the method of variable projection. Due to a well chosen partitioning of the design matrix, a numerically efficient implementation of the variable projection method was derived, based on an Eckart-Young-Mirsky matrix approximation. Numerical tests via Monte Carlo simulations, lead to the insight that the advantage regarding the computation time becomes more relevant the higher the polynomial degree of the trend estimation model is. Furthermore, the estimation accuracy of the unknown frequency is better the more cycles of the periodic pattern the investigated signal contains. These two results lead to the conclusion that the proposed method has its strength when investigating signals containing a high number of

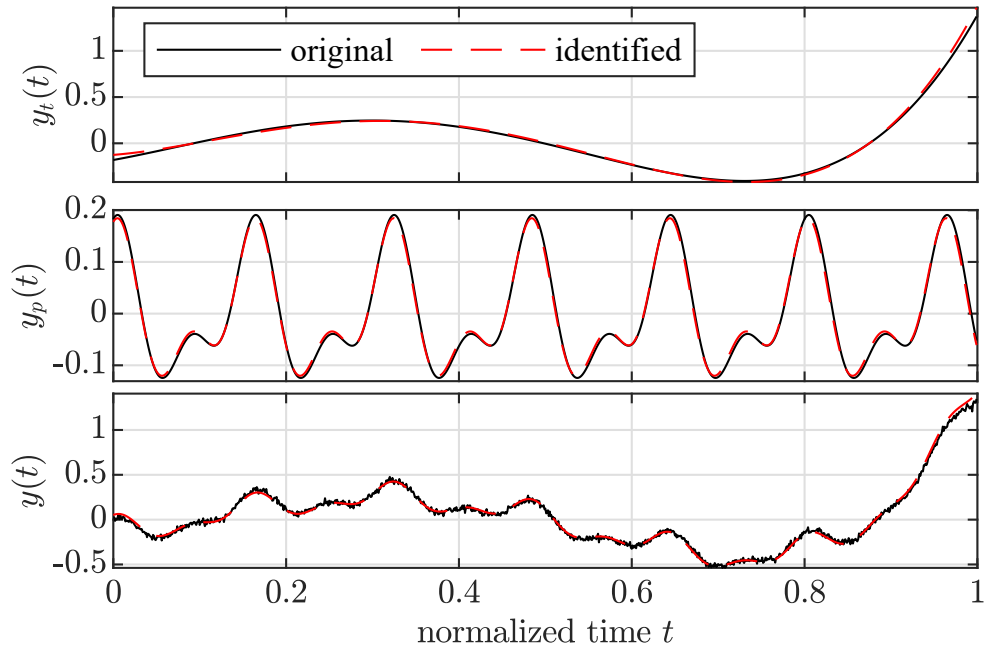


Figure 8: First synthetic test signal, consisting of the periodic component $y_p(t)$ here with the number of cycles in record being 6.25 and the trend component $y_t(t)$ created by a polynomial of degree four. The final test signal $y(t)$ is the sum of those two components superimposed with i.i.d. Gaussian noise.

cycles of the periodic pattern and a complicated trend component which needs to be modeled by a high order polynomial. These two things are usually present in the case of signals acquired over a longer period of time with respect to perturbation period.

Acknowledgements

This work was partially funded by:

1. The COMET program within the K2 Center “Integrated Computational Material, Process and Product Engineering (IC-MPPE)” (Project No 859480). This program is supported by the Austrian Federal Ministries for Transport, Innovation and Technology (BMVIT) and for Digital and Economic Affairs (BMDW), represented by the Austrian research funding association (FFG), and the federal states of Styria, Upper Austria and Tyrol.
2. The European Institute of Innovation and Technology (EIT), a body of the European Union which receives support from the European Union’s Horizon 2020 research and innovation programme. This was carried out under Framework Partnership Agreement No. 17031 (MaMMa - Maintained Mine & Machine).

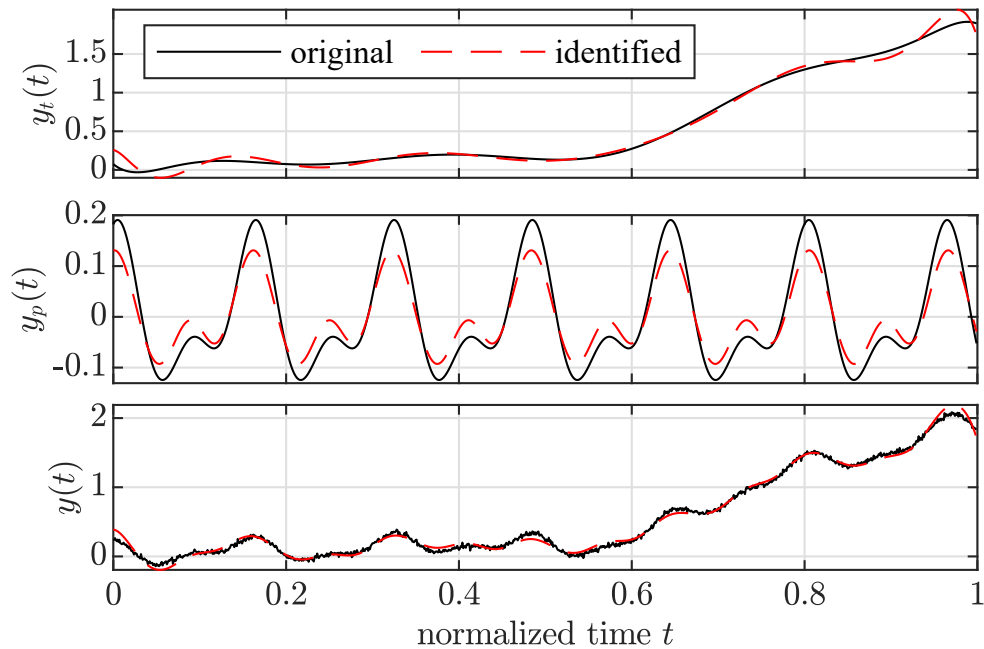


Figure 9: Second synthetic test signal, consisting of the periodic component $y_p(t)$ here with the number of cycles in record being 6.25 and the trend component $y_t(t)$ created by a polynomial of degree 13. The final test signal $y(t)$ is the sum of those two components superimposed with i.i.d. Gaussian noise.

The authors gratefully acknowledge this financial support.

Bibliography

- [1] L. Weizman, K. L. Miller, Y. C. Eldar, O. Maayan, and M. Chiew, “Pear: Periodic and aperiodic signal separation for fast fmri,” in *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2017, pp. 505–508.
- [2] D. A. Pierce, “A survey of recent developments in seasonal adjustment,” *The American Statistician*, vol. 34, no. 3, pp. 125–134, 1980.
- [3] D. Pollock, “Enhanced methods of seasonal adjustment,” *Econometrics*, vol. 9, 2021.
- [4] C. Furtmüller, L. delRe, H. Bramerdorfer, and Mörwald K., “Periodic disturbance suppression in a steel plant with unstable internal feedback and delay,” in *International Conference on Technology and Automation*, 2005.
- [5] K. Park, M. Chang, and D. Jeon, “Precise flowrate control of fluid gear pumps in automated painting systems using a repetitive controller,” *Applied Sciences*, vol. 9, no. 16, 2019.

- [6] Y.-C. Pei, H.-L. Xie, and Q.-C. Tan, "A non-contact high precision measuring method for the radial runout of cylindrical gear tooth profile," *Mechanical Systems and Signal Processing*, vol. 138, 2020.
- [7] W. Chen, L. Lu, W. Xie, D. Huo, and K. Yang, "A new surface topography-based method to quantify axial error of high speed milling cutters," *Journal of Manufacturing Science and Engineering*, vol. 140, no. 11, 2018.
- [8] X. Zhang, X. Pan, G. Wang, and D. Zhou, "Tool runout and single-edge cutting in micro-milling," *International Journal of Advanced Manufacturing Technology*, vol. 96, pp. 821–832, 2018.
- [9] X. S. Zhao, T. Sun, Y. D. Yan, Z. Q. Li, and S. Dong, "Measurement of roundness and sphericity of the micro sphere based on atomic force microscope," *Key Engineering Materials*, vol. 315-316, pp. 796–799, 2006.
- [10] R. Kurth, R. Tehel, T. Päßler, M. Putz, K. Wehmeyer, C. Kraft, and H. Schwarze, "Forming 4.0: Smart machine components applied as a hybrid plain bearing and a tool clamping system," *Procedia Manufacturing*, vol. 27, pp. 65–71, 2019.
- [11] W. U. Rehman, G. Jiang, Y. Luo, Y. Wang, W. Khan, S. U. Rehman, and N. Iqbal, "Control of active lubrication for hydrostatic journal bearing by monitoring bearing clearance," *Advances in Mechanical Engineering*, vol. 10, no. 4, 2018.
- [12] K. Aczél and I. Vajk, "Separation of periodic and aperiodic sound components by employing frequency estimation," in *16th European Signal Processing Conference*, 2008.
- [13] K. Vijayan, J. K. Dhiman, and C. S. Seelamantula, "Time-frequency coherence for periodic-aperiodic decomposition of speech signals," in *Interspeech 2017*, 2017, pp. 329–333.
- [14] A. J. Jerri, *The Gibbs Phenomenon in Fourier Analysis, Splines and Wavelet Approximations*. Boston, MA: Springer US, 1998.
- [15] G. H. Golub and V. Pereyra, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 413–432, 1973.
- [16] G. Golub and V. Pereyra, "Separable nonlinear least squares: the variable projection method and its applications," *Inverse Problems*, vol. 19, no. 2, pp. R1–R26, 2003.
- [17] D. P. O’Leary and B. W. Rust, "Variable projection for nonlinear least squares problems," *Computational Optimization and Applications*, vol. 54, pp. 579–593, 2013.

- [18] P. O’Leary and D. Ninevski, “Estimating parameters of a sine wave by the method of variable projection,” in *2021 IEEE International Instrumentation and Measurement Technology Conference*, 2021.
- [19] P. O’Leary and M. Harker, “An algebraic framework for discrete basis functions in computer vision,” in *Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.
- [20] G. Golub, A. Hoffman, and Steward G., “A generalization of the eckart-young-mirsky matrix approximation theorem,” *Linear Algebra and Its Applications*, vol. 88, pp. 317–327, 1987.
- [21] J. J. Moré and D. C. Sorensen, “Computing a trust region step,” *SIAM Journal on Scientific and Statistical Computing*, vol. 4, no. 3, pp. 553–572, 1983.

Real-Time Identification of Periodic Signals Using the Recursive Variable Projection Algorithm

Johannes Handler⁰⁰⁰⁰⁻⁰⁰⁰³⁻⁴⁶⁴⁰⁻²⁴⁶², Dimitar Ninevski⁰⁰⁰⁰⁻⁰⁰⁰³⁻⁰¹⁰¹⁻⁸⁶⁸⁶,
Mathias Rollett⁰⁰⁰⁰⁻⁰⁰⁰²⁻³⁸¹¹⁻¹³¹⁸ and Paul O’Leary⁰⁰⁰⁰⁻⁰⁰⁰²⁻¹³⁶⁷⁻⁸²⁷⁰

Chair of Automation
Department of Product Engineering
University of Leoben
8700 Leoben, Austria
Email: automation@unileoben.ac.at

Abstract

This paper presents a real-time parameter identification algorithm for periodic signals, based on the recursive variable projection (RVP) algorithm. The recursive implementation enables the tracking of time-varying parameters. The signal model is linear with respect to the amplitude parameters while being nonlinear with respect to the phase and frequency. This feature motivates the use of a variable projection based approach. Its performance is tested using Monte Carlo simulations and the results are compared with those obtained by a multiobjective Gauss-Newton (MGN) algorithm. Furthermore, the RVP algorithm is applied to measurement data acquired by a MEMS accelerometer and it is demonstrated that it can successfully track time-varying linear and nonlinear parameters.

Keywords: Frequency estimation, Parameter identification, Recursive least squares, Periodic signal, Variable projection method

1 Introduction

In the very recent past, there has been a rise in the implementation of low cost high performance MEMS accelerometers in condition monitoring [1, 2]. Vibration measurement, in particular, is essential for the detection and diagnosis of anomalous machine behavior [3–5]. Accelerometers

are well suited for vibration measurements because the sensor signal can be directly linked to the machine motion. This requires a method for the reliable identification of time-varying vibration signal parameters, i.e. frequency, amplitude, phase and DC component. The second requirement is a low computational complexity to enable embedded condition monitoring and inline process control. Therefore, this paper introduces a new approach to real-time identification of time-varying periodic signal parameters.

Considering the estimation of the sine wave parameters, defined in the IEEE-standard 1057 [6], i.e. frequency, phase, amplitude and DC component, the accurate estimation of the frequency received most attention in history. That is, because estimation of the other parameters is relatively easy, once the frequency is known. Some well known frequency estimation methods are: zero crossing detection [7], FFT based methods [8], recursive least squares (RLS) [9], gradient based methods [10] and machine learning approaches [11].

There are also advances in methods, estimating not only the frequency but all of the sine wave parameters. In [12], an adaptive estimation scheme based on a multiobjective Gauss-Newton (MGN) algorithm is proposed. Different algorithms, based on gradient descent, have been introduced by Xu et al. [13, 14]. Another approach, that has been presented recently, is based on the method of variable projection [15]. In general, the method of variable projection yields an efficient solution for separable nonlinear least squares problem [16, 17]. This is exactly the case being addressed here, since the sine wave model is linear with respect to the amplitude parameters, while being nonlinear with respect to the phase and frequency. However, the implementation presented in [15], and the method of variable projection in general, aims at batch data analysis, making it suboptimal for on-line parameter estimation. Fortunately, around the same time, Gan et al. developed a Recursive Variable Projection (RVP) algorithm [18], however primarily motivated by problems arising in machine learning.

The emergence of a general recursive formulation for the method of variable projection on the one hand, and the promising results shown in [15] on the other hand, motivate the here proposed approach. That is, a sine wave parameter estimation algorithm based on the RVP algorithm. The recursive formulation not only facilitates the application to on-line parameter estimation but also enables the tracking of time-varying parameters.

The value of this paper lies in: a detailed discussion of the RVP algorithm and its adaption to parameter estimation of periodic signals, including valid simplifications of the original RVP algorithm. The performance is compared to the MGN algorithm [12], since it is the closest and most relevant algorithm to the work presented here. That is because the MGN algorithm is able to estimate not only the frequency but also all the sine wave parameters. The proposed approach performs significantly better, especially for SNR below 30 dB, see Section 3 for more details. Furthermore, the successful implementation for MEMS accelerometer signals, having time-varying parameters is also presented.

2 Recursive Variable Projection Algorithm

The Recursive Variable Projection (RVP) algorithm introduced in [18] is a special extension of the Recursive Levenberg-Marquardt (RLM) algorithm [19], for separable nonlinear models.

In general, the method of variable projection [16, 17] is used to eliminate the linear parameters in a first step and compute solely the nonlinear parameters using a numerical iteration process, e.g. Levenberg-Marquardt [20,21]. Once the nonlinear parameters are determined, the linear parameter can be computed using linear least-squares.

The algorithm presented here follows the same concept. First, the variable projection approach is used to reduce the cost function only being dependent on the nonlinear parameters. Considering this cost function, a recursive update rule, for the nonlinear parameter only, is derived. In each recursion the nonlinear parameter is updated and subsequently the linear parameters are estimated using a Recursive Least-Squares (RLS) algorithm. In the following, the main steps presented in [18] will be revisited and the adaptations to the here presented problem discussed.

2.1 Signal Model

The widely used amplitude and phase form of the sine wave model is given as,

$$g(t) = a \sin(\omega t + \phi) + d, \quad (4.52)$$

which is nonlinear with respect to the frequency ω and the phase ϕ . However, here the four parameter form, also used in the IEEE-standard 1057 [6], is considered,

$$g(\theta; t) = a_c \cos(\omega t) + a_s \sin(\omega t) + d. \quad (4.53)$$

This has the advantage of reducing the number of nonlinear parameters right away. The four parameters which now need to be identified are the frequency ω , the amplitudes a_c and a_s and the DC component d . These four parameters are collected in the vector θ . Using this model to estimate a measurement signal $y(t)$, the residual is defined as,

$$r(\theta; t) = y(t) - g(\theta; t). \quad (4.54)$$

2.2 Recursive Levenberg-Marquardt Algorithm

First of all, the cost function $\varepsilon_t(\theta)$ is chosen to consider an exponential forgetting mechanism [22],

$$\varepsilon_t(\theta) = \frac{1}{2} \sum_{k=1}^t \lambda^{t-k} r^2(\theta; k) \quad (4.55)$$

where the forgetting factor λ is usually set as $0.95 \leq \lambda \leq 1$.

The gradient vector and the Hessian matrix of the function $\varepsilon_t(\theta)$ are denoted by $\Delta \varepsilon_t(\theta)$ and $H_{\varepsilon_t}(\theta)$ respectively. The derivation of the RLM algorithm is based on a Taylor approximation of

$\varepsilon_t(\theta)$. Let $\theta(t-1)$ be the estimation of the model parameter at time $t-1$. The quadratic Taylor approximation of $\varepsilon_t(\theta)$ around $\theta(t-1)$ is,

$$\begin{aligned} \varepsilon_t(\theta) &\approx \varepsilon_t(\theta(t-1)) + \Delta^T \varepsilon_t(\theta(t-1)) [\theta - \theta(t-1)] \\ &\quad + \frac{1}{2} [\theta - \theta(t-1)]^T \mathbf{H}_{\varepsilon_t}(\theta(t-1)) [\theta - \theta(t-1)]. \end{aligned} \quad (4.56)$$

The optimal θ at time t needs to minimize $\varepsilon_t(\theta)$, i.e. $\frac{d\varepsilon_t(\theta)}{d\theta} = 0$. Hence, differentiating (4.56) with respect to θ and equating to zero yields,

$$\theta(t) = \theta(t-1) - \mathbf{H}_{\varepsilon_t}^{-1}(\theta(t-1)) \Delta \varepsilon_t(\theta(t-1)), \quad (4.57)$$

which is a recursive formulation for computing the parameters θ . Now a recursive update rule for $\mathbf{H}_{\varepsilon_t}$ would also be desirable. Therefore, the gradient vector and Hessian matrix of the cost function (4.55) are necessary. The gradient vector is:

$$\Delta \varepsilon_t(\theta) = \lambda \Delta \varepsilon_{t-1}(\theta) - \eta(\theta; t) r(\theta; t), \quad (4.58)$$

where,

$$\Delta r(\theta; k) = -\frac{dg(\theta; k)}{d\theta} = -\eta(\theta; t). \quad (4.59)$$

The Hessian matrix on the other hand is:

$$\mathbf{H}_{\varepsilon_t}(\theta) = \lambda \mathbf{H}_{\varepsilon_{t-1}}(\theta) + \eta(\theta; t) \eta(\theta; t)^T - \mathbf{H}_r(\theta; t) r(\theta; t). \quad (4.60)$$

Assuming that $\theta(t)$ is within a small neighborhood of $\theta(t-1)$ the term $\mathbf{H}_r(\theta; t) r(\theta; t)$ can be neglected, leading to

$$\mathbf{H}_{\varepsilon_t}(\theta) = \lambda \mathbf{H}_{\varepsilon_{t-1}}(\theta) + \eta(\theta; t) \eta(\theta; t)^T. \quad (4.61)$$

Substituting (4.58) and (4.61) into (4.57) leads to the recursive relationship,

$$\begin{aligned} \theta(t) &= \theta(t-1) + \\ &\quad \mathbf{H}_{\varepsilon_t}^{-1}(\theta(t-1)) \eta(\theta(t-1); t) r(\theta(t-1); t) \end{aligned} \quad (4.62)$$

$$\begin{aligned} \mathbf{H}_{\varepsilon_t}(\theta(t-1)) &= \lambda \mathbf{H}_{\varepsilon_{t-1}}(\theta(t-1)) + \\ &\quad \eta(\theta(t-1); t) \eta(\theta(t-1); t)^T \end{aligned} \quad (4.63)$$

To realize a type of *Levenberg-Marquardt* algorithm, $\mathbf{R}(t) = (1 - \lambda) \mathbf{H}_{\varepsilon_t}(\theta(t-1)) + \delta I$ is chosen for the approximation of the Hessian in (4.63) and its inverse in (4.62) is just scaled [19], which finally leads to,

$$\begin{aligned} \theta(t) &= \theta(t-1) + (1 - \lambda) \mathbf{R}(t)^{-1} \eta(\theta(t-1); t) r(\theta(t-1); t) \\ \mathbf{R}(t) &= \lambda \mathbf{R}(t-1) + \\ &\quad (1 - \lambda) \left[\eta(\theta(t-1); t) \eta(\theta(t-1); t)^T + \delta I \right] \end{aligned} \quad (4.64)$$

with δ being a small positive number.

Now, in [18] and [19] these equations are further simplified using a matrix inversion lemma, since the inversion of $\mathbf{R}(t)$ in (4.64) requires a computational complexity of $\mathcal{O}(q^3)$ where $q = \dim(\theta)$. However, this is not necessary here, because by using the method of variable projection q is reduced to one.

2.3 Method of Variable Projection

The method of variable projection [16, 17], is suited for models defined as a linear combination of nonlinear functions; this is exactly how the sine wave model $g(\theta; t)$ is formulated in (4.53),

$$g(\mathbf{c}; \omega; t) = [\cos(\omega t) \quad \sin(\omega t) \quad 1] \begin{bmatrix} a_c \\ a_s \\ d \end{bmatrix} \quad (4.65)$$

where \mathbf{c} is the vector containing the linear parameter a_c , a_s and d .

Consider that the dynamical window data with length p are the measurements \mathbf{y}_p corresponding to the time vector $\mathbf{t}_p = [t_k, t_{k-1}, \dots, t_{k-p+1}]$, where t_k is the current time; i.e. the dynamical window data covers the last p measurement values. This results in the matrix of basis functions being,

$$\mathbf{B}_p(\omega) = [\mathbf{b}_{p1}(\omega), \mathbf{b}_{p2}(\omega), \mathbf{b}_{p3}] \quad (4.66)$$

where,

$$\mathbf{b}_{p1}(\omega) \triangleq \cos(\omega \mathbf{t}_p) \quad \mathbf{b}_{p2}(\omega) \triangleq \sin(\omega \mathbf{t}_p) \quad \mathbf{b}_{p3} \triangleq \mathbf{1}. \quad (4.67)$$

The notation $\mathbf{B}_p(\omega)$ indicates that the matrix \mathbf{B}_p is dependent on the nonlinear parameter ω . With the linear coefficient vector

$$\mathbf{c} = [a_c, a_s, d]^T \quad (4.68)$$

the model, considering the dynamical window data, can be written as the matrix vector equation,

$$g(\mathbf{c}; \omega; \mathbf{t}_p) = \mathbf{B}_p(\omega) \mathbf{c}. \quad (4.69)$$

For a given ω , a least squares estimate for \mathbf{c} is obtained from,

$$\mathbf{c} = \mathbf{B}_p^+(\omega) \mathbf{y}_p \quad (4.70)$$

whereby, $\mathbf{B}_p^+(\omega)$ denotes the Moore-Penrose pseudo inverse of $\mathbf{B}_p(\omega)$. Now by defining the vector of basis functions for the current time t_k as,

$$\mathbf{b}_k(\omega) = [\cos(\omega t_k), \sin(\omega t_k), 1]^T \quad (4.71)$$

the model equation evaluated at t_k , yields

$$g(\omega; t_k) = \mathbf{b}_k^T(\omega) \mathbf{B}_p^+(\omega) \mathbf{y}_p. \quad (4.72)$$

Furthermore, the residual r defined in (4.54) becomes,

$$r(\omega; t_k) = y_k - \mathbf{b}_k^T(\omega) \mathbf{B}_p^+(\omega) \mathbf{y}_p. \quad (4.73)$$

Note that, both the model equation (4.72), as well as the residual (4.73) only depend on ω . By using (4.72) and (4.73) instead of (4.53) and (4.54), in (4.64), $R(t)$ reduces to a scalar. So, with the method of variable projection approach, the recursive parameter identification in (4.64) now only computes the nonlinear parameter ω . However, this is not a problem, since the linear parameter can be determined using a RLS algorithm [23]. This will be discussed in more detail in the next section.

In order to compute $\eta(\omega; t_k)$ and in succession, the parameter update (4.64), the derivative of $g(\omega; t_k)$ w.r.t. ω needs to be determined, see (4.59). That again requires computing the differential of the pseudo inverse matrix $\mathbf{B}_p^+(\omega)$. This is done by using the rule [16, 24],

$$\partial \mathbf{B}^+ = -\mathbf{B}^+ \partial [\mathbf{B}] \mathbf{B}^+ + (\mathbf{B}^T \mathbf{B})^{-1} \partial [\mathbf{B}]^T (\mathbf{I} - \mathbf{B} \mathbf{B}^+). \quad (4.74)$$

Now, applying the product rule, the derivative of the model equation $g(\omega; t_k)$ is calculated as,

$$\eta(\omega; t_k) = \frac{d\mathbf{b}_k^T(\omega)}{d\omega} \mathbf{B}_p^+(\omega) \mathbf{y}_p + \mathbf{b}_k^T(\omega) \frac{d\mathbf{B}_p^+(\omega)}{d\omega} \mathbf{y}_p. \quad (4.75)$$

Defining,

$$\mathbf{Q}_p(\omega) = \frac{d\mathbf{B}_p(\omega)}{d\omega} = [-\mathbf{t}_p \circ \sin(\omega \mathbf{t}_p), \mathbf{t}_p \circ \cos(\omega \mathbf{t}_p), \mathbf{0}] \quad (4.76)$$

where \circ denotes the Hadamard product and

$$\mathbf{q}_k(\omega) = \frac{d\mathbf{b}_k(\omega)}{d\omega} = [-t_k \sin(\omega t_k), t_k \cos(\omega t_k), 0]^T, \quad (4.77)$$

equation (4.75) can finally be written as,

$$\eta(\omega; t_k) = \left[\mathbf{q}_k^T \mathbf{B}_p^+ - \mathbf{b}_k^T \mathbf{B}_p^+ \mathbf{Q}_p \mathbf{B}_p^+ + \mathbf{b}_k^T (\mathbf{B}_p^T \mathbf{B}_p)^{-1} \mathbf{Q}_p^T (\mathbf{I} - \mathbf{B}_p \mathbf{B}_p^+) \right] \mathbf{y}_p. \quad (4.78)$$

With this, the parameter update for ω finally becomes,

$$\begin{aligned} \omega(t_k) &= \omega(t_{k-1}) + \frac{1 - \lambda}{R(t_k)} \eta(\omega(t_{k-1}); t_k) r(\omega(t_{k-1}); t_k) \\ R(t_k) &= \lambda R(t_{k-1}) + (1 - \lambda) [\eta(\omega(t_{k-1}); t_k)^2 + \delta]. \end{aligned} \quad (4.79)$$

The initial value $R(t_0)$ should be chosen small if there is little confidence in $\omega(t_0)$. Moreover, to improve the rate of convergence, λ is also updated recursively with $\lambda(t_k) = \lambda_r \lambda(t_{k-1}) + (1 - \lambda_r)$, where typically $0.995 < \lambda_r < 1$ and $0.95 \leq \lambda(t_0) \leq 1$.

The computational complexity of the algorithm is determined by the numerical effort needed to evaluate (4.79) or (4.73) and (4.78), respectively. Evaluating these equations requires the derivation of $\mathbf{B}_p^+(\boldsymbol{\omega})$, which has a computational complexity of $\mathcal{O}(3p^2)$. Considering now (4.79) the overall computational complexity of the algorithm is $\mathcal{O}(8p^2 + 43p)$. So, p will be the determining factor regarding the computational complexity of the recursive parameter update. Additionally, p influences the stability of the frequency estimate, i.e. the larger p is, the more stable the estimate becomes. Therefore, by selecting p , a trade-off between computational complexity and estimation stability has to be made.

2.4 Linear Parameter Update

Once $\boldsymbol{\omega}$ is updated the second part of the RVP algorithm is to update the linear parameters using a RLS algorithm. In accordance with [18], a RLS algorithm with exponential forgetting factor [25] is used, where the exponential forgetting factor λ is chosen to be the same as in the nonlinear parameter update (4.79). However, if it is to be expected that the linear and nonlinear parameters vary at different rates, two separate forgetting factors should be chosen. The linear parameter update then follows as,

$$\begin{aligned} \mathbf{c}(t_k) &= \mathbf{c}(t_{k-1}) + \mathbf{K}(t_k)r(\mathbf{c}(t_{k-1}); \boldsymbol{\omega}(t_k); t_k) \\ \mathbf{K}(t_k) &= \frac{\mathbf{P}(t_{k-1})\mathbf{b}_k(\boldsymbol{\omega}(t_k))}{\lambda + \mathbf{b}_k^T(\boldsymbol{\omega}(t_k))\mathbf{P}(t_{k-1})\mathbf{b}_k(\boldsymbol{\omega}(t_k))} \\ \mathbf{P}(t_k) &= \frac{1}{\lambda} [I - \mathbf{K}(t_k)\mathbf{b}_k^T(\boldsymbol{\omega}(t_k))] \mathbf{P}(t_{k-1}) \end{aligned} \quad (4.80)$$

where $\mathbf{P}(t_k)$ is the covariance matrix of the estimate $\mathbf{c}(t_k)$ and $\mathbf{K}(t_k)$ is a gain vector. Therefore, $\mathbf{P}(t_0)$ should be chosen with large positive elements if there is little confidence in $\mathbf{c}(t_0)$; e.g. $\mathbf{P}(t_0) = 10^4 \mathbf{I}$.

2.5 Complete Algorithm

The presented computation steps lead to the final recursive parameter identification algorithm as presented in Algorithm 1.

3 Numerical Testing

The RVP algorithm is compared to the MGN algorithm [12] with respect to accuracy and numerical stability. Note, that the MGN algorithm is not able to identify a DC component. Therefore, the following parameters are selected to generate the synthetic test signal: $\boldsymbol{\omega} = 100\pi$, $a_c = 0.732$, $a_s = 0.682$ and $d = 0$. Note, that this synthetic signal is similar to the one used in [12] and it consists of $n = 200$ samples with the sampling frequency $f_s = 1.6$ kHz. Now Monte Carlo simulations,

Algorithm 1 RVP algorithm for sine wave parameter identification

Input: initial values $\omega(t_0)$, $c(t_0)$, $R(t_0)$ and $P(t_0)$; the forgetting-factor $\lambda(t_0)$ and the rate λ_r ; Levenberg-Marquardt damping factor δ ;

Output: identified parameter $\omega(t_n)$ and $c(t_n)$;

- 1: **for** $k = 1$ to n **do**
 - 2: Compute the derivative of the model equation using (4.78);
 - 3: Update the nonlinear parameter $\omega(t_k)$ using (4.79);
 - 4: Based on the updated $\omega(t_k)$, compute the linear parameters $c(t_k)$ using the RLS algorithm (4.80);
 - 5: **end for**
-

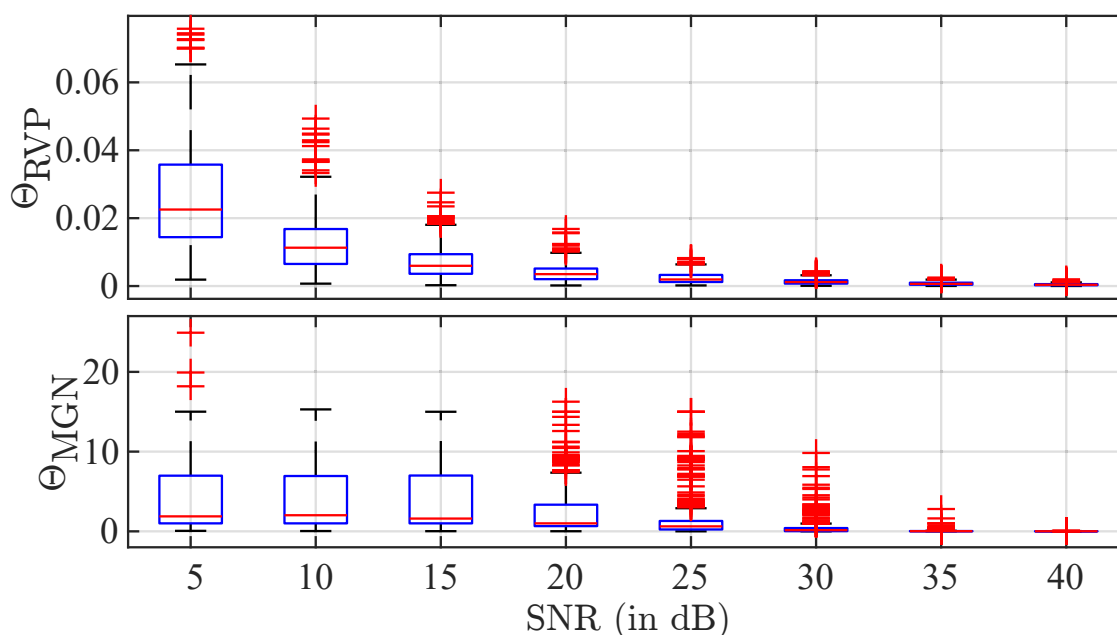


Figure 1: Box plots of of the parameter estimation error $\Theta = \|\hat{\theta} - \theta\|$ obtained from Monte Carlo simulations with $m = 500$ repetitions: (top) proposed RVP algorithm, (bottom) MGN algorithm [12]; note, that simulations where the algorithm failed to converge are not considered here. The signals with different SNR were all generated with $n = 200$ samples, and the parameters $\omega = 100\pi$, $a_c = 0.732$, $a_s = 0.682$ and $d = 0$.

where each run consist of $m = 500$ independent simulations, are performed for $k = 8$ different signal-to-noise ratios (SNR). The comparison of the parameter estimation error $\Theta = \|\hat{\theta} - \theta\|$, after the $n = 200$ recursions, for the different SNR levels is shown in Figure 1. It can be seen that the RVP algorithm is about two orders of magnitude more accurate than the MGN algorithm. Since in some cases the MGN algorithm became unstable and had to be terminated before reaching $n = 200$, only successful runs were considered for creating the box plots. The rate of successful

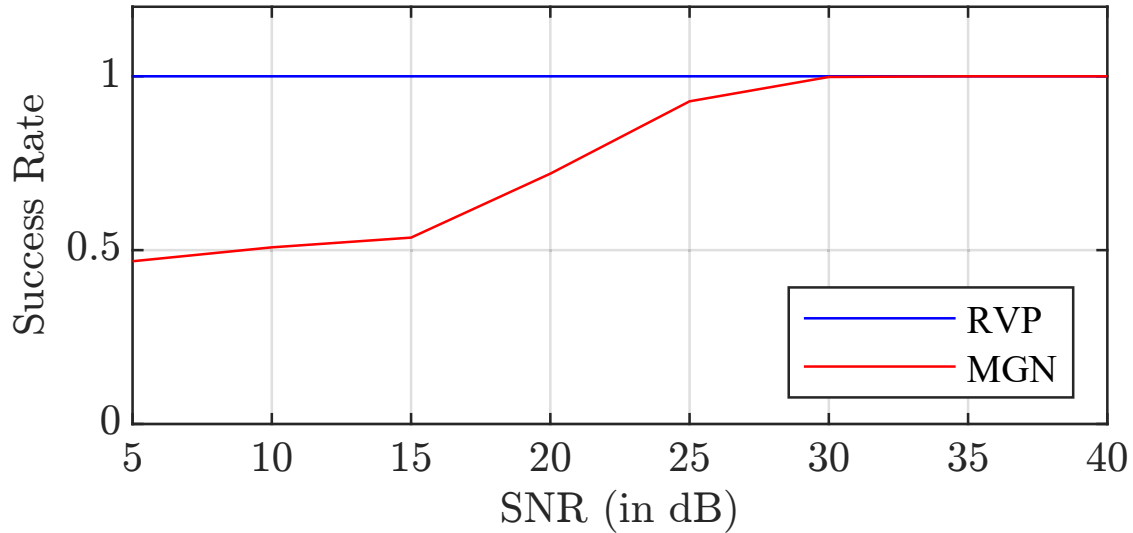


Figure 2: Comparison of the RVP and MGN algorithm with respect to their success rate depending on the SNR. Success rate is defined as the number of simulations for which the algorithm converged divided by the total number of simulations.

runs for the different SNR levels is presented in Figure 2. This shows that the RVP algorithm not only has a significant higher accuracy but also is more robust, especially for low SNR.

4 Experimental Verification

The proposed algorithm was also applied to vibrational measurement data with time-varying parameters, acquired by a MEMS accelerometer. An electrodynamic shaker generates the vibration and a 10 degree of freedom (DOF) smart sensor system, introduced in [26], was used to record the vibration with a sampling-rate of 4096 Hz. A reference-sensor which is directly fixed to the moving part of the shaker was used to verify the vibrational signal. See Figure 3 for the complete experimental setup. Note, that in the following the MGN algorithm is not considered for comparison due to multiple reasons. On the one hand it does not consider a DC component and on the other hand the measurement noise is too large for it to converge.

4.1 Signal with Phase Shift

The first case considers a measurement signal with a change in the phase of the signal at $t = 3.6$ s. Note, that this signal was generated by deleting a sequence of samples from a stationary vibration measurement. With respect to condition monitoring this would indicate slipping effects in the system. A phase change only effects the linear parameter a_c and a_s . The ability of the RLS algorithm to track time-varying parameters depends on $P(t)$. If $P(t) \rightarrow 0$ the parameter adaptability

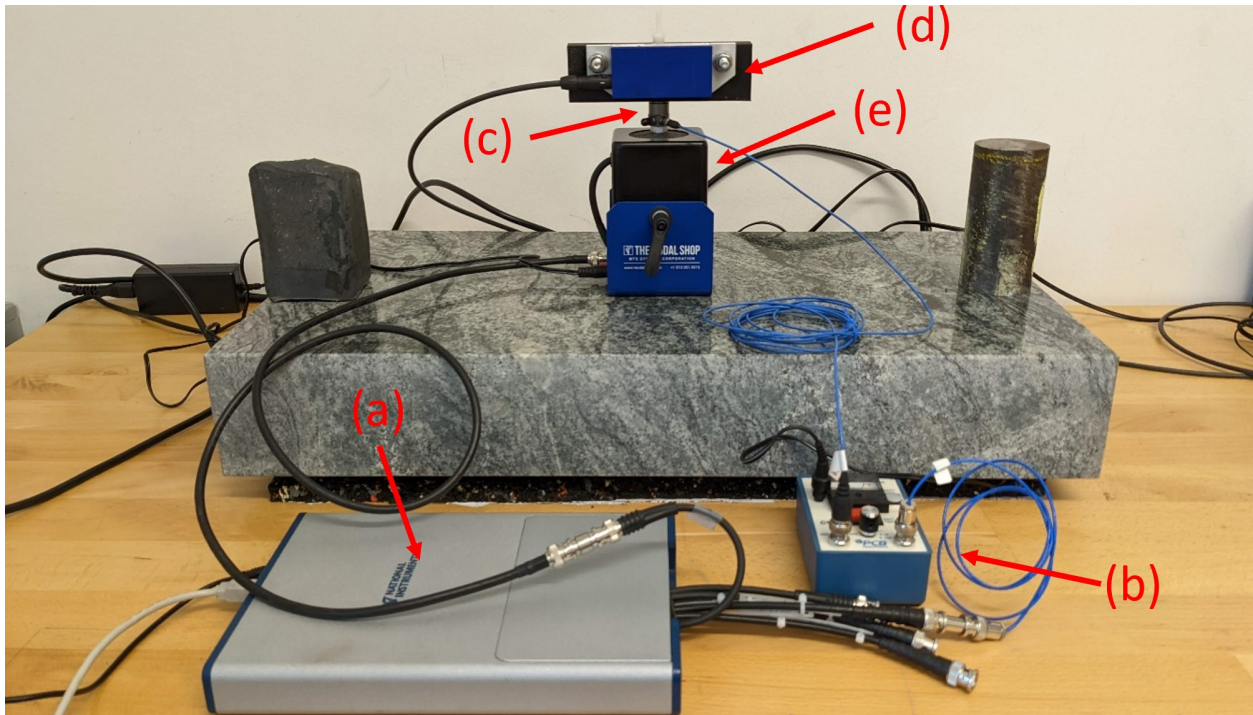


Figure 3: Laboratory test setup for the generation and acquisition of the test signals: (a) National Instruments data acquisition USB box, (b) signal conditioner, (c) 1-axis acceleration reference-sensor, (d) 10 DOF smart sensor system including a 3-axis accelerometer, (e) electrodynamic shaker.

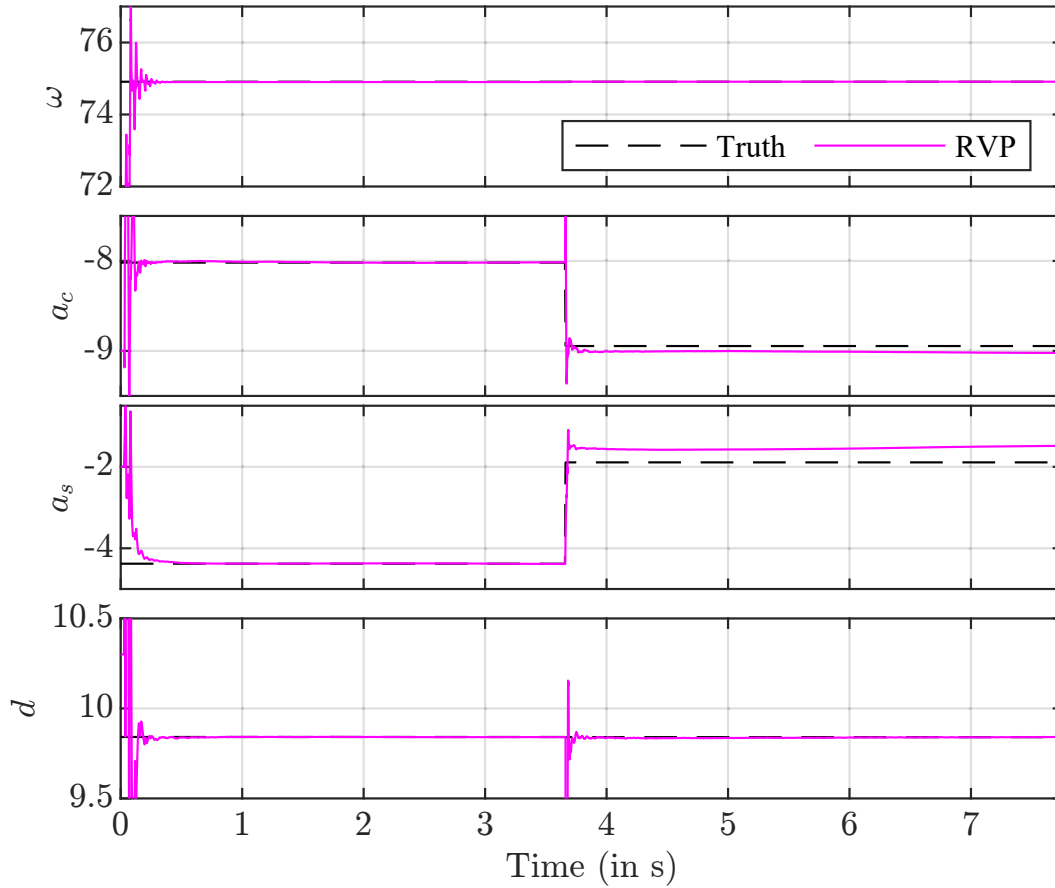


Figure 4: Convergence of the parameter estimate for a measurement signal with a step change in the linear parameters a_c and a_s , using the RVP algorithm. After the covariance resetting at $t = 3.6$ s it takes about one cycle to converge to the new parameter values.

is lost. One way to avoid that is, to set an upper limit for $\lambda(t)$, to insure $\lambda(t) < 1$ [25]. Another way, especially suited to track step changes, is covariance resetting [27]. This method is used here; once $|r(t)|$ exceeds a certain threshold, $P(t)$ is reset to its initial value. In Figure 4 the parameter estimate is given as it changes over time. It can be seen, that it takes about 0.2 seconds for the parameters to converge to the true value. The initial value of the parameters was set to 95% of the true value and the dynamical window length $p = 120$, which corresponds to approximately one third of a cycle. The true parameter values are the ones used to generate the input signal of the electrodynamic shaker and they were also verified with an offline identification method [15]. Figure 4 further shows, that the covariance resetting happens right after the parameter change. Afterwards the parameter estimate for a_c and a_s rapidly converges to the true value again. For this time sequence, the measurement signal and the model residual are shown in Figure 5. One can see, that the measurement signal is being well approximated. The fast reconvergence to the new parameter becomes also visible in the residual, since it returns to being predominantly random in

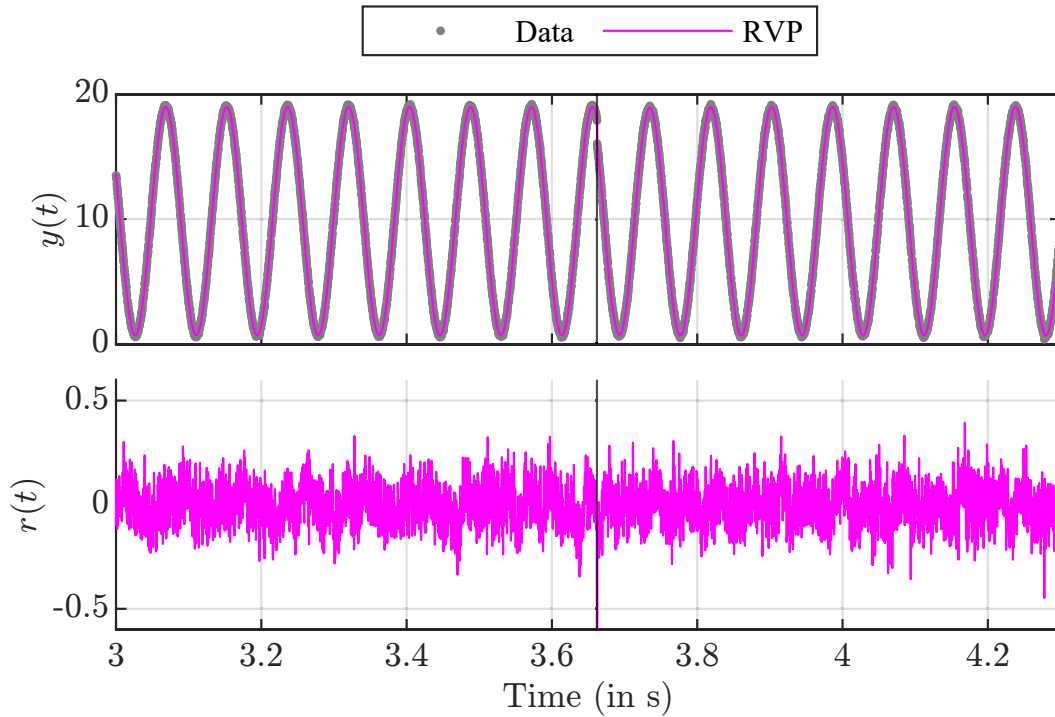


Figure 5: Measurement data with a change in the phase used to demonstrate the tracking ability of the RVP algorithm. (top) The measurement data y with a change in the phase at $t = 3.6$ s and the signal model determined by the RVP algorithm. (bottom) Residual r of the signal model.

nature right after the parameter change. This clearly shows the ability of the algorithm to successfully track step changes in the linear parameter.

4.2 Signal with Frequency Step

The second case additionally considers a step change in the frequency and the amplitude as well. In the condition monitoring context, this can indicate a velocity change in a power system. As in the previous example, the initial value of the parameters was set to 95% of the true value and the dynamical window length $p = 120$. In addition to the resetting of $P(t)$, here $R(t)$ is also reinitialized once $|r(t)|$ exceeds a certain threshold. This enables a fast adaptation of both, the nonlinear and linear parameters.

Figure 6 shows that the presented algorithm is in the position to successfully identify and adapt to these abrupt changing parameters. The actual signal model and residual can be seen in Figure 7. Note that only a short time window around the parameter change is presented there.

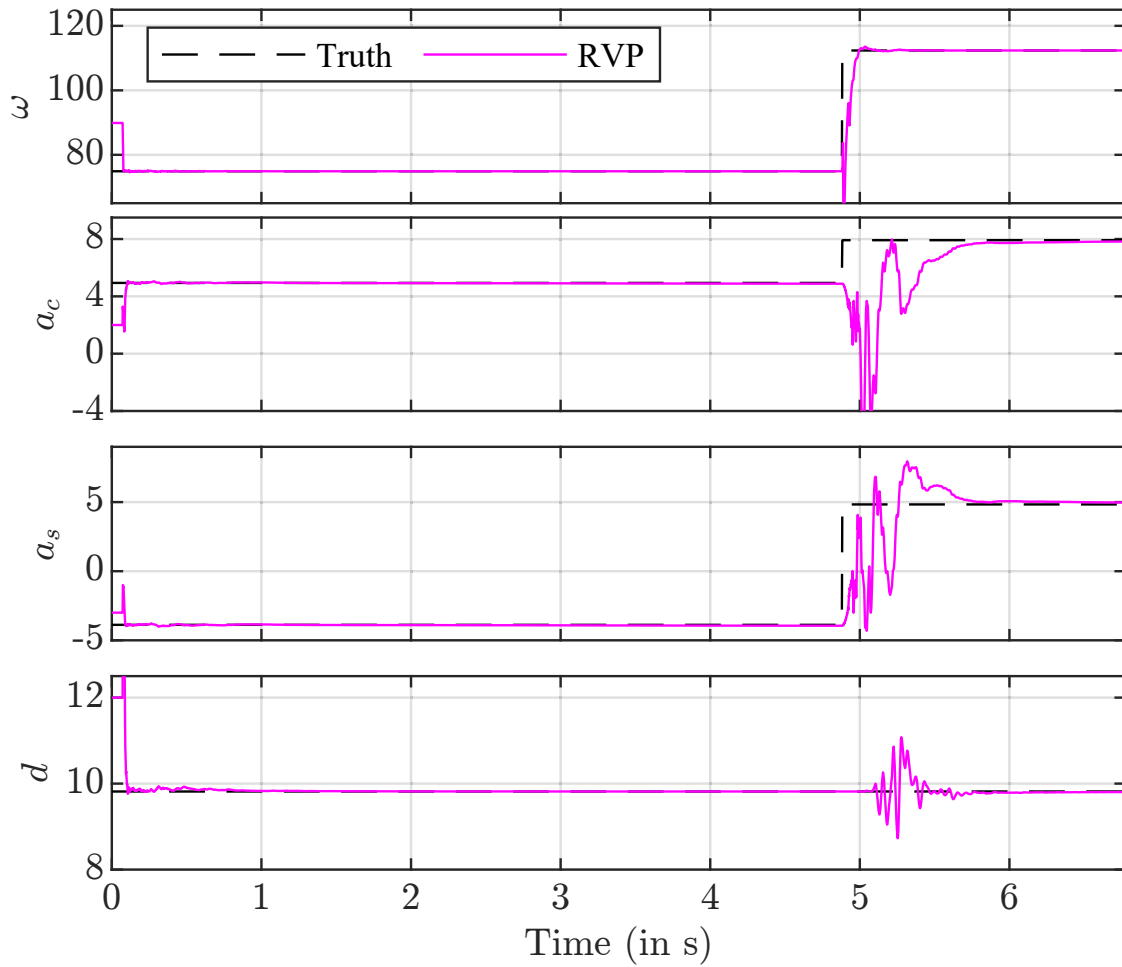


Figure 6: Convergence of the parameter estimate for a measurement signal with a step change in the linear parameters a_c and a_s as well as the nonlinear parameter ω , using the RVP algorithm. After the covariance resetting at $t = 4.9$ s it takes about one second for all parameters to converge to the new values.

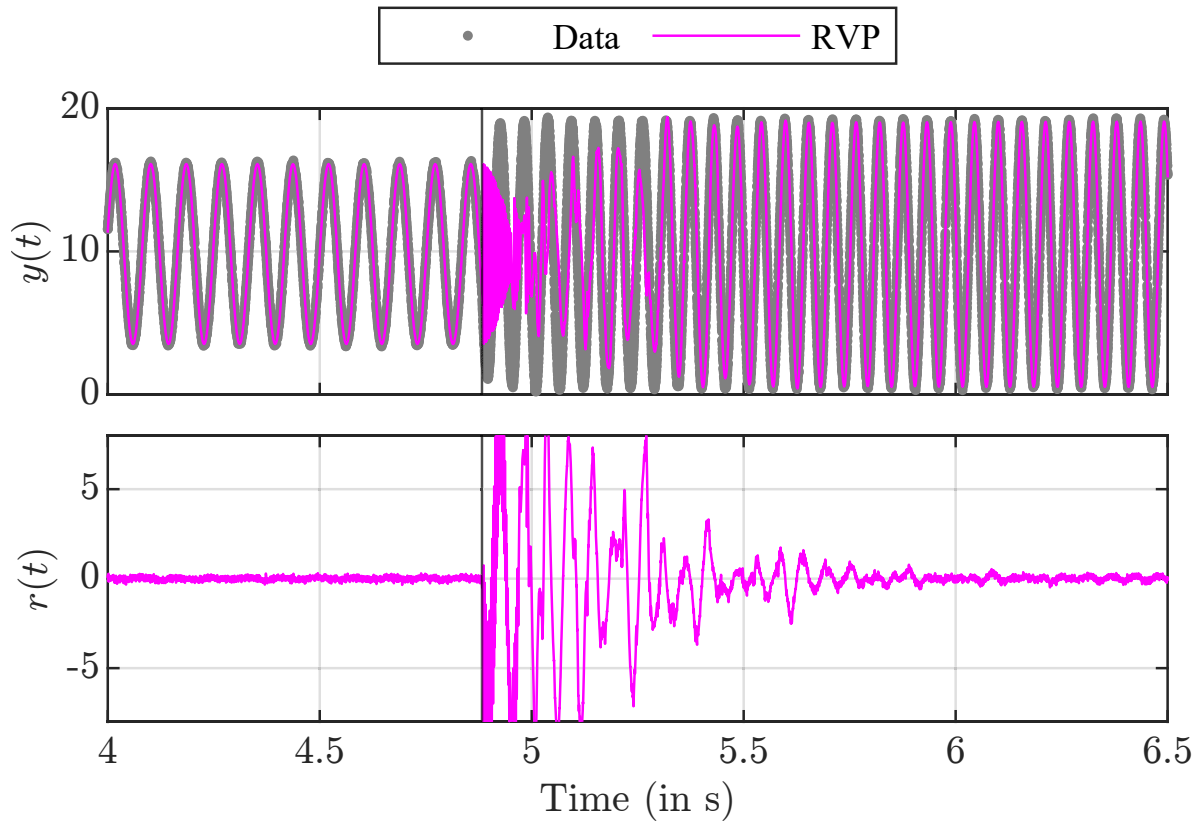


Figure 7: Measurement data with a change in the frequency, phase and amplitude of the signal used to demonstrate the tracking ability of the RVP algorithm. (top) The measurement data y with a change in the signal change at $t = 4.9$ s and the signal model determined by the RVP algorithm. (bottom) Residual r of the signal model.

5 Conclusion

This paper presented a new approach for the real-time parameter estimation of periodic signals, based on the Recursive Variable Projection (RVP) algorithm. The recursive implementation enables the tracking of time-varying signal parameter. Furthermore, the computational complexity of the new algorithm is $\mathcal{O}(8p^2 + 43p)$ where p is the number of samples in the dynamic time window. This, together with the applied hardware, determines the interval that can be achieved in a real-time context. Numerical testing via Monte Carlo simulations revealed that the new implementation has a higher accuracy and is more robust to measurement noise than past solutions. The successful tracking of time-varying parameter is shown using an experimental test setup, consisting of an electrodynamic shaker and a MEMS accelerometer.

Bibliography

- [1] A. Albarbar, S. Mekid, A. Starr, and R. Pietruszkiewicz, "Suitability of mems accelerometers for condition monitoring: An experimental study," *Sensors*, vol. 8, no. 2, pp. 2192–2196, 2008.
- [2] J. S. Lee, S. Choi, S. Kim, C. Park, and Y. G. Kim, "A mixed filtering approach for track condition monitoring using accelerometers on the axle box and bogie," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 3, pp. 749–758, 2012.
- [3] J. Saucedo-Dorantes, M. Delgado-Prieto, J. Ortega-Redondo, R. Osornio-Rios, and R. Romero-Troncoso, "Multiple-fault detection methodology based on vibration and current analysis applied to bearings in induction motors and gearboxes on the kinematic chain," *Shock and Vibration*, vol. 84, 2016.
- [4] A. Prudhom, J. Antonino-Daviu, H. Razik, and V. Climente-Alarcon, "Time-frequency vibration analysis for the detection of motor damages caused by bearing currents," *Mechanical Systems and Signal Processing*, vol. 84, pp. 747–762, 2017.
- [5] I. Koene, R. Viitala, and P. Kuosmanen, "Internet of things based monitoring of large rotor vibration with a microelectromechanical systems accelerometer," *IEEE Access*, vol. 7, pp. 92 210–92 219, 2019.
- [6] "IEEE Standard for digitizing waveform recorders," *IEEE Std 1057-2017 (Revision of IEEE Std 1057-2007)*, 2018.
- [7] C. Nguyen and K. Srinivasan, "A new technique for rapid tracking of frequency deviations based on level crossings," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-103, no. 8, pp. 2230–2236, 1984.

- [8] I. Santamaría, C. Pantaleón, and J. Ibanez, “A comparative study of high-accuracy frequency estimation methods,” *Mechanical Systems and Signal Processing*, vol. 14, no. 5, pp. 819–834, 2000.
- [9] H. So, “A comparative study of three recursive least-squares algorithms for single-tone frequency tracking,” *Signal Processing*, vol. 83, no. 9, pp. 2059–2062, 2003.
- [10] S. Y. Xue and S. X. Yang, “Power system frequency estimation using supervised gauss–newton algorithm,” *Measurement*, vol. 42, no. 1, pp. 28–37, 2009.
- [11] C.-Y. Hsu, P. Indyk, D. Katabi, and A. Vakilian, “Learning-based frequency estimation algorithms,” in *International Conference on Learning Representations*, 2019.
- [12] P. K. Dash and S. Hasan, “A fast recursive algorithm for the estimation of frequency, amplitude, and phase of noisy sinusoid,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4847–4856, 2011.
- [13] L. Xu, F. Chen, F. Ding, A. Alsaedi, and T. Hayat, “Hierarchical recursive signal modeling for multifrequency signals based on discrete measured data,” *International Journal of Adaptive Control and Signal Processing*, vol. 35, no. 5, pp. 676–693, 2021.
- [14] L. Xu, “Separable multi-innovation newton iterative modeling algorithm for multi-frequency signals based on the sliding measurement window,” *Circuits, Systems, and Signal Processing*, vol. 41, pp. 805–830, 2021.
- [15] P. O’Leary and D. Ninevski, “Estimating parameters of a sine wave by the method of variable projection,” in *2021 IEEE International Instrumentation and Measurement Technology Conference*, 2021.
- [16] G. H. Golub and V. Pereyra, “The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate,” *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 413–432, 1973.
- [17] G. Golub and V. Pereyra, “Separable nonlinear least squares: the variable projection method and its applications,” *Inverse Problems*, vol. 19, no. 2, pp. R1–R26, 2003.
- [18] M. Gan, Y. Guan, G.-Y. Chen, and C. L. P. Chen, “Recursive variable projection algorithm for a class of separable nonlinear models,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 11, pp. 4971–4982, 2021.
- [19] L. Ngia and J. Sjoberg, “Efficient training of neural nets for nonlinear adaptive filtering using a recursive levenberg-marquardt algorithm,” *IEEE Transactions on Signal Processing*, vol. 48, no. 7, pp. 1915–1927, 2000.

- [20] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
- [21] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [22] T. Söderström and L. Ljung, *Theory and Practice of Recursive Identification*. MA, USA: MIT Press, 1983.
- [23] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. Prentice-Hall, 1984.
- [24] J. H. Hong, C. Zach, and A. Fitzgibbon, "Revisiting the variable projection method for separable nonlinear least squares problems," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5939–5947.
- [25] R. M. Johnstone, C. R. Johnson, R. R. Bitmead, and B. D. O. Anderson, "Exponential convergence of recursive least squares with exponential forgetting factor," in *1982 21st IEEE Conference on Decision and Control*, 1982, pp. 994–997.
- [26] M. Rollet, E. J. Theussl, P. O’Leary, R. Fruhmann, and B. Ellensohn, "A miniature multi-sensor system and it’s application in a condition monitoring," in *6th International Conference on Sensors Engineering and Electronics Instrumentation Advances*, 2020.
- [27] G. C. Goodwin, E. K. Teoh, and H. Elliott, "Deterministic convergence of a self-tuning regulator with covariance resetting," in *IEE Proceedings D-Control Theory and Applications*, 1983, pp. 6–8.

Chapter 5

Industrial Applications

The focus of this chapter are mathematical methods in different industrial areas, presented through a series of papers. It demonstrates the usefulness of mathematical concepts in a wide variety of industrial applications. The mathematical topic covered in this chapter is dimensionality reduction, which is a method commonly used in pre-processing of data, or as a first step of the analysis.

1 Dimensionality Reduction

Dimensionality reduction refers to a variety of methods which reduce the dimensions of a dataset, with the goal of obtaining a lower-dimensional representation while still maintaining some key properties of the original data. This is often done when the original dataset has many dimensions, not all of which contain useful information necessarily. The methods which are relevant to the papers in this chapter are: Principal Component Analysis (PCA), Partial Least Squares (PLS) and Canonical Correlation Analysis (CCA).

1.1 Principal Component Analysis

Principal Component Analysis (PCA) is one of the most widely used dimensionality reduction techniques. It aims to find a lower-dimensional representation of a dataset, while maintaining as much of the variance of the data as possible. Suppose a multidimensional dataset is given as a matrix \mathbf{X} of dimension $m \times n$. The rows (m) are different samples and the columns (n) represent different channels. For example, if one monitors a machine with 11 sensors and collect data for 1 hour at a rate of 1Hz, the resulting matrix \mathbf{X} would have dimension 3600×11 . Thus each sample represents a point in 11 dimensional space, where each coordinate represents a different channel (sensor). The first principal component \mathbf{p} of the matrix \mathbf{X} is a linear combination of the columns of \mathbf{X} , which can be written as

$$\mathbf{p} = \mathbf{X}\mathbf{v} \tag{5.1}$$

where the values of \mathbf{v} (which are the coefficients of the linear combination) are determined so as to maximize

$$\max_{\mathbf{v}} \frac{|\mathbf{p}|}{|\mathbf{v}|}. \quad (5.2)$$

In other words, PCA finds the linear combination of columns with the biggest variance, or equivalently the direction of maximal variance. This is equivalent to maximizing

$$\max_{\mathbf{v}} \left(\frac{\mathbf{v}^T \mathbf{X}^T \mathbf{X} \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \right). \quad (5.3)$$

Since directions are invariant wrt. scalar multiples, the coefficients of the linear combination are relevant up to a scalar, so one can take that $\mathbf{v}^T \mathbf{v} = 1$ and thus the maximization problem becomes a constrained maximization problem of the form

$$\max_{\mathbf{v}} (\mathbf{v}^T \mathbf{X}^T \mathbf{X} \mathbf{v}) \quad \text{given that} \quad \mathbf{v}^T \mathbf{v} = 1. \quad (5.4)$$

This problem can be solved using Lagrange multipliers. Namely,

$$\frac{d}{d\mathbf{v}} (\mathbf{v}^T \mathbf{X}^T \mathbf{X} \mathbf{v} - \lambda (\mathbf{v}^T \mathbf{v} - 1)) = 0 \quad (5.5)$$

which becomes

$$\mathbf{X}^T \mathbf{X} \mathbf{v} = \lambda \mathbf{v}. \quad (5.6)$$

This means that \mathbf{v} is an eigenvector for the matrix $\mathbf{X}^T \mathbf{X}$, and thus it is a right singular vector for the matrix \mathbf{X} , denoted as \mathbf{v}_1 . The first principal component is then calculated as

$$\mathbf{p} = \mathbf{X} \mathbf{v}_1. \quad (5.7)$$

Similarly, the other principal components will be the other right singular vectors. So the first k principal components are calculated as

$$\underbrace{\mathbf{P}}_{m \times k} = \underbrace{\mathbf{X}}_{m \times n} \underbrace{\mathbf{V}_k}_{n \times k} \quad (5.8)$$

where \mathbf{V}_k contains the first k columns of the matrix \mathbf{V} . The matrix \mathbf{P} can be used as a lower dimensional representation of \mathbf{X} , since it has the same number of samples m , but less columns k instead of n .

1.2 Partial Least Squares

Partial Least Squares (PLS) is another standard dimensionality reduction method, similar to PCA. Given two matrices \mathbf{X} and \mathbf{Y} , which have the same number of rows (observations) but may have

different number of columns (variables), the goal of PLS is to find vectors \mathbf{a} and \mathbf{b} which maximize the covariance of $\mathbf{v} = \mathbf{X}\mathbf{a}$ and $\mathbf{u} = \mathbf{Y}\mathbf{b}$. In other words, the goal is to maximize $s_{\mathbf{v}\mathbf{u}}$, where

$$s_{\mathbf{v}\mathbf{u}} = \frac{1}{n-1} (\mathbf{v} - \bar{\mathbf{v}})^T (\mathbf{u} - \bar{\mathbf{u}}). \quad (5.9)$$

One can assume that the data has been made mean free before doing PLS, so the function that should be maximized is

$$\mathbf{v}^T \mathbf{u} = \mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b}. \quad (5.10)$$

Since the magnitudes of the vectors \mathbf{a} and \mathbf{b} are irrelevant, one can rewrite the cost function as

$$\max_{\|\mathbf{a}\|=1, \|\mathbf{b}\|=1} \mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b} \quad (5.11)$$

Using Lagrange multipliers, one gets

$$\begin{aligned} \frac{\partial}{\partial \mathbf{a}} [\mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b} + \lambda (\|\mathbf{a}\| - 1) + \mu (\|\mathbf{b}\| - 1)] &= 0 \\ \frac{\partial}{\partial \mathbf{b}} [\mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b} + \lambda (\|\mathbf{a}\| - 1) + \mu (\|\mathbf{b}\| - 1)] &= 0 \\ \frac{\partial}{\partial \lambda} [\mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b} + \lambda (\|\mathbf{a}\| - 1) + \mu (\|\mathbf{b}\| - 1)] &= 0 \\ \frac{\partial}{\partial \mu} [\mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b} + \lambda (\|\mathbf{a}\| - 1) + \mu (\|\mathbf{b}\| - 1)] &= 0 \end{aligned}$$

Looking at the first two equations, one gets

$$\begin{aligned} \frac{\partial}{\partial \mathbf{a}} [\mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b} + \lambda (\mathbf{a}^T \mathbf{a} - 1) + \mu (\mathbf{b}^T \mathbf{b} - 1)] &= \mathbf{X}^T \mathbf{Y} \mathbf{b} + 2\lambda \mathbf{a} = 0 \\ \frac{\partial}{\partial \mathbf{b}} [\mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b} + \lambda (\mathbf{a}^T \mathbf{a} - 1) + \mu (\mathbf{b}^T \mathbf{b} - 1)] &= \mathbf{Y}^T \mathbf{X} \mathbf{a} + 2\mu \mathbf{b} = 0 \end{aligned}$$

Substituting for $\mathbf{a} = -\frac{1}{2\lambda} \mathbf{X}^T \mathbf{Y} \mathbf{b}$ from the first into the second equation, one gets

$$-\frac{1}{2\lambda} \mathbf{Y}^T \mathbf{X} \mathbf{X}^T \mathbf{Y} \mathbf{b} + 2\mu \mathbf{b} = 0$$

In other words, \mathbf{b} is an eigenvector of the matrix $\mathbf{Y}^T \mathbf{X} \mathbf{X}^T \mathbf{Y}$. Similarly, one gets that \mathbf{a} is an eigenvector of the matrix $\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}$. So, the vectors that would maximize the cost function would be the eigenvectors corresponding to the largest eigenvalues.

Once the vectors \mathbf{a} and \mathbf{b} are found, the next step would be to do the same procedure for the matrices $\mathbf{X}_1 \triangleq \mathbf{X} - \mathbf{X} \mathbf{a} \mathbf{a}^T$ and $\mathbf{Y}_1 \triangleq \mathbf{Y} - \mathbf{Y} \mathbf{b} \mathbf{b}^T$. This removes the variance associated with the already computed vectors before computing the new vectors.

2 Canonical Correlation Analysis

Suppose now that two multidimensional datasets are given as matrices \mathbf{X} and \mathbf{Y} of dimension $m \times n_1$ and $m \times n_2$, meaning they have the same number of rows (samples) but different numbers of columns (sensors). Similar to PCA, the goal of Canonical Correlation Analysis (CCA) is to find linear combinations of the columns of \mathbf{X} and \mathbf{Y} with the highest correlation. In other words, it finds vectors \mathbf{a} and \mathbf{b} which maximize the correlation between $\mathbf{X}\mathbf{a}$ and $\mathbf{Y}\mathbf{b}$. The correlation of any two mean free vectors \mathbf{u} and \mathbf{v} can be calculated as

$$\text{cor}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \quad (5.12)$$

so the correlation between the vectors $\mathbf{X}\mathbf{a}$ and $\mathbf{Y}\mathbf{b}$ is

$$\text{cor}(\mathbf{X}\mathbf{a}, \mathbf{Y}\mathbf{b}) = \frac{\mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b}}{\sqrt{\mathbf{a}^T \mathbf{X}^T \mathbf{X} \mathbf{a} \mathbf{b}^T \mathbf{Y}^T \mathbf{Y} \mathbf{b}}}. \quad (5.13)$$

Similar to before, one can assume that \mathbf{a} and \mathbf{b} are unit vectors, so the maximization problem becomes

$$\max_{\|\mathbf{a}\|=1, \|\mathbf{b}\|=1} \frac{\mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b}}{\sqrt{\mathbf{a}^T \mathbf{X}^T \mathbf{X} \mathbf{a} \mathbf{b}^T \mathbf{Y}^T \mathbf{Y} \mathbf{b}}}. \quad (5.14)$$

Alternatively, one can impose a different assumption on the vectors \mathbf{a} and \mathbf{b} and write the problem as

$$\max_{\substack{\mathbf{a}^T \mathbf{X}^T \mathbf{X} \mathbf{a} = 1 \\ \mathbf{b}^T \mathbf{Y}^T \mathbf{Y} \mathbf{b} = 1}} \mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b}. \quad (5.15)$$

Using Lagrange multipliers to solve this, one obtains

$$\frac{d}{d\mathbf{a}} (\mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b} - \lambda (\mathbf{a}^T \mathbf{X}^T \mathbf{X} \mathbf{a} - 1) - \mu (\mathbf{b}^T \mathbf{Y}^T \mathbf{Y} \mathbf{b} - 1)) = 0 \quad (5.16)$$

$$\frac{d}{d\mathbf{b}} (\mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b} - \lambda (\mathbf{a}^T \mathbf{X}^T \mathbf{X} \mathbf{a} - 1) - \mu (\mathbf{b}^T \mathbf{Y}^T \mathbf{Y} \mathbf{b} - 1)) = 0. \quad (5.17)$$

This becomes

$$\mathbf{X}^T \mathbf{Y} \mathbf{b} - \lambda (2\mathbf{X}^T \mathbf{X} \mathbf{a}) = 0 \quad (5.18)$$

$$\mathbf{Y}^T \mathbf{X} \mathbf{a} - \mu (2\mathbf{Y}^T \mathbf{Y} \mathbf{b}) = 0. \quad (5.19)$$

Assuming that \mathbf{X} and \mathbf{Y} have linearly independent columns (meaning the sensors are independent of each other), from Chapter 2 it is clear that $\mathbf{X}^T \mathbf{X}$ and $\mathbf{Y}^T \mathbf{Y}$ are invertible, so from 5.18 one obtains

$$\mathbf{a} = \frac{1}{2\lambda} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \mathbf{b} \quad (5.20)$$

and substituting this into 5.19 one obtains

$$\mathbf{Y}^T \mathbf{X} \frac{1}{2\lambda} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \mathbf{b} = 2\mu (\mathbf{Y}^T \mathbf{Y} \mathbf{b}) \quad (5.21)$$

so

$$\left[(\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \right] \mathbf{b} = 4\mu\lambda \mathbf{b} \quad (5.22)$$

which means that \mathbf{b} is an eigenvector of the matrix $(\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$. Similarly, we get that \mathbf{a} is an eigenvector of the matrix $(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} (\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T \mathbf{X}$.

There is a way to compute the required vectors \mathbf{a} and \mathbf{b} using the SVD of the data matrices. Let

$$\mathbf{X} = \mathbf{U}_x \mathbf{S}_x \mathbf{V}_x^T, \quad \mathbf{Y} = \mathbf{U}_y \mathbf{S}_y \mathbf{V}_y^T, \quad (5.23)$$

thus the correlation $\mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b}$ now becomes

$$\mathbf{a}^T \mathbf{X}^T \mathbf{Y} \mathbf{b} = \mathbf{a}^T \mathbf{V}_x \mathbf{S}_x \mathbf{U}_x^T \mathbf{U}_y \mathbf{S}_y \mathbf{V}_y^T \mathbf{b}. \quad (5.24)$$

Additionally, let the SVD of the matrix $\mathbf{U}_x^T \mathbf{U}_y$ be

$$\mathbf{U}_x^T \mathbf{U}_y = \mathbf{U}_c \mathbf{S}_c \mathbf{V}_c^T. \quad (5.25)$$

Then, it can be checked through substitution that the solutions for the vectors \mathbf{a} and \mathbf{b} are given as the columns of the matrices \mathbf{A} and \mathbf{B} defined as

$$\mathbf{A} = \mathbf{V}_x \mathbf{S}_x^{-1} \mathbf{U}_c, \quad \mathbf{B} = \mathbf{V}_y \mathbf{S}_y^{-1} \mathbf{V}_c. \quad (5.26)$$

The benefit of this approach is that PCA is also performed at the start of the process of CCA, which in many applications is useful.

Bibliography

- [1] A. Eisinberg, G. Franzé, and N. Salerno, “Rectangular vandermonde matrices on chebyshev nodes,” *Linear Algebra and its Applications*, vol. 338, no. 1, pp. 27–36, 2001. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S002437950100355X>
- [2] H. Wertz, “On the numerical inversion of a recurrent problem: The vandermonde matrix,” *IEEE Transactions on Automatic Control*, vol. 10, no. 4, pp. 492–492, 1965.
- [3] J. Baik, T. Kriecherbauer, K. D.-R. McLaughlin, and P. D. Miller, *Discrete Orthogonal Polynomials. (AM-164): Asymptotics and Applications (AM-164): Asymptotics and Applications (AM-164)*. Princeton University Press, 2007. [Online]. Available: <https://doi.org/10.1515/9781400837137>
- [4] G. Szegő, S. G. Szego, and A. M. Society, *Orthogonal Polynomials*, ser. American Math. Soc: Colloquium publ. American Mathematical Society, 1939. [Online]. Available: <https://books.google.at/books?id=ZOhmnsXlcY0C>
- [5] M. Harker, *Fractional Differential Equations: Numerical Methods for Applications*, ser. Studies in Systems, Decision and Control. Springer Cham.
- [6] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- [7] I. Gelfand and S. Fomin, *Calculus of Variations*, ser. Dover Books on Mathematics. Dover Publications, 2012. [Online]. Available: <https://books.google.at/books?id=CeC7AQAQBAJ>
- [8] G. H. Golub and V. Pereyra, “The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate,” *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 413–432, 1973.
- [9] G. Golub and V. Pereyra, “Separable nonlinear least squares: the variable projection method and its applications,” *Inverse Problems*, vol. 19, pp. R1–R26(1), 01 2003.

Measurement of Relative Position and Orientation using UWB

Ernst-Johann Theussl, Dimitar Ninevski and Paul O’Leary
University of Leoben, A8700 Leoben, Austria
automation@unileoben.ac.at
<http://automatiom.unileoben.ac.at>

Abstract

This paper introduces a new approach to measure relative positioning and orientation (RPO), by instrumenting mobile equipment with ultra-wideband (UWB) distance measurements. In this experiment RPO is tested without a surrounding stationary UWB anchor network; all necessary UWB devices are directly mounted on the machinery. This results in a simplified implementation in the industry, but also challenges the RPO determination. Due to this, the precision and uncertainty of the UWB measurements were characterized in a real application environment, i.e., on a quay to determine if a large body of water would influence the high frequency signals. It was determined that the UWB distance measurements had an uncertainty of approximately 20 mm when measuring orthogonal to the antenna and 35 mm at large angles; both results for 95% confidence. Based on these results a new approach to calculating the RPO was implemented and tested. The goal was to minimize the effects of horizontal dilution of precision. Three new algorithms were implemented and tested, the best results were obtained with a non-linear covariance weighted least squares computation with self calibration. With this approach the standard deviation of the x and y positions were 64 mm and 13 mm respectively. This is a very significant improvement in performance with respect to trilateration, which is currently the standard approach.

Keywords: Positioning, Orientation, Localization, Ultra wideband technology, Location awareness.

1 Introduction

This paper presents the instrumentation of mobile machinery with ultra-wideband (UWB) distance measurements, so as to enable the measurement of their relative positions and orientations (RPO) without the need of a surrounding UWB-anchor network. RPO measurements are important in collision avoidance and to support the dynamic reconfiguration of plant and machinery. UWB has been chosen since the RPO must function both in- and outdoors, as well as in underground situations, (see [1] for an example of underground applications). There are situations where GPS and/or GNSS may not be available. The uncertainties of the distance measurements are characterized and probability models are established; from these results the horizontal dilution of precision (HDOP) associated with trilateration are determined. A new approach to computing the RPO is presented, which enables the use of a constrained covariance weighted least squares minimization to obtain higher precision in the measurement with reduced uncertainty¹.

Various wireless-based technologies are already introduced for positioning, each of them provides different characteristics in terms of positioning accuracy, reliability and reach. Bluetooth Low Energy (BLE) [2, 3], Radio Frequency Identification (RFID) [4], Chirp Spread Spectrum (CSS) [5] or Wi-Fi [5] do not fulfil the desired positioning accuracy of ± 10 cm, or do not cover the required application reach of 3 m to 20 m.

The used UWB-modules in this test (see Fig. 1) should reach accuracies in distance measuring or location information of 10 cm within a reach of 0 m to 20 m according to the manufacturer specifications [6]. These results have already been shown under obstacle-free and line-of-sight (LOS) conditions [7]. In the case of industrial conditions the distance measurement error increased [8].

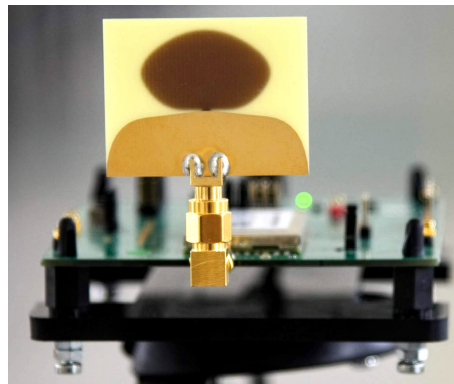


Figure 1: The used UWB modules on a tripod-mount in a previous experiment of range tests.

The increase of measurement error is based on the high frequency radio signal behaviour, affected by the environment. These effects appear as signal noise or outliers in the measured data. This is mostly caused by multipath effects or signal loss during the ranging process. Metal surfaces

¹The instrumentation and mathematical methods which enable the measurement of relative position and orientation with improved precision and reduced uncertainty is highly relevant to the I²MTC community.

and liquids especially influence the UWB measurements negatively [4] by an establishment of a dense multi-path environment [9]; conditions that often exist in the industry. For this reason the UWB-measurements were carried out on a ship quay under comparable settings.



Figure 2: Measurement setup with two containers at a ship quay.

2 Experimental setup

The experimental setup of UWB distance measurements was carried out between freight containers in an industrial environment under LOS conditions. Two 20ft-ISO-containers were directly equipped with UWB-modules to represent an arbitrarily scalable industrial RPO-system. Each UWB-module was mounted on the containers at the same height of $2m$. A level plane of all containers was assumed, which reduced the RPO-task down to two dimensions. The container-1 (C-I) was stationary positioned and defined the relative coordinate origin, container-2 (C-II) was moved in one-meter steps in positions from 2 to 10m (Fig. 3). Four lengths $l_{11}, l_{12}, l_{21}, l_{22}$ were measured through UWB and laser, to calculate the RPO of container-2 at each position. Due to further statistical investigation, each UWB length measurement was done with a sample number of $n = 997$. Reference distances are measured via laser for comparability. The used UWB-modules are based on the decaWave DW1000 UWB-chipset operating at a frequency of approximately 6.5 GHz using an external antenna.

The UWB-modules were placed in standard IP66 poly carbonate housings with an autonomous power source inside; directly onto the containers C-I and C-II. All measurements were done successively to avoid dependencies or interference between the measurements. The containers were aligned via forklift to predefined marked reference positions.

3 Characterizing UWB measurement errors

Each of the four distances l_{11}, l_{12}, l_{21} and l_{22} (see Fig. 3) were measured with a laser and with UWB, to enable a verification of the computational results. The laser measurements were considered to deliver the reference distances, $l_{t(ij)}$, multiple UWB measurements, $l_{m(ij)}$ were considered

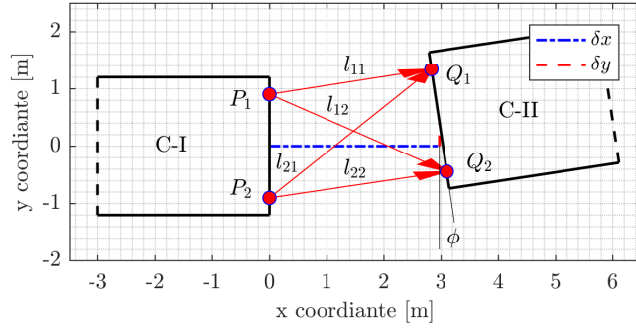


Figure 3: The two containers C-I and C-II with UWB module positions P_1 , P_2 and Q_1 , Q_2 , the 4 measured distances l_{11} l_{12} l_{21} l_{22} and the calculated parameters δx , δy and ϕ

as being perturbed with an error, ε_{ij} , such that,

$$l_{m(ij)} = l_{t(ij)} + \varepsilon_{ij}. \quad (5.27)$$

The box-and-whiskers for the errors ε_{ij} relative to the reference values is shown in Fig. 4, and relative to the median of the UWB data in Fig. 5. The statistical errors are given as standard deviations with their respective 95% confidence intervals in Table 5.3 . It was determined that the dominant errors are due to systematic errors; caused primarily through the angular dependence of the antenna.

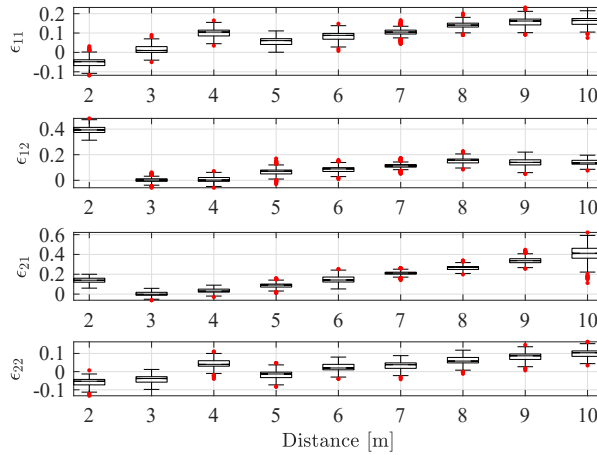


Figure 4: Box-and-whiskers plots of the measurement errors. The measurements are used to determine the position of the moving container between 2m to 10m in 1 m intervals. The red dots are outliers. The offsets of the straight-line measurement errors ε_{11} and ε_{22} at at 2 m are smaller than the offsets of ε_{12} and ε_{21} of the diagonal measured distances.

As seen in Fig. 4 and Fig. 5, the errors of the measurements ε_{11} and ε_{22} compared to ε_{12} and ε_{21} behave statistically different. Fig. 4 shows the largest offsets and standard deviations for

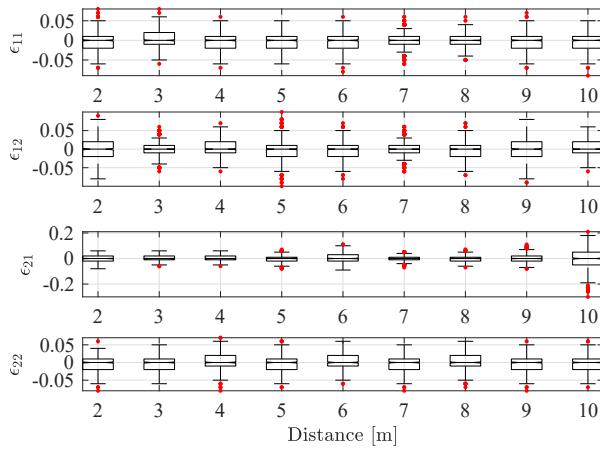


Figure 5: Box-and-whiskers plots of the errors centred at their respective medians to visualize the difference in their spread.

the diagonal measurements. According to Fig. 3, ϵ_{11} and ϵ_{22} correspond to straight-line distance measurements, while ϵ_{12} and ϵ_{21} correspond to diagonal distance measurements. Also, with the increase of distance, some measurements become less reliable. All of these things will be taken into consideration in the model later.

Since the introduced methods for RPO (see chapter V) require statistically independent data, this property must be checked. The fact that the measurements $l_{11}, l_{12}, l_{21}, l_{22}$ were made using different UWB-modules suggests that the measurements were independent. The correlation coefficients were calculated and the results were close enough to 0 to suggest the same (see Table 5.1).

Distance [m]	$\rho(l_{11}, l_{21})$	$\rho(l_{12}, l_{22})$
2m	0.034	0.059
3m	0.016	0.113
4m	0.033	0.027
5m	0.026	0.083
6m	0.017	0.021
7m	0.042	0.098
8m	0.051	0.058
9m	0.022	0.032
10m	0.058	0.055

Table 5.1: Correlation coefficients ρ of the measurement sets used to determine $Q_1(l_{11}, l_{21})$ and $Q_2(l_{12}, l_{22})$. The correlation coefficients are small enough to justify the assumption that the measurements are uncorrelated at any given distance.

This is an insufficient proof for independence. However, if the paired data, $[l_{11}, l_{21}]$ which was used for determining the position of Q_1 , and $[l_{12}, l_{22}]$ for Q_2 , comes from a bivariate normal distribution, then this would be sufficient to prove independence [10].

The test used to check this is the Henze-Zirkler Test for Multivariate Normality [11]. Using a MATLAB implementation of this test [12], it was confirmed that the data comes from a bivariate normal distribution (at the 95% significance level).

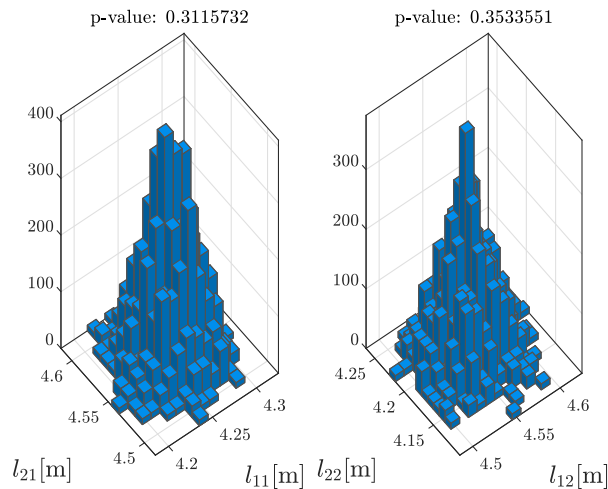


Figure 6: The joint histograms of the measurements $[l_{11}, l_{21}]$ and $[l_{12}, l_{22}]$ taken at 4 m. The shape resembles that of a bivariate normal distribution (considering the discretization of the measurements) and the Henze-Zirkler's test confirms this at the 95% significance level, as can be seen from the p-values.

Having proven that the paired data follows a bivariate normal distribution, together with the fact that they are uncorrelated, implies independence of the measurements. Also, this implies that the measurements themselves come from a normal distribution (since they are the marginal distributions of a bivariate normal distribution).

If only univariate normality is checked, the fact that the data is uncorrelated wouldn't be enough to prove that it is independent (for a counter-example, check [13]). However, checking that the paired data follows a bivariate normal distribution goes around this problem.

4 Trilateration and Uncertainty

A common method to determine an unknown position $P(x, y)$ in a two dimensional space is trilateration [4, 14]. Given two distances l_{11} and l_{21} from two different origins, it uses circle geometry to determine P . The positions of both UWB-modules on container C-I are denoted by $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$. The point being trilaterated, Q_1 , is the intersection of two circles centred at P_1 and P_2 with radii l_{11} and l_{21} (or l_{12} and l_{22} for Q_2) respectively.

Considering all possible values of the pairs (l_{11}, l_{21}) and (l_{12}, l_{22}) , different coordinates for Q_1 and Q_2 are obtained. So, if there are u different l_{11} values and v different l_{12} values with different frequencies, there will be uv different values for the Q_1 . The same applies to Q_2 . This gives a 2-D grid of possible positions, which is graphically shown in Fig. 7.

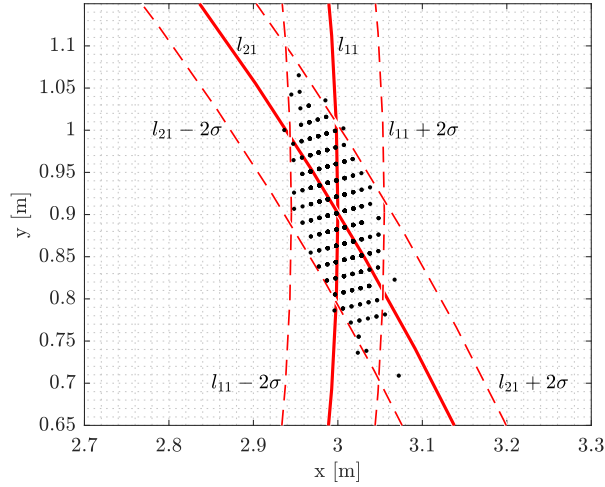


Figure 7: The positioning results of uncertain measurements: The analytical result of possible positions is defined as the area bounded by the dashed circles. The practical results measured in this experiment are shown as black dots. Based on the principle of triangulation the measured lengths are interpreted as circles centred at P_1 and P_2 . The reference distances are represented by l_{11} and l_{21} . The uncertain distances are limited by $l_{11} \pm 2\sigma$ and $l_{21} \pm 2\sigma$, where σ is the standard deviation of all measurements. Both shapes are almost identical.

The dashed circle arcs in Fig. 7 represent the 2σ confidence intervals of l_{11} and l_{21} , where σ is the standard deviation of all measurements. The area bounded by these circles defines theoretically the possible positions of Q_1 . The dots represent the actual position of Q_1 calculated through the UWB measured lengths, centred around the intersection of the full circles. This shows that the calculated positions Q lie within this area. The positions are centred in x and y to compensate the measurement offset and make the shape of positioning area comparable to the analytical result. Almost all measured positions are in the bounded area. Without centring the trilaterated points it would be apparent that the deviations in y are greater than in x direction due to the offset of the diagonal measurements.

The description of the possible accuracy in positioning depending on the enclosed area is already characterized as Horizontal Dilution of Precision (HDOP); this occurs in trilateration of positions through noise affected distance measurements. The computed HDOP [15] for the 2-dimensional case with a distance $2d$ between the UWB-modules and a distance x between the container is:

$$HDOP(x) = \sqrt{2} \frac{\sqrt{x^2 + 4d^2}}{2d} \quad (5.28)$$

Distance [m]	HDOP	Interpretation
2m	2.1	good
3m	2.7	good
4m	3.4	good
5m	4.2	good
6m	4.9	good
7m	5.7	moderate
8m	6.4	moderate
9m	7.2	moderate
10m	8.0	moderate

Table 5.2: The HDOP evaluated at 1 m intervals from 2 m to 10 m. Low values (< 5) correspond to smaller dilution and more precise results.

The HDOP value is an indicator for the possible accuracy of a trilateration based on uncertain distance measurements. A larger HDOP value is more uncertain in terms of positioning accuracy than a smaller value. This correlated to the analytical result, represented by the area between the boundaries in Fig. 7. However the trilaterated position at 2m showed larger deviations than the HDOP indicates. Since the trilateration of each position is based on two lengths, at least one of those lengths includes the distance which caused the deviations. According to Fig. 4, this behaviour is based on the more uncertain diagonal length measurements. The diagonal measurements had the common feature that the angle between the antennas changed depending on the container distance. The larger the distance between the containers, the smaller the antenna angle. This circumstance indicates a possible influence of the antenna angle on the accuracy of the UWB measurement.

In order to assess this influence, the angular dependence is measured separately at a distance of 5 m between -50° and 50° antenna angle in 10° steps.

The inaccuracy of measurement increases towards positive and negative increasing antenna angles (see Fig. 8). In this experiment, the difference between the 10° and the -50° measurement is approximately 6 cm, which is already close to the application limit of ± 10 cm.

5 Positioning models

The calculation of x and y positions through purely trilateration did not provide the desired accuracy. Therefore, improvements of the position determination were required. The UWB-modules

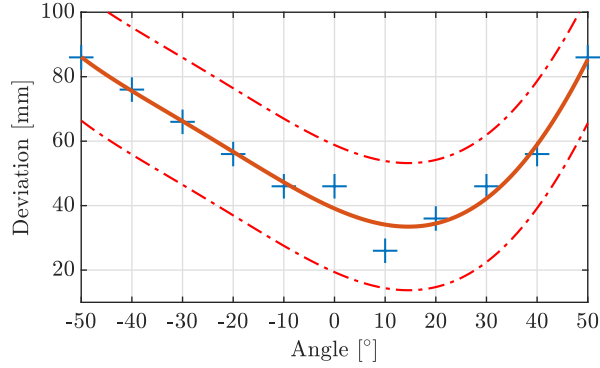


Figure 8: The accuracy of the distance measurement at 5 m depending on different antenna angles. The blue crosses are the differences of the median of the measured distances and the reference. The red line is a polynomial of degree 5, which approximates the data. The dashed lines are 2σ confidence intervals, where σ is the standard deviation of each measurement.

were mounted on fixed places on the containers, therefore the euclidean distance between them was invariant. Furthermore, the relative position and orientation of container-2 is modelled as the result of a rotation about the relative origin between P_1 and P_2 followed by a translation in δx and δy . Let

$$\mathbf{t} = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix} \quad (5.29)$$

be the translation vector and rotation matrix respectively. The position vector q is given by

$$\mathbf{q} = \mathbf{R}\mathbf{p} + \mathbf{t} \quad (5.30)$$

This model has 3 degrees of freedom $\delta x, \delta y, \phi$ and 4 independent measurements $l_{11}, l_{12}, l_{21}, l_{22}$. This results in an overdetermined system of equations. Therefore, the system must be approximated by the minimization of a cost function.

To determine $\delta x, \delta y, \phi$ given four measurements $l_{11}, l_{12}, l_{21}, l_{22}$, a Cartesian coordinate system is used. It is centred at the median of the two sensor positions on container-1. The coordinates of the UWB-modules on container-1 are $P_1(0, d)$ and $P_2(0, -d)$, where $2d$ is the distance between them. Then, the position vectors of the points Q_1 and Q_2 will be

$$\mathbf{q}_1 = \begin{bmatrix} \delta x - d \sin(\phi) \\ \delta y + d \cos(\phi) \end{bmatrix}, \quad \mathbf{q}_2 = \begin{bmatrix} \delta x + d \sin(\phi) \\ \delta y - d \cos(\phi) \end{bmatrix}, \quad (5.31)$$

By the determination of the distances between P_1, P_2, Q_1, Q_2 , 4 equations in 3 variables can be obtained. Solving this approximately gives a solution, which is then back-substituted into the coordinates of the points Q_1 and Q_2 to recalculate 4 distances $m_{11}, m_{12}, m_{21}, m_{22}$. Thus, it is now possible to compare the via UWB measured distances l_{ij} and the model-based recalculated distances m_{ij} .

	l_{11}	l_{12}	l_{21}	l_{22}
σ	$20.4^{+0.3032}_{-0.2944}$	$23.6^{+0.3508}_{-0.3406}$	$34.9^{+0.5183}_{-0.5033}$	$21.8^{+0.3245}_{-0.3150}$

Table 5.3: Standard deviations for the lengths l_{11} , l_{12} , l_{21} and l_{22} , together with and 95% confidence intervals. These results are computed over the ensemble of all measurements, i.e., there are $n = 8793$ individual measurements for each σ . The standard deviations are given in millimetres.

6 Constrained, covariance weighted, least squares model

Four different measurements of four different lengths on different positions (2m to 10m) were done. The lengths are denoted by $l_{11}, l_{12}, l_{21}, l_{22}$. Each length measurement at each position is done 997 times. The results of the measurements are collected in the vectors $l_{11}, l_{12}, l_{21}, l_{22}$. As explained in the previous section, the vectors $m_{11}, m_{12}, m_{21}, m_{22}$, are calculated.

The deviation of the measured lengths and the recalculated lengths are determined by

$$\begin{aligned}
r_{11} &= l_{11} - m_{11} \\
r_{12} &= \alpha_1 l_{12} + \alpha_0 - m_{12} \\
r_{21} &= \beta_1 l_{21} + \beta_0 - m_{21} \\
r_{22} &= l_{22} - m_{22}.
\end{aligned}$$

Because the diagonal measurements are less reliable than the straight measurements (see Table 5.3), linear models with the coefficients $\alpha_0, \alpha_1, \beta_0$ and β_1 are introduced in r_{12} and r_{21} . Also, because the measurements have different variances at the same distances, the actual residual constrained covariance weighted vector r_{cw} can be defined as

$$r_{cw} = \left[\frac{1}{\sigma_{11}} r_{11}, \frac{1}{\sigma_{12}} r_{12}, \frac{1}{\sigma_{21}} r_{21}, \frac{1}{\sigma_{22}} r_{22} \right]^T \quad (5.32)$$

where σ_{ij} is the standard deviation of $l_{ij} - m_{ij}$ for $i, j \in \{1, 2\}$. The standard deviations weigh each residual, so that larger standard deviations will decrease the individual residuals and weigh it less. The way r is defined also avoids the need for calibration prior to determining the position.

Finally, the cost function is defined as

$$C = r_{cw}^T r_{cw}$$

which will be minimized.

For comparison, the residual vector and the cost function are also calculated without con-

straints:

$$\begin{aligned} \mathbf{r}_{(u)11} &= \mathbf{l}_{11} - \mathbf{m}_{11} \\ \mathbf{r}_{(u)12} &= \mathbf{l}_{12} - \mathbf{m}_{12} \\ \mathbf{r}_{(u)21} &= \mathbf{l}_{21} - \mathbf{m}_{21} \\ \mathbf{r}_{(u)22} &= \mathbf{l}_{22} - \mathbf{m}_{22} \end{aligned}$$

using two different methods: with weighting

$$\mathbf{r}_{uw} = \left[\frac{1}{\sigma_{11}} \mathbf{r}_{(u)11}, \quad \frac{1}{\sigma_{12}} \mathbf{r}_{(u)12}, \quad \frac{1}{\sigma_{21}} \mathbf{r}_{(u)21}, \quad \frac{1}{\sigma_{22}} \mathbf{r}_{(u)22} \right]^T \quad (5.33)$$

and without weighting

$$\mathbf{r}_n = \left[\mathbf{r}_{(u)11}, \quad \mathbf{r}_{(u)12}, \quad \mathbf{r}_{(u)21}, \quad \mathbf{r}_{(u)22} \right]^T \quad (5.34)$$

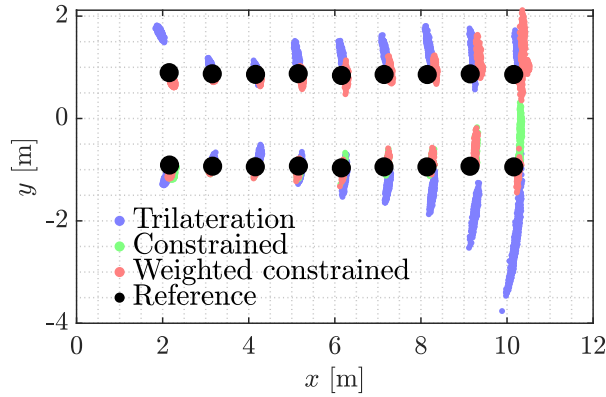


Figure 9: The comparison of positioning results for 3 different methods: Trilateration, constrained approximation and weighted constrained approximation using the lengths l_{11} , l_{12} , l_{21} and l_{22}

The different positioning methods are compared in Fig.9. It is apparent, that the covariance weighted constrained method gives the most accurate results. It can be seen how uncertain the positions are at far distances, especially at 10m. Due to this, the variance of these measurements compared to the others is significantly larger, hence it makes sense to weigh them less.

Furthermore, the constrained models enable much more accurate positioning at 2 m than purely trilateration. This is apparent in Fig. 10.

7 Conclusions

This paper introduces a new method for determining the relative position and orientation (RPO) of objects to align mobile machinery in industrial environments, without a surrounding UWB-anchor

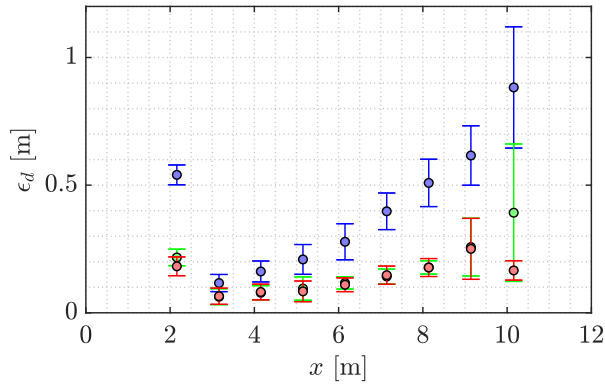


Figure 10: Comparison of the errors ε_d of positioning using different methods. The covariance weighting (red box-plots) increases the positioning accuracy compared to pure trilateration (blue box-plots) and unweighted approximation (green box-plots).

	r_n		r_{uw}		r_{cw}	
	Δx	Δy	Δx	Δy	Δx	Δy
	[mm]		[mm]		[mm]	
d_1	88.59	-198.34	46.69	-176.34	-59.24	6.93
d_2	-8.78	-49.78	-8.94	-53.06	-18.16	36.40
d_3	48.41	-33.48	46.71	-44.52	69.85	43.09
d_4	48.94	-68.56	41.79	-48.28	19.75	27.53
d_5	85.99	19.98	74.60	-45.87	51.28	54.51
d_6	117.74	58.00	118.74	86.64	70.85	44.26
d_7	156.43	-18.68	154.02	122.75	103.78	38.35
d_8	176.05	321.23	163.94	310.49	123.37	36.88
d_9	189.13	608.21	144.20	-66.76	130.07	36.06
σ_p	65.82	244.83	60.56	142.72	64.05	13.16

Table 5.4: The comparison between three different methods of positioning: Unweighted r_n ; unconstrained weighted r_{uw} ; constrained weighted r_{cw} . σ_p represents the standard deviation of all positioning errors.

network. Due to this challenging conditions, the need for optimization in positioning became apparent. The algorithms developed for this use-case were applied to the recorded data. Through comparison, it can be concluded that the method with constrained approximation and covariance weighting provides the best results overall. It reduced the standard deviation of the error in the y -direction down to 13 mm and the x -direction to 64 mm, which is within the desired accuracy of

± 10 cm. Compared to existing methods like trilateration, it has increased the accuracy significantly. Due to the fact that some algorithms are more accurate in x -direction than in y -direction, a future improvement will be the development of separate algorithms for x and y coordinate components. Another further step will be the combination of information of machinery acceleration and UWB distance measurement in a sensor-fusion model, including the existing method. This should provide more robust and stable results.

Bibliography

- [1] A. Chehri, P. Fortier, and P. M. Tardif, "UWB-based sensor networks for localization in mining environments," *Ad Hoc Networks*, vol. 7, no. 5, pp. 987–1000, jul 2009. [Online]. Available: <http://dx.doi.org/10.1016/j.adhoc.2008.08.007><http://linkinghub.elsevier.com/retrieve/pii/S157087050800125X>
- [2] F. Subhan, H. Hasbullah, A. Rozyyev, and S. T. Bakhsh, "Indoor positioning in Bluetooth networks using fingerprinting and lateration approach," *2011 International Conference on Information Science and Applications, ICISA 2011*, 2011.
- [3] M. Ji, J. Kim, J. Jeon, and Y. Cho, "Analysis of positioning accuracy corresponding to the number of BLE beacons in indoor positioning system," in *2015 17th International Conference on Advanced Communication Technology (ICACT)*, vol. 2015-Augus. IEEE, jul 2015, pp. 92–95. [Online]. Available: <http://ieeexplore.ieee.org/document/7224764/>
- [4] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, nov 2007. [Online]. Available: <http://ieeexplore.ieee.org/document/4343996/>
- [5] R. Mautz, "Indoor Positioning Technologies," *Institute of Geodesy and Photogrammetry*, no. February 2012, p. 127, 2012.
- [6] Nanotron, "Nanotron swarm bee er product specifications," 2018, https://nanotron.com/EN/pr_protect-php/ [2018-11-30].
- [7] A. Jimenez and F. Seco, "Comparing Decawave and Bespoon UWB location systems: Indoor/outdoor performance analysis," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, vol. 66, no. October. IEEE, oct 2016, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/7743686/><http://ieeexplore.ieee.org/document/7891540/>
- [8] A. R. Jimenez Ruiz and F. Seco Granja, "Comparing Ubisense, BeSpoon, and DecaWave UWB Location Systems: Indoor Performance Analysis," *IEEE Transactions*

on Instrumentation and Measurement, vol. 66, no. 8, pp. 2106–2117, aug 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7891540/>

- [9] F. Ramírez-Mireles, “On the performance of ultra-wide-band signals in Gaussian noise and dense multipath,” *IEEE Transactions on Vehicular Technology*, vol. 50, no. 1, pp. 244–249, 2001.
- [10] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*. Wiley, 1958.
- [11] N. Henze and B. Zirkler, “A class of invariant consistent tests for multivariate normality,” *Communications in Statistics - Theory and Methods*, vol. 19, no. 10, pp. 3595–3617, 1990.
- [12] A. Trujillo-Ortiz, R. Hernandez-Walls, K. Barba-Rojo, and L. Cupul-Magana, “Hmzmvntest: Henze-Zirkler’s multivariate normality test,” 2007, <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=17931> [2018-11-22].
- [13] E. L. Melnick and A. Tenenbein, “Misspecifications of the normal distribution,” *The American Statistician*, vol. 36, no. 4, pp. 372–373, 1982.
- [14] K. Al Nuaimi and H. Kamel, “A survey of indoor positioning systems and algorithms,” *2011 International Conference on Innovations in Information Technology, IIT 2011*, no. September, pp. 185–190, 2011.
- [15] R. B. Langley, “Dilution of Precision,” *GPS World*, vol. 10, no. May, pp. 52–59, 1999. [Online]. Available: <http://www2.unb.ca/gge/Resources/gpsworld.may99.pdf>

Computational Methods for the Detection of Wear and Damage to Milling Tools

Dimitar Ninevski¹, Julia Thaler¹, Paul O’Leary¹,
Thomas Klünsner², Manfred Mücke², Lukas Hanna²,
Tamara Teppernegg³, Martin Treichler⁴, Patrick Peissl⁵
and Christoph Czettl³

¹Chair of Automation, University of Leoben,
Peter-Tunner-Strasse 25, A-8700 Leoben, Austria
automation@unileoben.ac.at

²Materials Center Leoben Forschung GmbH, Roseggerstraße 12,
A-8700 Leoben, Austria

³Ceratizit Austria GmbH, Metallwerk-Plansee-Straße 71,
A-6600 Reutte, Austria

⁴Tool Consulting and Management GmbH, Technologiepark 3,
A-8510 Stainz, Austria

⁵Ro-Ra Aviation Systems, Gewerbepark 8,
A-4861 Schörfling, Austria

Abstract

The current paper presents a new computational approach to detect wear and damage to milling tools’ cutting edges. The proposed approach is independent from exact information on tool-workpiece interaction conditions and only requires that they remain constant for compared milling operations. Additionally, the approach was thoroughly tested on time-series data obtained from an industrial-scale milling process, instrumented by commercially available instrumentation equipment, during which 18 identical parts were milled. The time-series data contains the bending moments in the x and y directions as well as the torque and tension acting on the milling tool. Some measures used are systematic in nature, based on shape, rotation and work needed for milling, whereas others are statistical in nature, describing the change in the

distribution of the data. All of the measures proposed in the current work are relative and mutually invariant, meaning they address different information content of the data independently. A comparison of the mentioned measures with the real-world damage evolution of the milling tool's cutting edges for multiple produced parts yielded consistent results and suggests a high potential for practical tool damage detection in industrial production.

Keywords: Condition monitoring, Milling tool damage, Time-series sensor data.

1 Introduction

Today, the unexpected failure of cutting edges of milling tools in industrial-scale production processes of metal parts is an issue of high relevance to process safety and productivity. During milling, the cutting edges of the tools involved are unavoidably subject to deterioration. This can lead to unsatisfactory surface quality of the manufactured workpiece and in some cases, even lead to damage of the milling machine. The associated costs can be reduced by introducing Condition monitoring (CM) of the cutting tools. The primary goal is the diagnosis of the used tool's damage state at a given point in time within the machining process, in order to facilitate decision-making on modifications to the milling process parameters [1, 2] or replacement of tooling.

A possible approach to CM is based on automated but sporadic visual inspection of the tool's cutting edge contours aided by automated image recognition techniques [3]. The discontinuous nature of this CM approach inherently limits the productivity of industrial production processes, which ought to be performed without interruptions.

Another important class of CM approaches for milling tools involves analytic modelling of the cutting forces and bending moments that change with the progressing deterioration of a milling tool's cutting edges during milling [4]. These model-based approaches enable the indication of significant deviations from the undamaged tool state [5]. Unfortunately, they are also associated with considerable computational effort to determine optimal model parametrization. Additionally, this requires knowledge of the tool-workpiece interaction, such as the axial and radial penetration of the workpiece by the milling tool. These are not readily provided with the required μm -scale resolution by conventional CAD-CAM software solutions. Furthermore, the model parameters are a function of the applied cutting speed and the feed per tooth [6]. This complicates the monitoring during the production of complex workpieces with changing cutting conditions.

Direct monitoring techniques are based on the analysis of data acquired from additional sensors used to instrument the machine. This permits diagnosis of the tool condition during the milling process. Instrumented collet chucks provide a means of measuring as close to the cutting process as possible, ensuring a higher information content of the sensor signal wrt. the process state [7]. In general, machining experiments in a controlled laboratory environment often apply custom or specialized sensor solutions [1, 8]. These may deliver more detailed information on cutting edge temperatures, particular relevant when monitoring surface coatings [9]; however, they often

require time-consuming off-line data analysis [8]. Instrumented tool holders have become available for industrial processes, which yield information on the temporal evolution of the distribution of forces, bending moments, torque [10] and cutting edge temperature [8].

Machine learning approaches have been applied to the evaluation of data from instrumented tool holders. They are able to estimate a milling tool's damage state from two-dimensional distributions, i.e. polar plots of bending moment without the prior knowledge of the tool-workpiece interaction conditions, but require a large set of training data to train the involved algorithms [11]. Especially in the case of industrial milling applications with small workpiece batch sizes, these training data may not be available for new tool-workpiece combinations.

Pro-Micron GmbH & Co. KG has applied for, and been awarded the US Patent [12]. The claims of the patent, i.e., the legally relevant portion, define exactly what has been patented. Here they describe the device and some parameters that may be derived from the acquired data. Starting with *claim 3*, they mention the possibility of deriving parameters relating to the shape and temporal changes in the form of the measured curves. However, few to no details are provided on which characteristics are relevant and how they may be computed. The new computational methods presented in this paper go far beyond those inferred in the Pro-Micron patent or otherwise available [8, 10].

The organization and main contributions of this paper are as follows:

- Section 2 describes in detail the experimental work performed to acquire the data used in this paper.
- Section 3 describes the preprocessing of the data. Whereby, Figure 3 shows a typical *flower pattern*, such patterns are also presented by Pro-Micron. Here the representation has been enriched with the temporal nature of the data. More significantly the data is resampled in a polar coordinate system, so as to provide information on the statistical behaviour of the data. This provides a method for dealing with missing data, a ubiquitous feature of the acquired data. The statistics also provide a quantification of the uncertainty of the data and permits the calculation of confidence intervals, a new aspect in the data processing.
- Section 4 describes the data analysis performed. A probability density function for the cutting action is computed, see Figure 4. This yields new information as to the location on the tool where the statistically dominate portion of the work is being performed. This distribution changes over time without any significant change in the shape of the flower, see the data for parts 1...10 as shown in Figure 6. A change in the tool performance without this being reflected in any significant change to the flower pattern is an aspect not addressed in previous work. The concept of work is also used, which is a physically meaningful metric for the effort required by the tool and how it changes with use. Finally in this section, the Elliptical Fourier Descriptors are applied to obtain dimensionless relative measures for various aspects of the shape of the flower pattern.

- Section 5, describes a totally new approach to the evaluation of the data. It is based on characterizing the statistical nature of the data, not the shape of the flower pattern.
- Section 6 describes the optical inspection of the tool during the experiment. It provides some optical verification on the state of the tool as time progresses.

The final two sections discuss the results and draw conclusions from the work. The mathematical models for the progress of wear, see e.g. Figure 19, indicate that the analytical and statistical metrics provide comparable results and are consistent with the optical inspection.

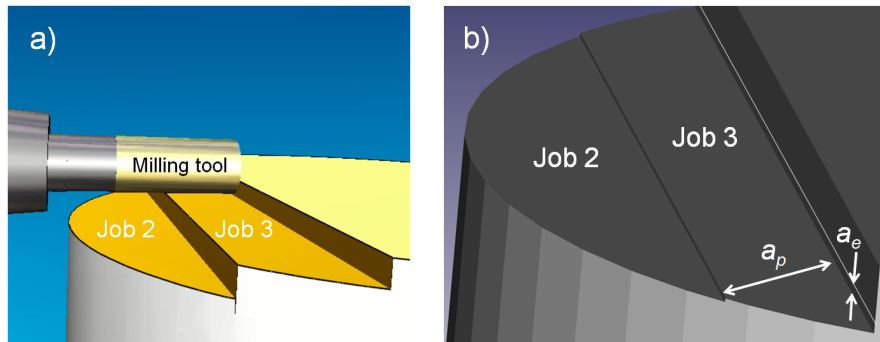


Figure 1: Schematic representation of relative spatial arrangement of the tool and workpiece. (Left) Shows the cutting orientation of the tool and (Right) defines the cutting parameters: a_p in the axial- and a_e in the lateral-direction, relevant for Job 3.

2 Experimental Data Acquisition

The data analysed in the current work was attained in an industrial-scale milling process of 18 cylindrical workpieces of a diameter of 204 mm made of Ti6Al4V. Portions of the milling operations necessary to produce the final part were performed with a commercially available wireless instrumented tool holder [13]. Details of the tooling and instrumentation can be found in [8, 9]. The occurring cutting force in spindle axis direction, the torsion around the spindle axis direction, as well as the bending moments in two perpendicular directions x and y , oriented normal to the milling tool's rotational axis were acquired at a sampling rate of 1600 Hz. The milling tool used in all these instrumented milling operations was a cylindrical end mill, made of WC-Co hard metal hard-coated with a TiAlN-based coating, with four cutting edges and a diameter of 16 mm.

The applied values of cutting speed and feed per tooth were 90 m/min and 0.125 mm, respectively. The instrumented milling operations were divided in four different jobs, with their respective numbering in the nomenclature of the production program being 2, 3, 4 and 5, see Figure 1. Note that there were two pairs of jobs with identical geometrical tool-workpiece interaction conditions, namely jobs 2 and 4, and jobs 3 and 5, see Figure 1b). Each job was subdivided in eight

linear cut lanes. The radial immersion of the tool into the workpiece a_e , see Figure 1b), was 1.5 mm for all the cut lanes in each of the jobs, except for the first lanes of jobs 3 and 5, in which a_e was smaller with a nominal value of 0.9mm. Note, that there is some uncertainty in the actual values of a_e for the first cut lanes in each of the jobs, due to possible differences in the workpiece contours stemming from the previous workpiece shaping procedure. Therefore, the first cut lanes of the individual jobs have not been considered in the following data analysis. To reduce the complexity of the data analysis, it was performed only for data from job 3, with its quasi-constant axial depth of cut a_p , see Figure 1b), of 22mm for a portion of 144mm of its linear path length of 178mm. To compare milled lanes with equivalent tool-workpiece interaction conditions, the data analysis procedure was performed comparing cut lanes with equal lane index for job 3 for the 18 milled workpieces. After each individual workpiece had been machined by the four described milling jobs, digital microscopy (VHX5000, Fa. Keyence) was applied to characterize the damage status of the four involved cutting edges.

One important thing to note about the data acquisition is that unfortunately, there are brief, temporally random interruptions of the transmission, leading to missing data. This is due to the harsh electromechanical environment in which the equipment is being operated. Later in the data handling, steps are taken to circumvent this issue.

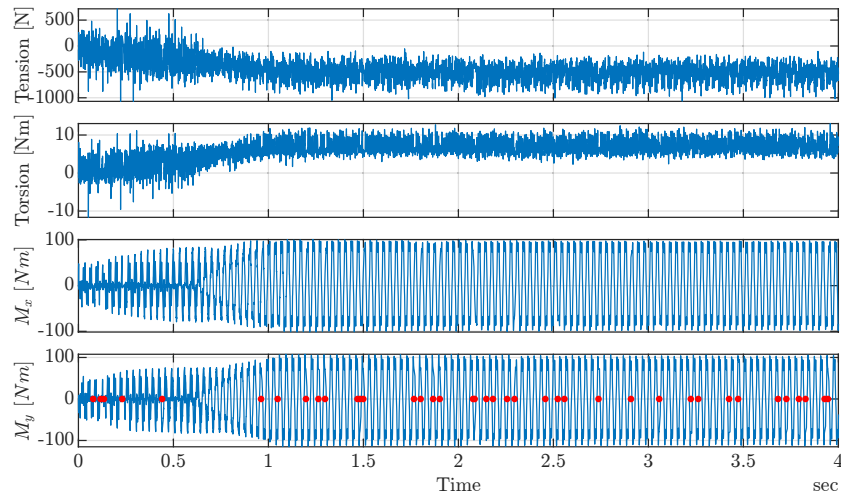


Figure 2: The first 20% of a complete time-series for one cut lane as acquired from the machine. The data is sampled at $1.6kHz$ and a typical data set has $m = 15000 \dots 20000$ samples depending on the length of the lane being cut. Here the *cutting-in* process can be seen, i.e., the time where the milling tool has not reached the phase with constant a_p , see Figure 1b). A similar phase can be observed at the end of the lane. This data set is from Part-1, Job-3 at Lane-2. The red dots indicate points where there was a brief loss of data, due to interference of the transmission.

3 Data pre-processing

The real-time data acquired from the machine has four relevant channels: tension and torsion experienced by the tool, as well as the bending moments in the x and y directions, denoted by m_x and m_y respectively. The data is sampled at 1600Hz and a typical data set has $m = 15000 \dots 20000$ samples depending on the length of the lane being cut. The first 20% of a complete time-series¹ for one cut lane, as acquired from the machine, is shown in Figure 2.

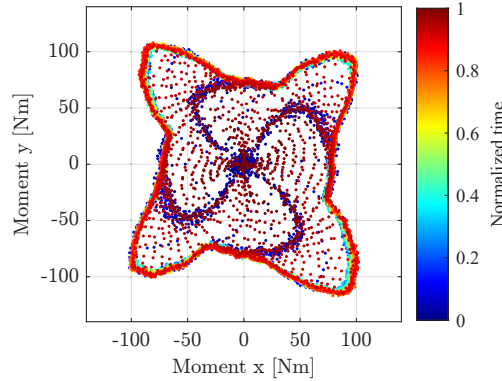


Figure 3: Flower pattern for the time-series data shown in Figure 2. The colour of the scatter plot is used to represent normalized time when the data was created during the cutting process. The arms apparently spiraling into and out of the center are associated with the cutting-in and cutting-out phases.

The task now is to extract information from this data which yields relevant knowledge about the wear and possible damage to the tool. This has been classically approached by analysing the so called *flower pattern* of the moment data. The flower pattern is created by simply using the x - and y -moment values as coordinates for a scatter plot. In the example shown here, see Figure 3, the flower has been extended to represent the time when the value occurred as colour in the scatter plot. This makes the run-in and run-out processes at the start and end of the milling process visible.

The flower pattern is more suitable for qualitative evaluation of the data; since, with the structure of the data in this form it is difficult to extract more detailed information.

The random locations of missing data in the time-series make the direct quantitative comparisons between lanes difficult: for example, FFTs with missing data lose their numerical efficiency: since, given n points, an FFT has the computational efficiency $O(n) = n \log n$, whereas with missing points the pseudo-inverse [14] will require approximately $O(n^3)$. Furthermore, the flower pattern does not deliver any information of the density of the sampled points. It is necessary to remember that there is no direct measurement of the angle of rotation of the tool. The values for m_x and m_y are used to determine the angle; however, there may be local retarding of the angular

¹Only the first 20% is shown, since the inner structure of the data is not visible if a complete data set is plotted.

velocity with respect to the cutting moment. This will yield a non-uniform spacing of the samples in terms of angles, which will become more evident later.

Consequently, a new approach is taken here, where the data is pre-processed, based on automatically segmenting and re-sampling the data in a polar coordinate frame.

3.1 Segmentation and resampling

The segmentation is relatively simple, the unsteady regions at the start and end of data, corresponding to the cutting-in and cutting-out, are detected and removed. Local descriptive statistics are computed, these permit the automatic detection of the stable cutting region within the data. The cropped data set is considered to represent the stable milling process suitable for characterisation of tool wear.

Since the milling process is a rotary process, it is fully cyclic in 360° (2π); consequently, the vectors of data can be converted to polar coordinates without loss of information or generality, i.e.,

$$\mathbf{m}_x, \mathbf{m}_y \mapsto \mathbf{m}_\phi, \mathbf{m}_r. \quad (5.35)$$

The data is now binned, as with a histogram, but into k equi-angular segments. The desired polar resolution δ_ϕ is defined via the number of bins k in 360° , i.e.,

$$\delta_\phi = \frac{2\pi}{k}, \quad (5.36)$$

here, for convenience, $k = 360$ has been chosen, yielding $\delta_\phi = 1^\circ$. In this manner a vector of uniformly spaced bin edges $b_e = \delta_\phi [0, 1, \dots, k]$ is computed. All the data channels are binned into the corresponding b_e . This data can now be used to compute the properties of the data and their statistical variations in a polar frame. A major advantage of this re-sampling process is, that the data is now equally spaced in ϕ , and has the same number of samples for each measurement and channel, independent of the length of the original data sequence.

4 Data analysis

The preprocessing provides all the necessary data in a structured manner for analysis.

4.1 Median curve and its properties

The first analysis consists of computing four properties for each element of the structure $\mathcal{S}(k)$:

1. The median angle $\phi_m(k)$ and median moment $m_m(k)$ for each of the k segments. These can be used to create a polar plot for the moment as a function of angle, similar to the flower pattern. However, sampled to have a constant number of samples per revolution.

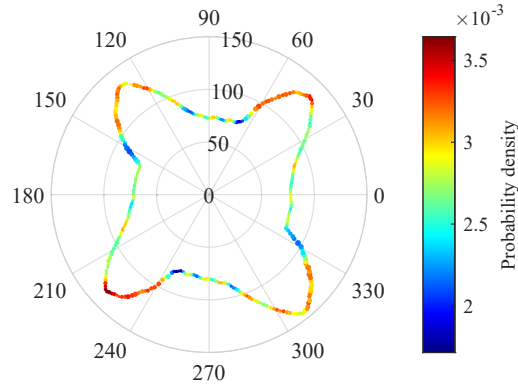


Figure 4: Polar plot of the segmented and resampled moments in x and y . Note, the colour indicates the probability density of a measurement at that specific angle. That is the density of data points at a specific angle. The width of the line indicates the interquartile range at each angle, in this case it is almost uniform. This is for the data shown in Figure 2

2. With a constant sampling frequency f_s and a uniform rotational speed, a constant number of samples $n(k)$ per angular segment would be expected. However, when $n(k)$ is determined, it is observed that they are not constant. The probability density as a function $PDF(k)$ of angle, can be computed by normalizing $n(k)$ so that the integral around the 360° is one.
3. Furthermore, the 25% and 75% quartiles, $q_{25}(k)$ and $q_{75}(k)$ are computed for each segment. Additionally, using these values the interquartile range (IQR) can be calculated as $IQR(k) = q_{75}(k) - q_{25}(k)$.

These four quantities can be represented in a single polar scatter plot, see Figure 4. With this data set, the interquartile range indicates that there is a low uncertainty in the observed moments over the full revolution. However, the angular PDF reveals a non-uniform loading of the tool during a rotation. This is the first new result which can be obtained with the proposed computational technique. The fact, that the PDF is not uniform for each of the four flutes indicates that they do not experience equal loading.

4.2 Torsion and tension

The torsion and tension data is resampled in the same manner as the moments and the resulting information is added to the structure $\mathcal{S}(k)$. The angular PDF for torsion and tension will be the same as for the moments, since the number of samples is the same. Consequently, only the median curve and the respective interquartile range (IQR) are computed and visualized, see Figure 5. The torsion and tension are subject to much higher uncertainties as can be seen from the figures. Consequently, they are not used for further analysis.

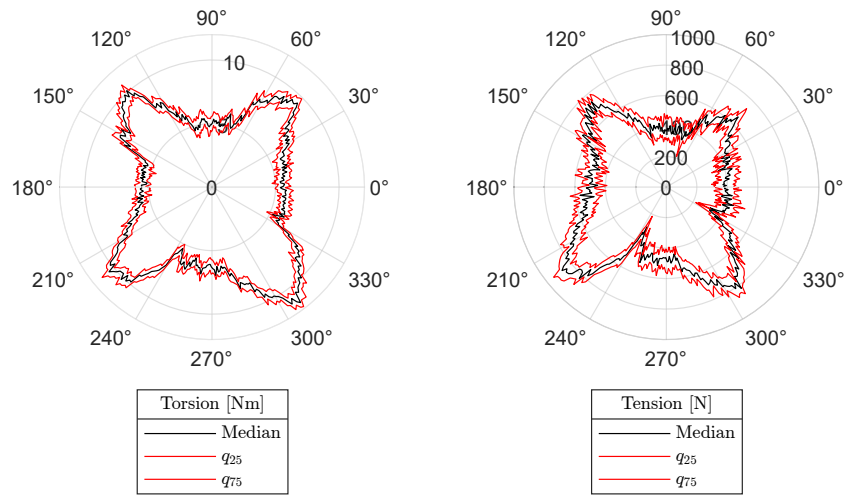


Figure 5: Torsion (Left) and tension (Right) from the data binned into angular portion $\Delta \phi = 1^\circ$, i.e. 360 bins per revolution. Each figure shows the median values together with the Q25 and Q75 quantiles.

4.3 Comparison across multiple parts

In the data acquisition for this project 18 identical parts were manufactured in sequence. This now makes it possible to compare the performance of the milling tool, for exactly the same operation, across the manufacturing of multiple parts.

To enable a first comparison, the median moment curves with probability density functions are shown for job 3 and lane 2 for all the 18 parts in Figure 6. There are some obvious patterns which can be observed immediately:

1. The orientation of the pattern rotates with an increasing number of parts being milled. This would indicate, that the median angle of cutting engagement is changing.
2. The area enclosed by the median moment curves grows larger with increasing part number. This will yield a relative measure for the increased work required to produce the parts. Moment has the dimension Nm, which is the same as force times distance, i.e., work. Consequently, the integral of the area enclosed by the curve is proportional to the work required for the operation. Normalization with respect to the work required for the same operation with a new tool yields a relative measure for degradation of the tools performance. This is a practical measure, since it does not need to know anything about the cutting profile, it is only required to be the same from one part to the next.
3. The PDF for the measurements around the tool spreads more evenly as time proceeds.

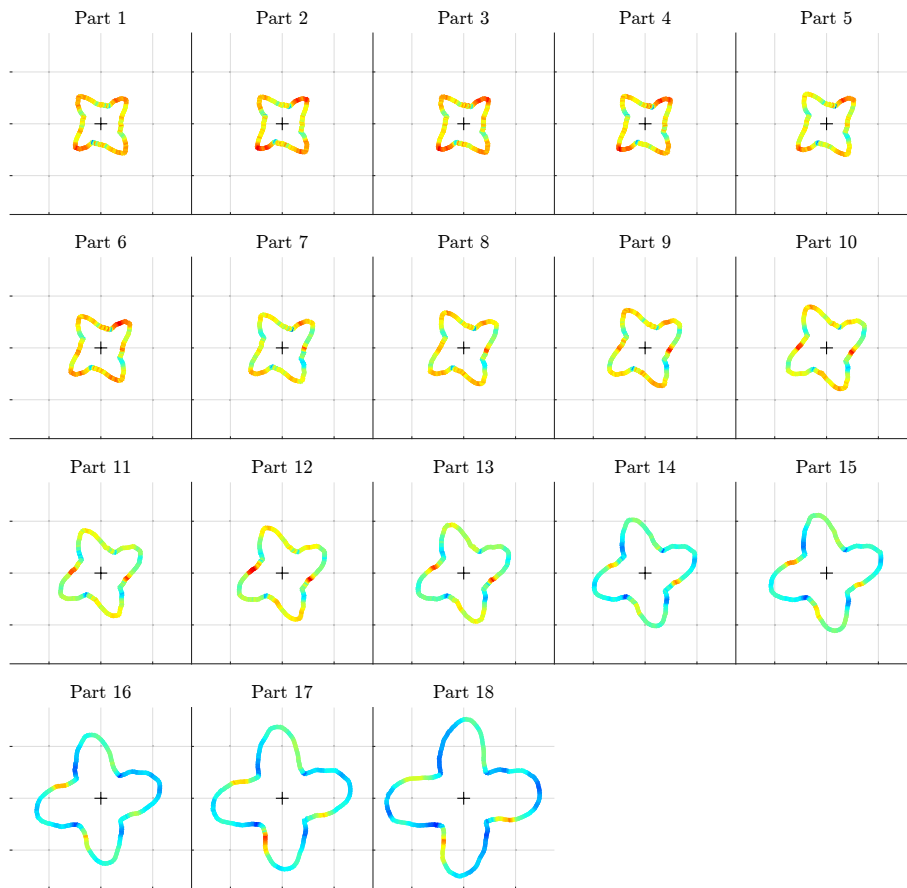


Figure 6: The median curves for all 18 parts manufactured. The data is in all cases for job 3 lane 2, so that it is compatible with previous figures. It can be observed that: the pattern rotates with the increasing number of parts produced, the area enclosed by the median curves grows larger and the pdf spreads more evenly.

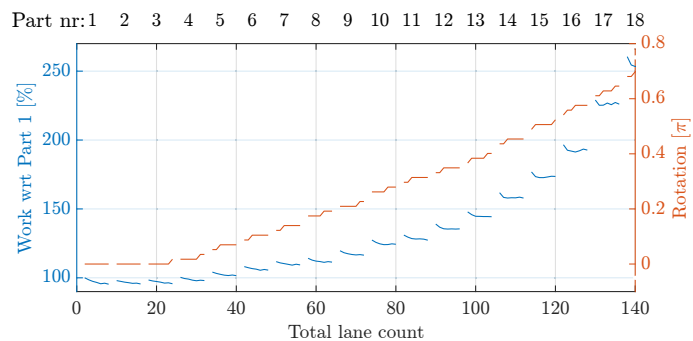


Figure 7: Left scale: Work required to mill the lane relative to part one. Right scale: the relative rotation of the median curve. There are eight lanes milled for each of the 18 parts. The first lane in each part may have some variation due to misalignment of the metal block relative to the tool path.

4.4 Determining rotation and relative work

If part 1 is defined to be a reference, i.e. a part produced with a new tool, then its median moment $m(\phi)$ as a function of ϕ , can be used as a reference r_{ref} vector, to which the measurement data from all other parts are compared. To determine the relative rotation and scaling, the circular convolution of r_{ref} with r_k , the radii obtained from the other measurements is calculated. The location of the maximum peak in the circular convolution corresponds to the group delay between the two data sets. Since the data is perfectly cyclic and circular convolution is being used, the group delay is directly proportional to the relative rotation of the patterns. Additionally, the magnitude of the peak normalized by the 2-norm of r_{ref} yields the ratios of the areas enclosed by the two curves. This is a relative measure with respect to the first part for the work required to produce the subsequent parts.

The results of this calculation for all 18 parts and all lanes for job 3 is shown in Figure 7. These measures yield qualitative relative values to compare the condition of the tool from the production of one part to the next.

4.5 Fourier Spectrum and Fourier Elliptical Descriptors

The resampling of the data into polar coordinates ensures that the data is perfectly cyclic over 360° . Consequently, discrete Fourier spectra can be computed without being corrupted by spectral leakage and Gibbs errors [15]. There is no need to performing windowing [16] prior to computing an FFT. The magnitude of the Fourier spectrum is shown in Figure 8, for the same data as in Figure 2. This spectrum yields a number of insights:

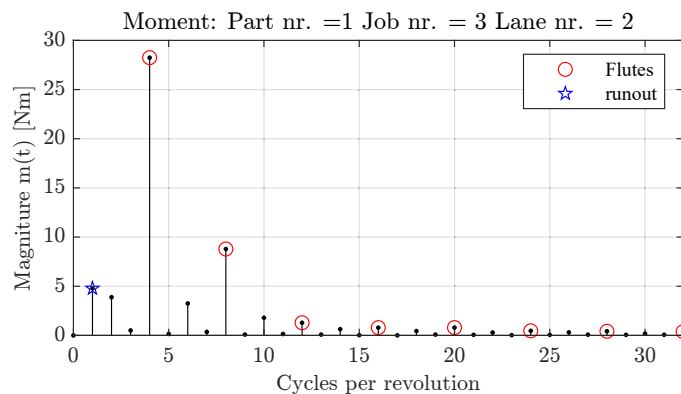


Figure 8: The first $n = 32$ harmonic components of the moment $m(t)$. This spectrum is free from Gibbs and spectral leakage error, since the data is perfectly cyclic. Two main issues can be determined directly from the spectrum. 1) The fourth harmonic is due to the mill having four flutes. This harmonic and multiples of it (i.e. 4, 8, 12,...) describe the shape of the moment in terms of the Elliptical Fourier Descriptors EFD. It is a relative measure and independent of scale and rotation. 2) The first harmonic is that portion that changes once per revolution.

1. The magnitude of the Fourier spectrum is independent of phase, when no windowing is being applied. Consequently, these results are independent of the rotary position of the tool and/or the pattern.
2. The dominant fourth harmonic is associated with the mill having four flutes. Mills with different numbers of flutes should be evaluated correspondingly.
3. The ratio of the 4th harmonic to its overtones is a description of the shape of the median curve; this is known as an elliptical Fourier descriptor [17]. The scalar value it yields is dimensionless and invariant with respect to scale and rotation of the pattern. It yields a relative measure to compare the patterns.
4. The first harmonic is the change in the moment once per revolution, this measure corresponds to the runout of the tool. A different term may be more appropriate here, since this is the runout observed in the bending moments, not in its physical motion.

4.6 Comparison of tool runout

As seen above, the elliptical Fourier descriptors of the median moment curves yield a measure for tool runout as observed in the bending moment. This was computed for all lanes for job 3 in all the manufactured parts. The results are shown in Figure 9. Note, part 16 has an exceptional runout. There is need for some care when interpreting this result: after each part the tool was removed for optical inspection, then replaced in the collet. The tool may have some offset when mounted for part 16; nevertheless, this descriptor does provide a means of detecting exceptional bending moment runout.

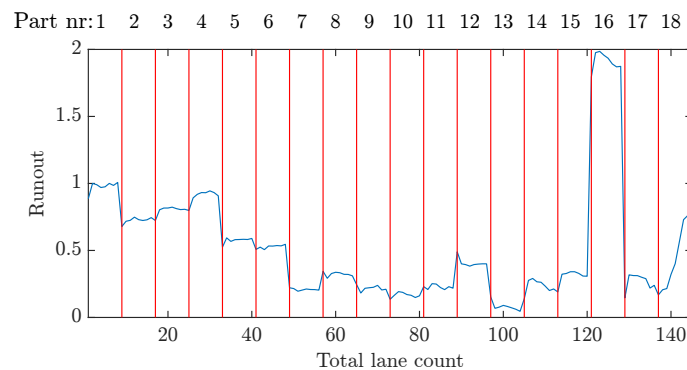


Figure 9: The first harmonic (runout) as a function of part and lane milled. Note: part 16 has an exceptional runout.

4.7 Shape factor

The elliptical Fourier descriptors offer the possibility of computing scale and phase independent shape descriptors. Here it is desirable to define a shape descriptor that is independent of the relative work and rotation of the pattern, but nevertheless is characteristic for changes in the shape of the pattern. This gives the measure an orthogonality to previous measures, i.e., it is accessing a different information

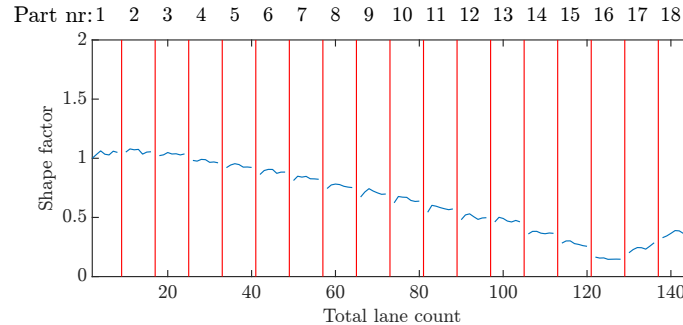


Figure 10: The ratio of the power in the fourth harmonic to the power in the first 7-overtones. This is a shape factor which is scale and phase invariant. As can be seen, the signal becomes more sinusoidal up to part 16. See Figure 6 for a comparison of parts 1 to 18.

Here the ratio of the power in the primary frequency - the 4th harmonic since the mill has four flutes - to its first seven overtones is proposed. The results of this calculation are shown in Figure 10 for all lanes and parts. A systematic tendency towards lower harmonics is shown up to part 16. The lower harmonics suggest that the curves are becoming more sinusoidal as more lanes are milled; this is demonstrated in Figure 11 for parts 1 and 18. This shape factor is invariant with respect to magnitude and phase of the pattern.

The change in runout at part 16 and the change in shape factor suggest not only runout caused by the removing of the tool from the collet, as mentioned in 4.6, but also some damage to the tool at this point in time. The methods presented here can detect both these sources of runout, but further investigation is required in order to distinguish between them based on the data available.

5 Statistical changes in the data

The focus up until now has been on systematic changes in the median curve pattern as more and more milling time proceeds. The goal here is to view changes in the statistical behaviour of the acquired data.

The first observation is that there is a relationship between the bending moment and torsion. On the left hand side of Figure 12 the plots of moments vs. torsion are presented for parts 1 and 13: note, the distributions of the points has changed significantly. Principal component analysis (PCA) [18] was applied to the moment and torsion data as a mean of capturing the maximum

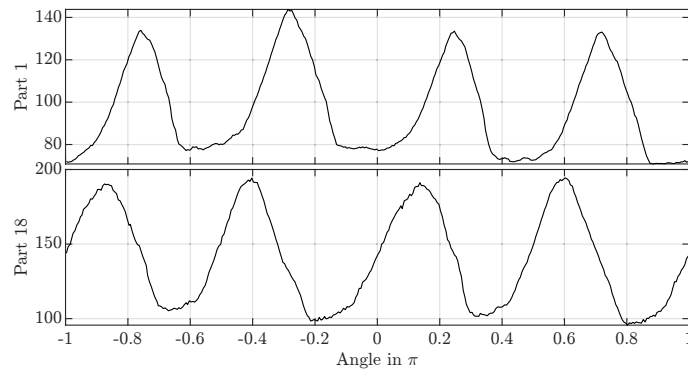


Figure 11: Moments for parts 1 and 18. The rotation of the flower pattern is seen here as a phase shift. Additionally, the curve is more sinusoidal for part 18, indicating lower harmonic components.

amount of information in these signals. The histograms for the first principle components are shown on the right hand side of Figure 12. Note, the histograms are bimodal, it can be seen that there is a significant change in the distributions.

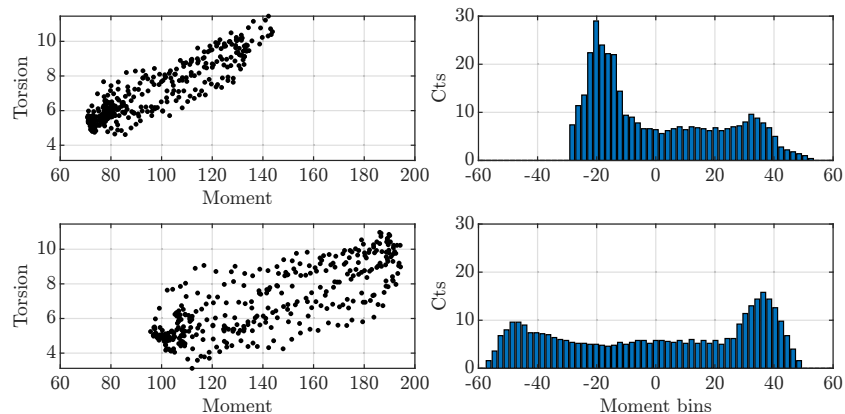


Figure 12: Change in performance as determined by statistics. (Left) Bivariate distribution of moment and torsion for parts 1 and 13. (Right) The corresponding histograms when projected onto the first principal component.

Performing the same computation for all lanes, for job 3, across all parts yields a data-set that can be viewed as a heatmap, see Figure 13. Each column of the heatmap corresponds to the histogram for a single lane; in this manner the significant change in statistical behaviour as a function of time can be clearly seen. The distribution becomes wider and the bi-modal parameters change. This reveals a significant change from part 12 to 13.

Two measures are calculated from the histograms in Figure 13: i.e., the width of the respective distribution, see Figure 14 and the magnitudes of the two bi-modal dominant peaks, see Figure 15. The width of the histogram is determined as the distance from the left to the right outer edges.

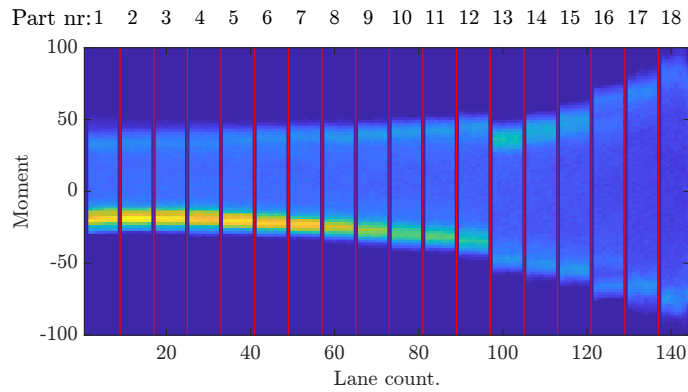


Figure 13: Histograms for the first principal component of moments and torsion as a function of part and lane milled. Part number noted on the top and lane number at the bottom. Note: there is a major change in the distribution at part 13. See Figure 12 for a comparison of part 1 and 13.

The width of the distributions follows an exponential function, see Figure 14. Consequently, the data is modelled by Equation 5.37, yielding the results: $a = 1.4002$, $b = 0.2439$ and $c = 72.5046$. Given the parameterized model, the $3dB$ break point for the curve was computed, yielding $n_{3dB} = 12.1$. The $3dB$ point is the level where the power required to perform a task had doubled. This is a commonly used criterion in digital signal processing [19]. It is proposed here as a means of defining an appropriate decision level as to when significant wear has occurred.

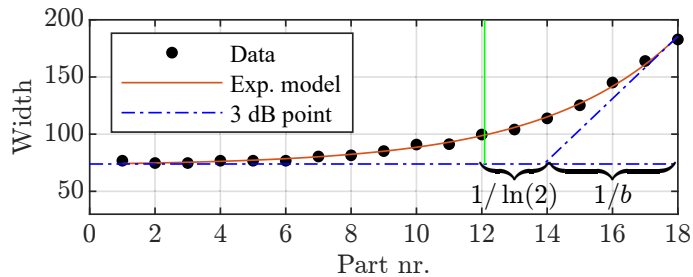


Figure 14: Width of the respective histogram distribution (data) as a function of the number of parts manufactured, an exponential model for the data and the determination of the $3dB$ break point, $n_{3dB} = 12.1$.

The break point between parts 12 and 13 is further collaborated by determining the magnitude of the two dominant peaks of the bi-modal histograms, see Figure 15, which converge at the same location.

6 Optical inspection

After the milling of each part the tool was removed for optical inspection and photographic documentation, an example of such an image can be seen in Figure 16. The goal is to provide inde-

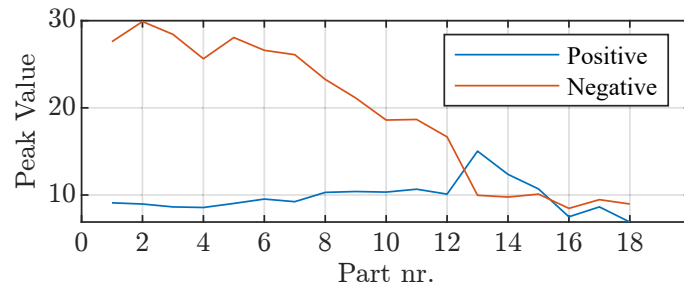


Figure 15: Magnitudes of the two dominant peaks of the bimodal histograms as a function of part number. Note: the convergence of the two values close to part 13. See Figure 12 for a comparison of distributions for parts 1 and 13.

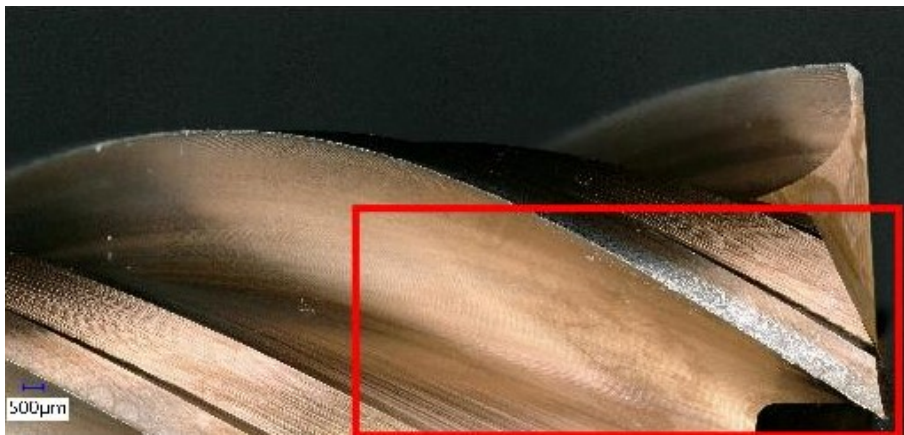


Figure 16: An image of the milling tool with the region of interest marked with a red rectangle.

pendent quality measures to which the results of the newly proposed methods can be compared. For each of the four cutting edges the width of the edge wear mark was measured, the results are shown in Figure 17. Note that the mentioned wear mark width represents the maximum extent of a form of wear, including homogeneous cutting edge blunting and breakouts at the cutting edge. It was observed that this edge wear width is well modelled as an exponential of the form,

$$w(n) = a \exp^{bn} + c, \quad (5.37)$$

where n is the part number. This exponential model was selected, since both the data and its first derivatives are well modelled by exponentials. This is a characteristic for selecting such a model. This model is also shown in Figure 17. For a selected number of parts, which are most relevant for the discussion of our results, the zoomed images of the cutting edges are shown in Figure 18. Note, that microscopic breakouts were observed at the cutting edge of the milling tool for part 12 that increase in size with increasing number of milled parts.

7 Discussion

To enable a comparison of the results of the manual optical measurement of the edge-wear-width (from Figure 17); the work² required per part manufactured (from Figure 7) and the width of the histograms of the first principal component of moment and torsion (from Figure 13), are shown together in Figure 19. As can be seen the results derived from the statistics and from the relative work correspond highly. The relative work curve is somewhat smoother; this is to be expected, since the correlation method used to determine these values is optimal in terms of noise propagation.

There is a strong correlation between the optical measurements of edge wear width and the results obtained from the data processing. Whereby, the relative work and the statistical measures give earlier indications of changes to the cutting edge damage state than the manual inspection of the microscopy images of the cutting edges. This is due to the fact that the edge wear width is difficult to determine reliably for small extent of wear due to volatile workpiece material transfer features on the tool surface with similar reflectivity as hard metal tool material and the helical shape on the tool's cutting edge that is associated with a limited depth of field in the microscopy images.

The rotation of the flower-pattern, the relative work and the Fourier descriptors are all mutually invariant. That is, they provide independent measures for the damage state of the tool and describe a trajectory in 3D space characteristic for the wear of the tool. This will provide a better view for the estimation of anomalous behaviour that can be achieved with a single parameter.

The new data processing also yields the probability density function for the bending moment

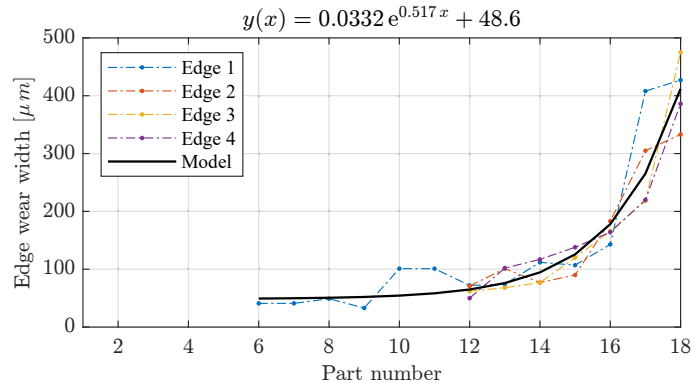


Figure 17: Edge-wear-width as measured from the images of the tool after manufacturing each part. Note: for small wear the width could not be reliably measured also due to some uncertainty introduced by volatile workpiece material transfer features on the tool surface with similar reflectivity as hard metal tool material. The wear width as a function of part production appears to be well-modelled by an exponential function.

²Note: torque has the dimension Newton-meters [Nm], the same as physical work. Consequently, the integral over 360° in the polar plot yields a measure for the physical work required to manufacture a part.



Figure 18: Digital images of the milling tool after having milled different amounts of workpieces. Black arrows indicate the positions of breakouts at the cutting edge that influence the distributions of the sensor signals discussed in the current work. The increase in wear and damage feature size is visibly increasing with the number of workpieces milled, which is expected.

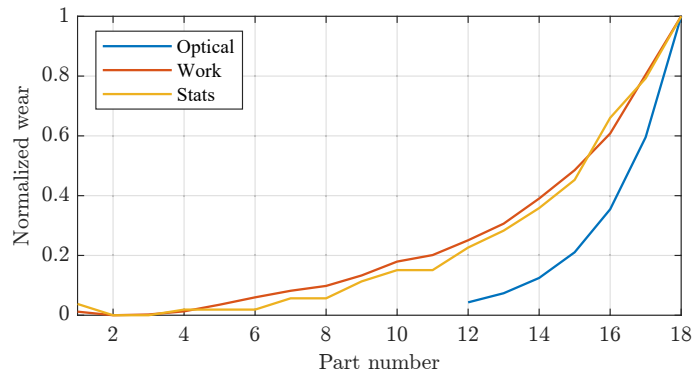


Figure 19: Optical: normalized edge-wear-width as manually measured from the images. Work: normalized relative work required to manufacture a part, measured from the bending moments. Stats: The width of the respective histogram of the first principal components.

around the flower pattern and how this changes with the deterioration of the tool, see Figure 6. This needs further investigation as it would suggest that it is possible to statistically identify changes to individual flutes which have no visible effect on the pattern as a whole.

The statistical characterization of the tool performance as can be seen in Figure 13, is attractive for tool condition monitoring purposes because of the low numerical effort required to do the involved computations. The convergence of the magnitude of the two dominant peak heights in the bi-model histogram plotted in Figure 13 occurs simultaneously with the emergence of breakouts at the cutting edges shown exemplarily in Figure 18. This fact can be used as an efficient means of indication of the change in the damage mode of the tool, since breakout formation is associated

with a significant rise in the deterioration kinetics of the tool's cutting edge, compare Figure 17. Note that the proposed measure for the tools damage state indicates the system damage mode change shortly before the limit of edge wear width of $300\mu m$ is reached, above which the tool is considered not any longer suitable for cutting. This finding is interpreted to facilitate the efficient and not overly conservative use of the proposed approach for condition monitoring of milling tools.

8 Conclusion

The systematic collection of measurement data over a series of parts manufactured with the same instrumented milling tool has proved very valuable for the development of reliable measures for the damage state of milling tools.

The approach taken here relies on relative measures and does not need to have exact information of the absolute interaction of the cutter and work piece other than that, they are equal for the compared milling jobs. This is suitable for use with CNC machine tools since the repeatability of the tool's motions can be achieved. In this case the tool damage state can be compared on a path-to-path basis. Compared to other condition monitoring approaches this greatly simplifies the reliable detection of changes in the tool damage state.

The tool wear was demonstrated to be highly non-linear as a function of milling time. The different measures provided consistent results in this respect. The rotation of the flower pattern, as the number of parts manufactured increases, is the most linear measure. It also indicates that the load distribution of the cutting edges of the tool changes with progressing tool deterioration.

The new computational approaches taken to evaluate the real-time moments, tension and torsion data as measured with the used milling tool has yielded new insights into possibilities for detecting wear and the formation of breakouts at the cutting edges of milling tools. A number of mutually invariant measures have been defined, computed and successfully compared with manual visual inspection results for the tool. The resampling of the data into a polar coordinate system with a constant number of sub-divisions, has proven to be highly effective in reducing the numerical effort required to compute spectra, while eliminating spectral leakage and Gibbs phenomena.

Acknowledgements

The authors gratefully acknowledge the financial support under the scope of the COMET program within the K2 Center "Integrated Computational Material, Process and Product Engineering (IC-MPPE)" (Project No 859480). This program is supported by the Austrian Federal Ministries for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) and for Digital and Economic Affairs (BMDW), represented by the Austrian research funding association (FFG), and the federal states of Styria, Upper Austria and Tyrol. The authors also want to thank E. Hieslmayr for the performance of data handling tasks.

Bibliography

- [1] Z. Yuqing, W. Xue, Review of tool condition monitoring methods in milling processes, *The International Journal of Advanced Manufacturing Technology* 96 (05 2018). doi:10.1007/s00170-018-1768-5.
- [2] I. Marinescu, D. Axinte, An automated monitoring solution for avoiding an increased number of surface anomalies during milling of aerospace alloys, *International Journal of Machine Tools & Manufacture* 51 (2011) 349–357.
- [3] R. G. Lins, P. R. M. de Araujo, M. Corazzim, In-process machine vision monitoring of tool wear for cyber-physical production systems, *Robotics and Computer-Integrated Manufacturing* 61 (2020) 101859. doi:<https://doi.org/10.1016/j.rcim.2019.101859>.
URL <https://www.sciencedirect.com/science/article/pii/S0736584519301838>
- [4] S. Engin, Y. Altintas, A. Fellow, Generalized modeling of milling mechanics and dynamics: Part i - helical end mills, American Society of Mechanical Engineers, Manufacturing Engineering Division, MED 10 (01 1999).
- [5] M. Nouri, B. K. Fussell, B. L. Ziniti, E. Linder, Real-time tool wear monitoring in milling using a cutting condition independent method, *International Journal of Machine Tools and Manufacture* 89 (2015) 1–13. doi:<https://doi.org/10.1016/j.ijmachtools.2014.10.011>.
URL <https://www.sciencedirect.com/science/article/pii/S0890695514400014>
- [6] H. Wang, J. Wang, J. Zhang, K. Tao, D. Wu, Identification and analysis of cutting force coefficients in the helical milling process, *Journal of Advanced Mechanical Design, Systems, and Manufacturing* 14 (2020) JAMDSM0020–JAMDSM0020. doi:10.1299/jamdsm.2020jamdsm0020.
- [7] R. Corne, C. Nath, M. E. Mansori, T. Kurfess, Enhancing spindle power data application with neural network for real-time tool wear/breakage prediction during inconel drilling, *Procedia Manufacturing* 5 (2016) 1–14, 44th North American Manufacturing Research Conference, NAMRC 44, June 27-July 1, 2016, Blacksburg, Virginia, United States. doi:<https://doi.org/10.1016/j.promfg.2016.08.004>.
URL <https://www.sciencedirect.com/science/article/pii/S235197891630004X>
- [8] A. Nemetz, W. Daves, T. Klünsner, C. Praetzas, W. Liu, T. Tepperneegg, C. Czettl, F. Haas, C. Bölling, J. Schäfer, Experimentally validated calculation of the cutting edge temperature during dry milling of ti6al4v, *Journal of Materials Processing Technology* 278 (2020)

116544. doi:<https://doi.org/10.1016/j.jmatprotec.2019.116544>.

URL <https://www.sciencedirect.com/science/article/pii/S0924013619305175>

- [9] A. Nemetz, W. Daves, T. Klünsner, W. Ecker, J. Schäfer, C. Czettl, T. Antretter, Cyclic heat-up and damage-relevant substrate plastification of single- and bilayer coated milling inserts evaluated numerically, *Surface and Coatings Technology* 360 (02 2019). doi:10.1016/j.surfcoat.2019.01.008.
- [10] Pro-Micron, Wireless sensors, <https://www.pro-micron.de>, accessed: 2021-03-25 (2021).
- [11] S. Goetz, D. Schraknepper, G. Faustini, T. Bergs, Process monitoring in end milling using polar figures, *Journal of Machine Engineering* 20 (2020) 95–105. doi:10.36897/jme/119690.
- [12] L. C. Rainer Wunderlich, Martin Lang, Method for setting and/or monitoring operating parameters of a workpiece processing machine (1 US Patent: US9864362B2, 2018 Jan 1.).
- [13] Pro-Micron, Spike, <https://www.pro-micron.de/spike/>, accessed: 2021-04-20 (2021).
- [14] G. H. Golub, C. F. Van Loan, *Matrix Computations*, 3rd Edition, The Johns Hopkins University Press, 1996.
- [15] R. Bracewell, *The Fourier Transform and its Applications*, 2nd Edition, McGraw-Hill, 1986.
- [16] F. Harris, On the use of windows for harmonic analysis with the discrete fourier transform, *Proceedings of the IEEE* 66 (1978) 55–83.
- [17] P. Lestrel, *Fourier Descriptors and their Applications in Biology*, Cambridge University Press, 1997. doi:10.1017/CBO9780511529870.
- [18] I. Jolliffe, Springer-Verlag, *Principal Component Analysis*, Springer Series in Statistics, Springer, 2002.
- [19] A. V. Oppenheim, R. W. Schaffer, *Digital signal processing*, Prentice-Hall Englewood Cliffs, N.J, 1975.

Sensor-based Particle Size Determination of Shredded Mixed Commercial Waste based on two-dimensional Images

Lisa Kandlbauer¹, Karim Khodier², Dimitar Ninevski³,
Renato Sarc¹

¹Department of Environmental and Energy Process Engineering;
Chair of Waste Processing Technology and Waste Management,
Montanuniversitaet Leoben
A-8700 Leoben, Austria

²Department of Environmental and Energy Process Engineering;
Chair of Process Technology and Industrial Environmental Protection,
Montanuniversitaet Leoben
A-8700 Leoben, Austria

³Department Product Engineering; Chair of Automation,
Montanuniversitaet Leoben,
A-8700 Leoben, Austria

Abstract

To optimize output streams in mechanical waste treatment plants dynamic particle size control is a promising approach. In addition to relevant actuators – such as an adjustable shredder gap width – this also requires technology for online and real-time measurements of the particle size distribution. The paper at hand presents a model in MATLAB which extracts information about several geometric descriptors – such as diameters, lengths, areas, shape factors – from 2D images of individual particles taken by RGB cameras of pre-shredded, solid, mixed commercial waste and processes this data in a multivariate regression model using the Partial Least Squares Regression (PLSR) to predict the particle size class of each particle according to a drum screen. The investigated materials in this work are lightweight fraction, plastics, wood,

paper-cardboard and residual fraction. The particle sizes are divided into classes defined by the screen cuts (in mm) 80, 60, 40, 20 and 10. The results show assignment reliability for certain materials of over 80%. Furthermore, when considering the results for determining a complete particle size distribution – for an exemplary real waste – the accuracy of the model is as good as 99% for the materials wood, 3D-plastics and residual fraction for each particle size class respectively as assignment errors partially compensate each other.

Keywords: Particle size determination, Sensor-based measurement, Municipal Solid Waste, Particle size descriptors, PLS.

1 Introduction

The concept of a Smart Waste Factory Network 4.0 (SWFN4.0) is a research vision for mechanical waste treatment plants. According to the definition from [1], Smart Waste Factory Network means:

“The SWFN4.0 describes a system consisting of several waste treatment plants, which perform different tasks in the waste management system and are interconnected via data streams and logistics systems (e.g. sorting plants, production plants for Solid Recovered Fuels, etc.). The individual processes and machines within the plants as well as the individual plants are digitally connected with each other. This connection of the individual machines and systems and the real-time analysis of the waste streams enable dynamic process control and various actuator systems actively intervene in the processes. In addition, people can cooperate interactively with the technology around them.”.

Such Smart Waste Factory concepts and plants will support further waste management developments and the reaching of higher sorting and recycling rates for valuable waste particles available in mixed non-hazardous waste streams. The status of recycling in European municipal waste management is given in [2] and as it can be seen, technical developments are required to reach high recycling targets set up by the European Commission within the “Circular Economy Package 2018” [3].

In mechanical waste treatment plants, a shredder is usually the first machine for the treatment of solid municipal and commercial waste [4–6]. Together with the properties of the input material it affects the particle sizes of the materials and thus influences the efficiency of all subsequent machines, such as screens, magnetic separators, or sensor-based sorting machines. To beneficially influence the particle sizes in real-time – e. g. to keep them as constant as possible regardless of the variability of the input material – three things are required according to [7]: optimization algorithms (for example based on artificial neural networks), controllable actuators such as the gap width or the speed of the shaft rotation of the shredding unit, as addressed for example by [8], as

Abbreviations: NIR, near-infrared; pa, paper-cardboard; PLS(R), Partial Least Squares (Regression); re, residual fraction; RD, realistic distribution; RGB, red-green-blue; SWFN4.0, Smart Waste Factory Network 4.0; UD, uniform distribution; wo, wood.

Nomenclature

A_{Part}	Projected area of the particle	n	Number of objects (particles)
P_{Part}	Length of the perimeter of the polygon of the projected particle	p	Number of Y-variables
B	Matrix of PLS-Regression coefficients	P	Loading matrix
C	Y-weight matrix	sf_i	Shape factor of form i (rectangle, triangle, circle)
E	Matrix of X-residuals	T	Score matrix
F	Matrix of Y-residuals	X	Explanatory variable
k	Amount of calibration steps in cross-validation	Y	Response variable
l	Number of components in a PLS model		
m	Number of X-variables		

well as real-time measurements of the particle size distribution, where the latter being the focus of this work.

For an output stream characterization in waste treatment plants, material composition and particle size distribution are basic information. At present, this data is determined when the material has already left the plant, which does not allow further manipulation of the quality. [9] investigated an image analysis tool for characterising the composition of waste-derived fuels in a test set-up. Additionally, methods that use 3D laser triangulation and special techniques of image analysis to detect objects on a conveyor or from bulk [10–12] give information regarding particle measurements, however, neither of the mentioned authors used the methods on mixed (non-hazardous) waste materials. Another approach for the characterization of material streams is to determine measurements and descriptors of particles – such as projected area/circumference, Feret diameters, shape factors, bounding shapes – from two-dimensional images which is presented in this contribution. This approach was also tested on coal particles by [12].

To enable particle size analysis in real-time, machines must be equipped with the necessary sensors/cameras to record image data. It is necessary to create prediction models for particle sizes that use for example image analysis, mathematical or statistical methods to evaluate image data by software. In order to evaluate the effect on the particle size distribution the models must be combined with the information regarding individual particle weights. [13] already investigated material-specific surface weights for Solid Recovered Fuels. This approach could be used in combination with the prediction models to produce true screening lines.

To describe the size and shape of irregularly formed objects different options are possible. On one hand, projected area and circumference are common measurements, whereas the indication of meaningful lengths or diameters due to the irregularity of the outlines often turns out to be problematic. For example, in the field of process engineering equivalent diameters are often used (e.g. diameter of a circle of the same area, Sauter diameter, Martin diameter [14]). Additionally, it is possible to describe a two-dimensional object through geometrical shapes (e.g. circle, triangle, rectangle), which enclose the particle with the smallest possible area. A special type of diameter, which is often used for screening, is the Feret diameter, which is defined by the distance between

two parallel tangents that fully enclose the particle (principle of a calliper) [14]. Also, dimensionless coefficients like roundness or ratios between the projected area of the object and the areas of bounding shapes are possible options to describe particle shapes and sizes [12, 15–20]. Another approach to describe shapes is by applying Elliptical Fourier Transformation. This method has been in use for decades and uses Fourier descriptors to describe ellipses to approximate the closed contour of a shape based on a chain-code [21].

This paper presents results which were gained by using the regression method partial least squares (PLS). Here, this method was chosen because multiple, correlated variables are present to predict the corresponding particle size class of each particle. PLS extracts orthogonal factors that further allow building a regression model and identifies latent variables that are linear combinations from the original variables. These latent variables fulfil the criterion of maximal covariance between the explanatory X and response Y variables and allow modelling Y as a function of X . If Y consists of a single vector of data, the method is known as PLS1 if it represents a table of data with two or more variables PLS2 can be applied as well, but might not lead to better results in all cases [22]. PLS enables one to find a small number of factors to fit the data. Here, the choice of the considered number of components affects the ability of prediction of the model and must be chosen individually for each investigation. [23] A high number of components will usually benefit the prediction of the data, but increases the risk of overfitting (fitting the model to the noise in the data). Therefore, the ratio of components to data points must be well considered. To examine for overfitting, k -fold cross-validation [22] is a proposed method for choosing the correct number of components, which doesn't use the same data for the calibration of the model and the error prediction. Here, the data set is randomly split into k calibration steps ($5 \leq k \leq 15$), where each subset of data is tested on the model that was built from the remaining sets. k is often chosen to be five or ten, which is based on experience [24].

The paper at hand presents methods to generate image material of individual objects with known particle size- and material class, and software that was developed in MATLAB 2019 to process and analyse the image data. The software calculates particle descriptors from binary images, which were transformed from RGB images, to predict the particle size class through a regression model. The outcomes of the regression model are combined with a theoretical approach applied to a realistic waste composition to show the accuracy of the model.

2 Materials and methods

Here, the methods to generate binary image data from particles with assigned information regarding material and particle-size are described. Additionally, the used ways to calculate particle descriptors, as well as the applied regression models, are presented. A visualization of the individual steps is shown in Fig 1.

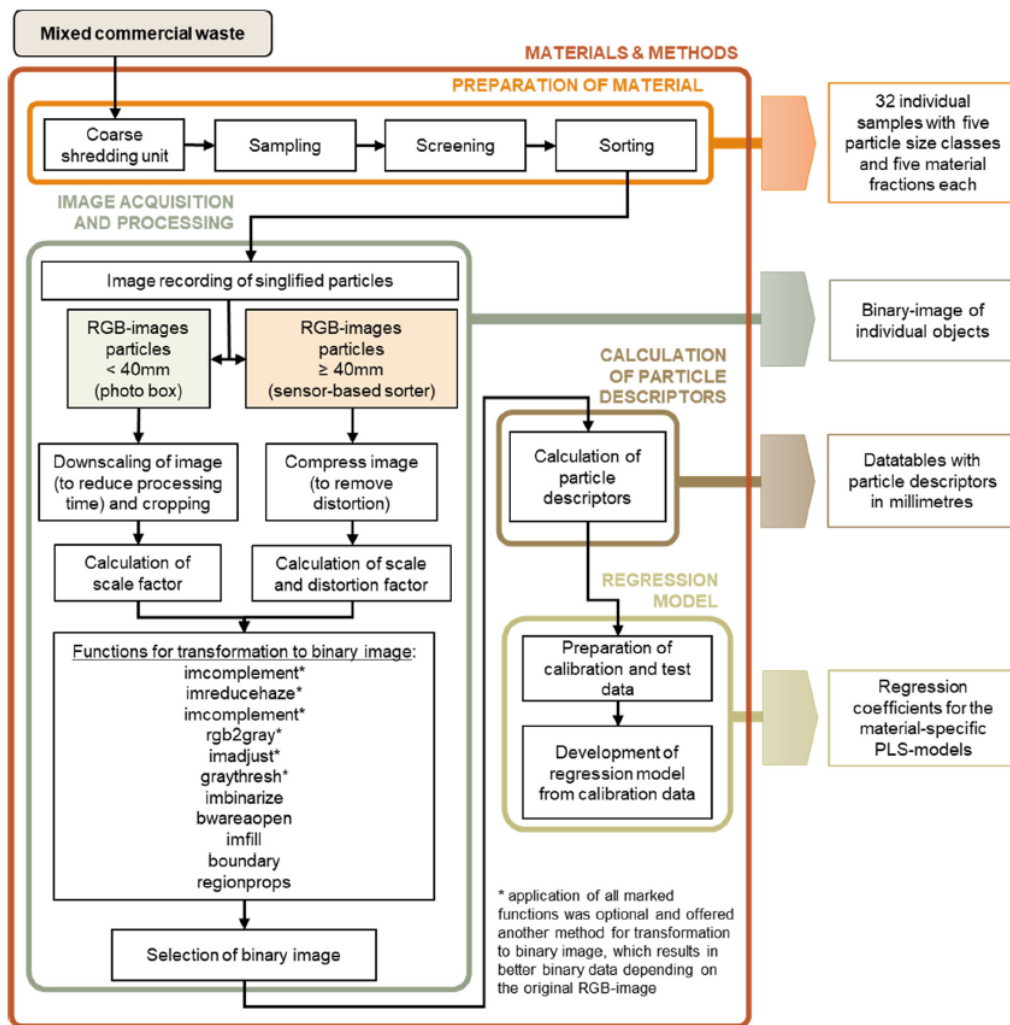


Figure 1: Schematic layout of the individual applied steps with an overview of the used MATLAB functions for the image-processing step.

2.1 Preparation of material

The work subsequently presents results that are obtained from images of individual particles from coarsely shredded solid mixed commercial waste which was collected in Styria (AUT) in October 2019. The processing of the material was performed with industrially sized machinery from the company Komptech and included a shredding step with following particle size classification through a screening process and additional manual sorting into six material classes. In total, 32 individual, representative samples were used to generate fractions with defined particle size and material class. The shredding process was performed through single-shaft shredding units (Komptech Terminator 5000 SD with cutting units F, V and XXF, described in detail by [8]) with different settings regarding shaft rotation speed and width of the radial cutting gap. This ensured the influence of different settings to be considered in the results. The screening was carried out with a drum screen (Komptech Nemus 3000), where cylindrical drums with 2m in diameter and 5.5m in length were used. All used drums had round holes with the diameters (in mm) 80, 60, 40, 20 and 10. To approximate ideal screening, a maximum volume of 1m³ of material was used per batch. This amount was spread out on the conveyor belt of the feeding bunker of the screen over a length of 4m. The speed of the conveyor was set to 0.026m/s, while the rotation speed of the drum was set to 10 rpm. The sorting process involved the assignment for each particle to one of the following material classes: lightweight fraction (e. g. foils, 2D-plastics), plastics (3D-plastics), wood, metal, paper-cardboard and residual fraction.

To reduce the effort for manual sorting due to the large number of samples, machine-aided separation was used to gain the lightweight fraction. Here, a perforated tube with air nozzles was attached in the expansion cover of a sensor-based sorting machine. The tube was of the length of the working width of the sorting machine and fed via compressed air. The lightweight fraction was separated from the rest of the material stream based on the principle of an air classifier. Due to this, the lightweight fraction contained mainly two-dimensional material like plastic foils, but also light objects of styrofoam and foamed plastic. Manual post-sorting of the material was carried out after the separation to make sure each object was assigned to the correct fraction. Hence, a manual evaluation was still the final criterion for sorting decisions.

2.2 Image acquisition and processing

The developed software requires images of single objects. In this work, this criterion was met by photographing each object individually. Due to technical and time limitations, two different approaches (image acquisition methods) for the recording of the images were applied. Particles assigned to particle sizes ≥ 40 mm were recorded by an RGB-camera in *.png format from the sensor-based sorting machine Redwave 2i. Depending on the particle size, the images were roughly 200x250 pixels in size with a resolution of 2.403mm²/pixel. Here, the camera was mounted over the accelerating conveyor and able to photograph each particle on the conveyor individually, as long as near-infrared (NIR)-spectra were detected from the particle (as NIR information is used

by the machine for defining objects). Also, it must be noted that an individual particle was defined as a cluster of material-pixels, surrounded by a defined background, which was the conveyor belt. Therefore, overlapping or touching particles were detected as one object in an image and sorted out manually in the image-processing step. Images from objects smaller than 40mm were taken with an ordinary digital camera. This size threshold was chosen because the small objects would have been difficult to handle in the experimental setup of the sensor-based sorting machine, because of the large-scale equipment with several transfer points on conveyor belts where the objects might have been lost during the process. The images were saved in *.jpg format, where each particle was photographed manually. Here, the resolution of the images for the software was originally set to 6,000x4,000 pixels, but later downsized and cropped to size in the software to roughly 400x500 pixels for faster processing. Despite the downsizing of the images, a resolution of 0.0511mm²/pixel was reached. In both cases of image acquisition, the cameras were mounted parallel to the surface where the objects were placed, and the position of the cameras was fixed to ensure a consistent distance between particle and camera lens. For manual image recording, this can be achieved using a tripod or similar. Additionally, LED-lights were used to ensure the high visibility of the particles and a well-lit area for the recording of the images. High contrast between the background and the particle itself helps to obtain images with sufficient quality for the further processing steps. In this case, all images were taken using a black background, which leads to the discarding of dark objects in the investigation, which were also not recorded by the sorter due to missing NIR-signal of black particles [25]. Due to the use of different methods for image acquisition, several steps were considered to ensure comparability between all images. Firstly, two (one for each image acquisition method) individual scaling factors – which allow the transformation between the pixel and metric measurements – were calculated as well as a distortion factor to remove the elongation which was detected on the images taken from the RGB-camera in the sorting machine due to the used combination of conveyor belt speed and imaging frequency of the line-camera. The scaling and distortion factors were calculated from images of objects with known measurements and stored in variables in the software for later access.

The software evaluates calculated measurements of objects in binary images, which were converted from the RGB-photos. In MATLAB this was achieved through transforming the original image (with the distortion factor considered, if required by the method of image acquisition) over a greyscale image (function “rgb2gray”) to a binary data set (function “imbinarize”), where background information was indicated with zero (black) and pixels of the particle with one (white). The individual steps are shown in Fig. 2. Additionally, Table 1 gives an overview of all the used functions with a short description as well as additional information regarding in- and output data of each function. To receive the best results, several steps for image improvement were applied. These involve transforming the image to a negative image or enhancing the colours, where all entered parameters must be matched with the colour of the chosen background and/or particle. In MATLAB the functions “imcomplement”, “imreducehaze” and “imadjust” were used for these steps. Latter considered a “gamma factor” of 0.2, for the transformation to a binary image a global



Figure 2: RGB image (left), greyscale image (middle), and binary image (right) of singlified waste particle.

threshold according to Otsus Method [26] was calculated (function “graythresh”), to achieve the best results.

Due to the image processing, small holes were detected in the binary data of the objects. In this context, holes are identified as regions indicated as background within the particle. To ignore falsely identified holes, all pixels of holes with a size smaller than 1% of the total image size were relabelled in the binary image from background to particle. Holes bigger than the chosen threshold were ignored in this step. Mechanical stressing from the sorting process of the material led to the separation of dust and fine particles, which were detected in the images. To ignore these objects in the evaluation only the biggest region of connected pixels was detected as the particle and displayed in the final binary image. The mentioned steps for identification and relabelling of holes as well as identifying the biggest region were achieved with the functions “bwareaopen”, “imfill” and “regionprops”.

To assure that the software only considers images where the binary data represents an object correctly (no overlapping, no cropped images due to missing contrast or NIR data) a manual check was necessary, where the best of three options was chosen. The first option considered the steps for image improvement as well as the gamma factor, which is stated in the text above. In the second option, the methods for colour improvement and the gamma factor are not applied. If the binary image was wrongly transformed, the third option allowed to not consider the image in the following analysis. Research shows that the extraction of individual particles (no touching/overlapping) from one image is possible [15]. MATLAB allows this with the function “regionprops”.

Table 1

Overview of the used functions in the software with given source references. Note: if no information regarding In- and Output data is given in the Table, the function was automatically called from another function in the process.

Name of function	Description	Input	Output	Source reference
antipodalPairs	called by the function "feretProperties"			Eddins (2017)
boundary	boundary of a set of points	coordinates of the perimeter	point indices representing a single conforming 2-D boundary around the given points	MathWorks (2020a)
bwareaopen	remove small objects from binary image	binary image	binary image without small objects	MathWorks (2020b)
bwboundaries	traces the exterior boundaries of objects, as well as boundaries of holes inside these objects, in the binary image	binary image	pixel locations for boundaries	MathWorks (2020c)
feretDiameter	called by the function "feretProperties"			Eddins (2018a)
feretProperties	calculation of Feret diameters	binary image of (perimeter of) object	tables with edge lengths, endpoint coordinates, position (angles) of min. and max. Feret diameter	Eddins (2018a)
graythresh	global image threshold using Otsu's method	greyscale image	global threshold	MathWorks (2020d)
imadjust	adjust image intensity values or colormap	greyscale image	greyscale image with adjusted colour values	MathWorks (2020e)
imbinarize	binarize 2D greyscale image	greyscale image	binary image	MathWorks (2020f)
imcomplement	complement image	image data	complement image	MathWorks (2020g)
imfill	fill image regions and holes	binary image	filled binary image	MathWorks (2020h)
imreducehaze	reduce atmospheric haze	colour or greyscale image	dehazed image	MathWorks (2020i)
incircle	compute the maximal inner circle of the polygonal convex hull of a set of points in the plane	coordinates of the perimeter	radius and centre coordinates of the circle	D'Errico (2014)
inpoly	called by the function "max_inscribed_circle"			Birdal (2011)
max_inscribed_circle	compute the centre coordinates and radius of the maximum inscribed circle of a given object	Binary image of the particle outline	radius and centre coordinates of the circle	Birdal (2011)
maxFeretDiameter	called by the function "feretProperties"			Eddins (2018a)
minAreaBoundingBox	called by the function "feretProperties"			Eddins (2018a)
minboundcircle	compute the minimum radius of the enclosing circle of a set of points in the plane	coordinates of the perimeter	radius and centre coordinates of the circle	D'Errico (2014)
minboundrect	compute the minimal bounding rectangle of points in the plane	coordinates of the perimeter	coordinates that define the rectangle, area and perimeter of the rectangle	D'Errico (2014)
minboundtri	compute the minimum area bounding triangle of points in the plane	coordinates of the perimeter	coordinates that define the triangle	D'Errico (2014)
minFeretDiameter	called by the function "feretProperties"			Eddins (2018a)
pixelHull	called by the function "feretProperties"			Eddins (2018c)
plsregress	plsregress(X,Y,ncomp) computes a partial least-squares (PLS) regression of Y on X, using ncomp PLS components	predictor and response variable, amount of PLS components	predictor and response loadings and scores, PLS regression coefficients	MathWorks (2020j)
regionprops	measure properties of image regions	binary image	measurements for the set of defined properties for each 8-connected component in the binary image	MathWorks (2020k)
rgb2gray	convert RGB image or colormap to	truecolour RGB image or colormap	grayscale image	MathWorks (2020l)
signedTriangleArea	called by the function "feretProperties"			Eddins (2018a)
triangleHeight	called by the function "feretProperties"			Eddins (2018a)
zscore	returns the z-score for each element such that columns are centred to have mean 0 and scaled to have standard deviation 1	data with non-standardized values	data with standardized values	MathWorks (2020m)

2.3 Calculation of particle descriptors

With the correctly displayed binary images available the next step was to calculate individual particle descriptors based on the binary data. To calculate the area of the particles, simply the number

of pixels of the projected area was counted. The perimeter of the object was calculated with the option “bwboundaries”. Due to the high image quality, the edge of the particles was displayed comparatively unevenly, and the length of the perimeter would have been wrongly calculated for the use here. To even out the irregular edge of the objects the perimeter was formed by a surrounding polygon with a shrink factor of 0.5 (see Fig. 3, left), which was achieved by applying the MATLAB function “boundary”. This resulted in the overall best approximation of the particle outline. It must be pointed out as well, that especially for objects with a low number of pixels (small objects and/or low-resolution images), the squared shape of the pixels (resulting in a stair-like boundary) may affect the calculation of parameters. Since in this work, all relevant objects were far over a few hundred pixels in size (A_{part}), this was not considered here. The Feret diameters were calculated with software-codes from [27–29] (function “feretProperties”). This work considers the maximal as well as the minimal Feret diameter (see Fig. 3 (left)) because they are also successfully used in other works for particle description [17, 18, 30].

The aim of using bounding shapes was to describe irregularly shaped particles more simply. In the work the following forms were determined: Smallest circumscribing rectangle (bounding box) of the particle, smallest circumscribing circle of the particle (bounding circle), largest circle in the circumscribing polygon (inscribed circle of the polygon) as well as the maximum inscribing circle of the particle (incircle) and the smallest circumscribing triangle of the particle. Of all the mentioned shapes, the dimensions that are needed to describe the shape (edge lengths, radius) were calculated and used to determine the circumference and area of the shape. Later these values were used to compare the measurements of the particle with the assigned bounding shapes. Fig. 3 (right) shows an example of some of the mentioned bounding shapes. In the software, functions from [31] (“minboundrect”, “minboundcircle”, “incircle”, “minboundtri”) and [32] (“max_inscribed_circle”) were used for the calculation.

Shape factors sf_i are dimensionless factors and show the ratio between different particle descriptive areas. Here, the following sf are considered: bounding box, bounding circle, bounding triangle, inner circle polygon and incircle. Additionally, the circularity was considered as a shape factor as well, which explains the difference of the particle from a circle. This factor was defined through Eq.(5.38) according to [16], where A_{part} is the projected area of the particle and P_{part} is the perimeter of the circumscribing polygon, and was defined in a way, so that it becomes 1 for a circle.

$$Circularity = \frac{4\pi A_{part}}{P_{part}^2} \quad (5.38)$$

The descriptors with additional information regarding particle size and material were calculated for each image and stored in table form, resulting in a data table containing the information regarding each particle in one row and the values for the individual descriptors in columns. All measurements were calculated in the unit pixel and were transformed in millimetre measurements. Here, the previously evaluated scale factor for the respective particle size class was considered in addition to the dimension of the unit of the descriptor (two dimensions for e. g. areas, one dimension for

e.g. lengths, diameters and zero dimensions for shape factors). In the whole software the material fractions are examined separately, which leads to five individual data tables. These were used as input for the regression models, where one for each material class was considered.

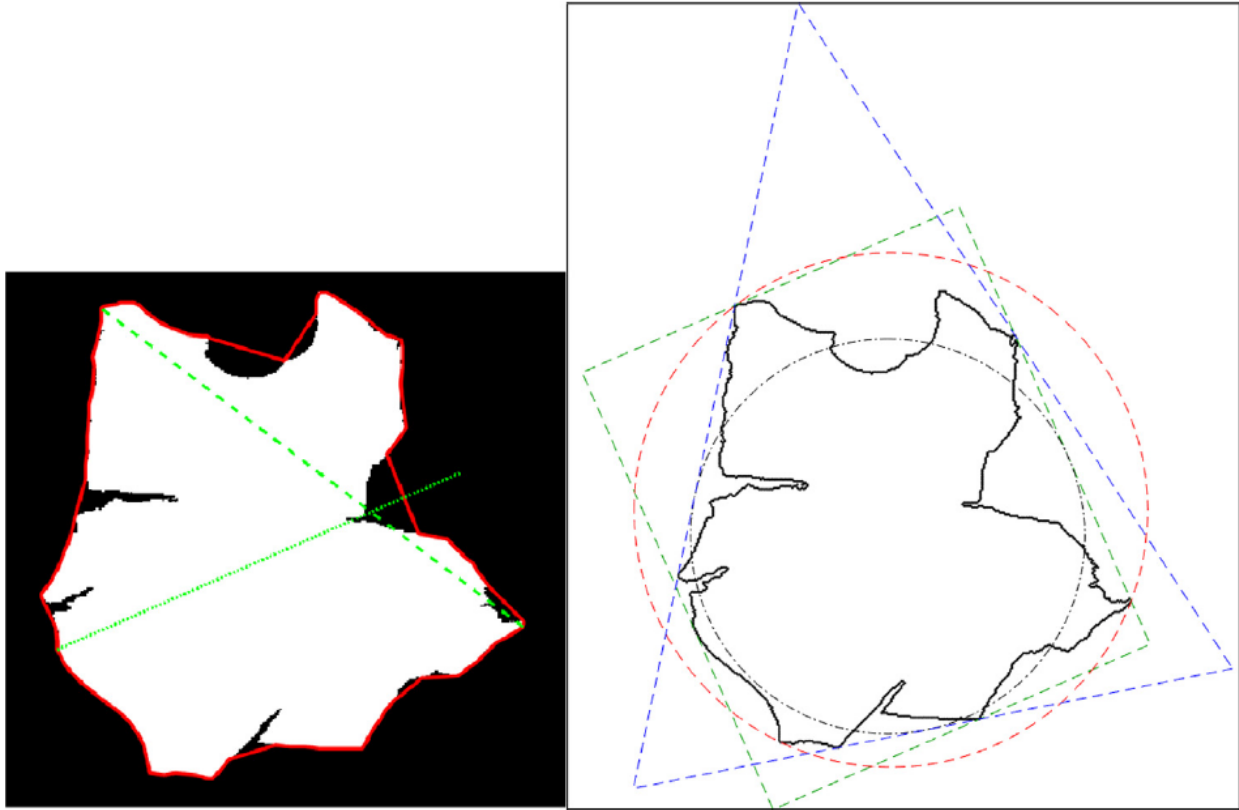


Figure 3: Left: Particle with polygon perimeter (solid line), max. Feret diameter (dashed line) and min. Feret diameter (dotted line); right: Particle outline with exemplary bounding shapes (bounding box, bounding triangle, bounding circle (dashed line), inscribed circle of the circumscribing polygon (dash-dotted line))

2.4 Regression model

Prior investigations from this work show that the prediction performance of the particle sizes based on only one parameter performs poorly, because of the broad scattering of the values and collinearity between the variables [33]. Therefore, the regression was performed with PLS1, where all the information of particle descriptors was used as predictor variables, and the information regarding particle size class was used as the one-dimensional response variable. The PLS method was chosen because it offers solutions for highly collinear or linearly dependent predictors [34], which were determined in this case. The basic concept of PLS is stated with the following explanations and formulae from [23, 34] and [22, 35].

Table 2

Overview of the considered variables (descriptors) and their explanation.

Descriptor	Definition
Area particle (A_{part})	projected area of the object
Perimeter (P_{part})	length of the perimeter of the polygon of the projected area
Area bounding box	area of the smallest rectangle, which surrounds the object
Area bounding triangle	area of the smallest triangle, which surrounds the object
Shape factor ^a	ratio between the area of the object and area of the respective (bounding) shape
Circularity	explains difference from a circle and is defined through Eq. (1)
Diameter incircle	diameter of the largest inscribed circle of the particle
Feret diameter (maximal, minimal)	maximal/minimal distance between two parallel tangents, which fully enclose the object (principle of a calliper)

^a Bounding box, bounding circle, inscribed circle of polygon, incircle, bounding triangle.

The idea of PLS is to find a regression model for multivariate data for predicting purposes of new data by a small number of relevant factors. Here, the matrix of explanatory variables X is decomposed in a set of orthogonal factors which are used for fitting the matrix of response variables Y . In PLS latent variables (cannot be measured directly) are built that account for as much of the manifest variable (can be measured directly) variation as possible and are represented as linear combinations of the manifest variables. The final regression model is in the linear form of $Y=XB$, with B being regression coefficients which are calculated over the PLS factors between the explanatory and the response variables. The general model that underlies the PLS is given in Matrix form in equation (5.39) and (5.40) from [35] as

$$X = TP^T + E \quad (5.39)$$

$$Y = TC^T + F \quad (5.40)$$

where X is an $n \times m$ matrix of predictors, Y is an $n/timesp$ matrix of responses. T is an $n \times 1$ matrix containing projections of X (X -score, component or factor matrix). P is an $m \times 1$

orthogonal loading matrix, C is an $m \times 1$ matrix and contains the regression coefficients which relate T to the variables in Y . The residuals are collected in the matrices E and F which are the error terms. PLS allows a variable reduction where the available variables are combined through linear combinations to fewer ones with more significance. Additionally, the method allows finding certain variables, that are essential for the description of the data and identify the ones that don't provide additional information. This can be done by interpreting the values of the weights w , which can be identified through the scores T .

To evaluate the models properly the available data tables were split into two separate groups, one containing 90% of the data from each particle size class, which was randomly picked. This data is later called the calibration data. The remaining data is used to test the quality of the developed model and is therefore called test data. In MATLAB the function "plsregress" was applied, which is based on the algorithm from [34]. Additionally, the MATLAB function "zscore" was used to standardize the variables in a way that each column in the data set had a mean of 0 and a standard deviation of 1. A list of the used predictor variables and their explanation is given in Table 2. Due to the chosen methods of material preparation and image acquisition an unequal amount of images in each material-particle size class was available. To consider this fact in the regression model and avoid the more precise modelling of particle size classes with a higher number of images a way to compensate this factor was necessary. This was done by adding a weighting factor, which is defined as the inverse value of the number of available images in each material-particle size class, respectively. Since the PLS is a method that allows the reduction of dimensions and eliminates noise in the data, the number of considered dimensions was chosen to be four. This was evaluated as the best option for the regression models, without under- and overfitting the data based on k-folds cross-validation with k being chosen to be 10, where at least 94% of the explained variance in Y was considered in the models.

The result of the models were regression coefficients for the predictor variables based on the calibration data, which were later applied to the test data sets to predict the particle size. Based on the fact that in this work five different particle size classes are present, the classification for the values to the particle size classes was implemented by considering a limit value for each particle size class. This limit value was calculated as the mean value between the predicted values of the 80th percentile of the lower particle size class and the 20th percentile of the higher particle size class (e. g. the limit value to divide particle size class 40–60mm from 60–80mm was the mean from the 80th percentile of the predicted values of the 40–60mm particles and 20th percentile of the predicted values of the 60–80mm particles).

3 Results and discussion

In this investigation in total over 11,000 particles were characterized with the chosen descriptors listed in Table 2. The breakdown according to particle size class and material fraction are shown in Table 3. Here, the values show the number of particles that were used for the regression models

due to the manual selection of the binary images. The values in brackets show the share from the total amount of images that were taken from the cameras that is represented through the number of characterized particles. It can be detected that less than half of the amount of the available pictures in total were used in the regression. The following different factors explain the issue for particles $\geq 40\text{mm}$. Firstly, especially large objects were cropped and not fully displayed on one image, but on two or even more. This was also detected on particles with a very small expansion in one dimension, which often led to the detection of two or more individual objects. Additionally, images were often cropped when objects consisted of several different materials (composite materials) and the material was partly not reflecting NIR-signals, and therefore not detected by the sensor. It is important to mention that latter was only relevant for particles $\geq 40\text{mm}$ since only they were recorded by the sensor-based sorting machine. Further, the surface condition was an important factor for image quality. If the particle was too dark in colour, see-through, too shiny, or had a reflective surface the image was often not properly transformed into a binary image and was not suitable for the regression model. These factors explain the significantly lower number of particles in the residual fraction when compared to the other materials, where e.g. glass, metal-coated plastics, composite materials and dark textile fibres and fabrics accounted for a high number of particles. Due to the composition and structure of the lightweight and the residual fraction (fluffy material that was tangled up with other particles) no photos were taken for these two material fractions in the particle size 10–20mm. Also, it should be noted that the metal fraction was not considered in the evaluation. Here, a low number of objects was initially found in the samples. A significant amount of these metallic objects were wires, which were often not correctly displayed on the images due to the one-dimensional appearance, flat metallic objects were often too shiny. Therefore, only an exceedingly small number of usable image files (less than 40 images in all particle size classes combined) was available and forced the fraction to be taken out from the investigation.

The given regression coefficients from the regression models were applied to the test data sets for each material fraction. The results were values that were assigned to a particle size class based on the classification method explained in the previous chapter. Because the size of the test data sets was dependent on the number of available images some material-particle size specific fractions held less than ten images. To evaluate the regression models in a way that even a small number of available objects gives meaningful information regarding the accuracy of the models the results presented here are calculated as a mean of 1,000 individual runs. In more detail, this means that 1,000 calibration data sets with data from randomly selected particles were used to generate the regression models. Each of the models was then tested by applying the resulting regression coefficients to the respective test data set and calculate the falsely and correctly assigned particles for each size class. These values can be seen as instant results from the regression models and are used for the prediction of the particle size. The mean results from 1,000 runs for the individual materials are presented in Table 4 as a significant result to show the accuracy of the method. Overall, the models for the materials wood and residual fraction show the best results regarding

Table 3

Number and share of characterised particles regarding particle size class and material fraction. The numbers in brackets give the share of successfully analysed particles for each material-particle-size fraction.

	>80 mm		60–80 mm		40–60 mm		20–40 mm		10–20 mm		Total	
Plastics	1,422	(0.28)	748	(0.46)	998	(0.45)	409	(0.61)	241	(0.56)	3,818	(0.38)
Wood	495	(0.58)	452	(0.66)	409	(0.59)	350	(0.62)	225	(0.49)	1,931	(0.59)
Lightweight fraction	204	(0.19)	231	(0.36)	1,113	(0.45)	291	(0.57)	–	–	1,839	(0.39)
Paper-cardboard	2,061	(0.44)	266	(0.60)	232	(0.66)	363	(0.64)	119	(0.52)	3,041	(0.49)
Residual	61	(0.19)	70	(0.30)	133	(0.24)	237	(0.40)	–	–	501	(0.30)
Total	4,243	(0.35)	1,767	(0.49)	2,885	(0.46)	1,650	(0.57)	585	(0.52)	11,130	(0.42)

the correct classification of the particles. However, the fraction paper-cardboard shows a very low correct classification of particles between the sizes 20–60mm. To better evaluate the effects of the wrongly classified objects, the results from Table 4 are applied on particle size distributions. As no information regarding the individual particle weights was examined during this investigation, for this first approach it was assumed that each object in the same material-particle size class has the same weight. Two different options were investigated here.

Firstly, a uniformly distributed (UD) composition of the individual particle size classes within the material fractions was considered. Since five different particle classes were examined, each class was considered with a 20% mass fraction. For the material groups lightweight and residual fraction no images in the particle size class 10–20mm were available, which is why these could not be included in the regression. In these cases, the particle size classes were each taken into account with 25% of the mass fraction.

Additionally, a realistic distribution (RD) of particle size classes within individual material fractions in mixed solid commercial waste according to [36] was considered. This distribution was chosen because it deals with the same input material, similar individual material fractions and similar particle size classes. The individual fractions are split up in more detailed material classes as in the present paper, but their combination allows the representation of the investigated fractions in this contribution. For this, the individual fractions “paper” and “cardboard” were summed up for the representation of the paper fraction. For the residual fraction the values from the “inert material” and “textiles” were added to the “residual fraction”. The used values for the further calculations are presented in Chapter 4. Since in that study no data according to material-specific composition in fractions smaller 20mm was investigated, the particle size class 10–20mm was not considered in the results presented here.

The aforementioned assumption regarding particle masses combined with the two options of particle size distributions allows to constitute the impacts of the regression models and compare the original particle size distribution with the one predicted through the respective model. The extent of incorrectly assigned particles on the particle size distribution is shown, considering the partially compensating effect of wrong classifications. The results of the according hypothetical sensor-based screening analysis in form of the percentual error of the mass fractions between the predicted values and the original particle size distribution is shown in Table 5. Furthermore, the errors of the models can be given by calculating the mean error in terms of mass share for all

Table 4

Mean accuracy of the PLS-classification for 1,000 individual randomly picked test data sets in shares for the individual material fractions (wo: wood, 3D: 3D plastics, pa: paper&cardboard, 2D: lightweight fraction, re: residual fraction). Bold numbers represent the share for correctly classified objects.

	to 10–20 mm [%]	to 20–40 mm [%]	to 40–60 mm [%]	to 60–80 mm [%]	to >80 mm [%]	absolute number of particles [–]
wo from 10–20 mm	100	0	0	0	0	22
wo from 20–40 mm	0	99	1	0	0	35
wo from 40–60 mm	0	1	88	11	0	41
wo from 60–80 mm	0	0	12	71	17	45
wo from >80 mm	0	0	2	16	82	49
3d from 10–20 mm	100	0	0	0	0	24
3d from 20–40 mm	0	100	0	0	0	41
3d from 40–60 mm	0	0	57	39	4	100
3d from 60–80 mm	0	0	38	51	11	75
3d from >80 mm	0	0	1	12	87	142
pa from 10–20 mm	100	0	0	0	0	12
pa from 20–40 mm	0	18	38	44	0	36
pa from 40–60 mm	0	72	15	13	0	23
pa from 60–80 mm	0	4	8	88	0	27
pa from >80 mm	0	0	0	0	100	206
2d from 20–40 mm	0	72	27	1	0	29
2d from 40–60 mm	0	29	65	6	0	111
2d from 60–80 mm	0	0	7	65	28	23
2d from >80 mm	0	0	2	25	73	20
re from 20–40 mm	0	100	0	0	0	24
re from 40–60 mm	0	0	100	0	0	13
re from 60–80 mm	0	0	0	96	4	7
re from >80 mm	0	0	0	6	94	6

Table 5

Material-specific error (in %) between the mass share of the predicted regression model and the mass share of the considered distribution (UD, RD) for each particle size class, as well as the mean error per particle size class (wo: wood, pa: paper&cardboard, 3D: 3D plastics, re: residual fraction, 2D: lightweight fraction, UD: uniformly distributed, RD: realistic distribution).

particle size [mm]	wo	wo	pa	pa	3D	3D	re	re	2D	2D
	UD	RD	UD	RD	UD	RD	UD	RD	UD	RD
10–20	0	–	0	–	0	–	0	–	0	–
20–40	0	0	–1	1	0	0	0	0	0	1
40–60	1	1	–8	–9	–1	1	0	0	0	1
60–80	–1	–1	9	8	0	2	0	2	0	14
>80	0	0	0	0	1	–3	0	–2	0	–16
Error per particle size class	0.2	0.25	1.8	2.25	0.2	0.75	0	0.5	0	4

particle size classes, which are also presented in Table 5. Here, it is shown that the approach of uniformly distributed (UD) material regarding particle size classes shows overall better results than considering the realistic particle size distribution (RD). Nevertheless, the accuracy of the predicted particle size distribution for the fractions wood, plastics (3D) and residuals was at least 99% for each particle size class respectively. It is also shown that the error between UD and RD for each material is in the same range, except for lightweight material. Here, the difference in error can be interpreted by the difference between the particle size shares in the distributions, where RD considers approx. 70% of particles ≥ 80 mm, while UD just 25%.

4 Conclusions

The presented methodology, which predicts the particle size based on parameters from 2D images, shows promising results for measuring the particle size distribution of the investigated material fractions. Considering a realistic particle size distribution with the approach of identical particle weights in each particle size class the regression model led to a correct prediction of at least 99% the individual particle size classes for wood, plastics (3D) and residual fraction. The fractions paper-cardboard and lightweight materials showed significant errors but were still over 96% accurate for each particle size class respectively. Nevertheless, the following points should be noted as a limitation of the stated method.

For the recording of images by the RGB sensor of the sensor-based sorting machine a detected NIR-signal was crucial. This factor mainly caused dark (especially black and grey) objects not to be considered in the evaluation. Additionally, certainly shaped objects (one-dimensional) were recorded on multiple separate images and therefore not useable in the investigation. The chosen way of image processing in the software requires a manual check of the binary images. This is not feasible for a practical application and could not be implemented due to the real-time request in a plant. Hence, a software extension is necessary, by which this control can be carried out automatically to eliminate the manual effort.

Another point of criticism is that large objects were often cropped on the images due to the limiting size of the images. This led to missing particle information in bigger particle size classes. The fine fraction below 10mm was not considered in the investigation at all but is crucial information when evaluating particle size distributions.

The lightweight fraction was classified by the air classifier and consisted mostly of plastic foils. Additionally, the whole material stream was checked by hand as well to avoid wrongly classified objects (e. g. foils wrapped up with wires). In this paper, an approach to evaluate the material-specific particle size class from the whole (mixed) material stream was investigated. Especially the differentiation between the materials lightweight fraction and 3D plastics based only on the information from a NIR-sorting machine is very limited in a real plant, since both fractions include objects from the same materials [37]. Also, an efficient singlification of the material stream on a conveyor for sufficient detection of particle shapes is not always possible due to limited space and upkeeping of the mandatory throughput of the plants. Here, a bypass for smaller quantities could be used as a technical solution for material analysis. Additionally, the image recording must be tested when the particle size classes are not separately recorded by the cameras. Here, especially large foils $\geq 80\text{mm}$ would probably cover up smaller particles on the conveyor belt, so that a screening step might be necessary. Nevertheless, although further research is needed to develop a Smart Waste Factory, the presented approach shows high potential to be used as an automated method to measure particle size distributions of solid, mixed waste streams to process large amounts of data.

Besides, particle weights must be examined and combined with the models to evaluate the effect of the wrongly classified particles for more realistic information regarding particle size dis-

tribution. I.e. waste management will become particle-, sensor- and data-based-management and information from every single particle will become increasingly relevant in the future, especially because of higher recycling rates set up by the EU Circular Economy Package.

Further, approaches regarding machine learning could be investigated to evaluate if additional information about particle shapes can be extracted based on the chosen particle descriptors, or other information in the images. The images (true colour or greyscale) can also be processed over Convolutional Neural Networks, which can detect characteristic shapes and edges of objects and may lead to better identification of shapes.

Ultimately, it must be mentioned that the results are based on material that is classified by a drum screen. If the method should be applied on other screen types or drum screens with different screen perforations, the efficiency of the respective screen on material fractions and particle size classes must be investigated separately but could give information about particle sizes when combined with the presented method here.

Funding

Partial funding of this work was provided by: The Center of Competence for Recycling and Recovery of Waste 4.0 (acronym ReWaste4.0) (contract number 860 884) under the scope of the COMET – Competence Centers for Excellent Technologies – financially supported by BMK, BMDW and the federal state of Styria, managed by the FFG.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank their students who actively supported the execution of experiments. We also express our thanks to our project partners who significantly contributed to the successful completion of the test series by providing equipment and infrastructure.

Appendix A. Supplementary material

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.wasman.2020.11.003>.

Bibliography

- [1] R. Sarc, A. Curtis, L. Kandlbauer, K. Khodier, K. Lorber, R. Pomberger, Digitalisation and intelligent robotics in value chain of circular economy oriented waste management – a review, *Waste Management* 95 (2019) 476–492. doi:<https://doi.org/10.1016/j.wasman.2019.06.035>.
- [2] R. Pomberger, R. Sarc, K. Lorber, Dynamic visualisation of municipal waste management performance in the eu using ternary diagram method, *Waste Management* 61 (2017) 558–571. doi:<https://doi.org/10.1016/j.wasman.2017.01.018>.
- [3] European Commission, Directive (EU) 2018/851 of the European Parliament and of the Council of 30 may 2018 amending Directive 2008/98/EC on waste. (2018).
- [4] R. Sarc, I. Seidler, L. Kandlbauer, K. Lorber, R. Pomberger, Design, quality and quality assurance of solid recovered fuels for the substitution of fossil feedstock in the cement industry – update 2019, *Waste Management & Research* 37 (9) (2019) 885–897. doi:[10.1177/0734242X19862600](https://doi.org/10.1177/0734242X19862600).
- [5] R. Sarc, K. Lorber, R. Pomberger, M. Rogetzer, E. Sipple, Design, quality, and quality assurance of solid recovered fuels for the substitution of fossil feedstock in the cement industry, *Waste Management & Research* 32 (7) (2014) 565–585. doi:[10.1177/0734242X14536462](https://doi.org/10.1177/0734242X14536462).
- [6] R. Sarc, K. Lorber, Production, quality and quality assurance of refuse derived fuels (rdfs), *Waste Management* 33 (9) (2013) 1825–1834. doi:<https://doi.org/10.1016/j.wasman.2013.05.004>.
- [7] K. Khodier, A. Curtis, R. Sarc, M. Lehner, P. O’Leary, R. Pomberger, Smart solid waste processing plant: vision and pathway (2019).
- [8] K. Khodier, C. Feyerer, S. Möllnitz, A. Curtis, R. Sarc, Efficient derivation of significant results from mechanical processing experiments with mixed solid waste: Coarse-shredding of commercial waste, *Waste Management* 121 (2021) 164–174.
- [9] S. Peddireddy, P. Longhurst, S. Wagland, Characterising the composition of waste-derived fuels using a novel image analysis tool, *Waste Management* 40 (2015) 9–13. doi:<https://doi.org/10.1016/j.wasman.2015.03.015>.
- [10] S. Flamme, S. Hams, M. Kölking, ARGOS-Entwicklung eines Multisensor-Systems zur Echtzeitanalyse von metallreichen Aufbereitungsprodukten [Development of a multisensor-system for real-time analysis of metal-rich processing products], *Vorträge-Konferenzband zur 14. Recy & DepoTech-Konferenz. Recy & DepoTech-Konferenz, Leoben* (2018) 119–124.

- [11] M. Kontny, Machine vision methods for estimation of size distribution of aggregate transported on conveyor belts, *Vibroengineering PROCEDIA* 13 (2017) 296–300. doi:10.21595/vp.2017.19151.
- [12] Z. Zhang, J. Yang, X. Su, L. Ding, Analysis of large particle sizes using a machine vision system, *Physicochemical problems of mineral processing* 49 (2013).
- [13] P. Krämer, *Entwicklung von Berechnungsmodellen zur Ermittlung relevanter Einflussgrößen auf die Genauigkeit von Systemen zur nahinfrarotgestützten Echtzeitanalytik von Ersatzbrennstoffen*, Shaker Verlag, 2017.
- [14] A. Yang, M. Zhu, J. Zhang, G. Li, Early cambrian eodiscoid trilobites of the yangtze platform and their stratigraphic implications, *Progress in Natural Science* 13 (11) (2003) 861–866. doi:10.1080/10020070312331344560.
- [15] G. Dunnu, T. Hilber, U. Schnell, Advanced size measurements and aerodynamic classification of solid recovered fuel particles, *Energy & Fuels* 20 (4) (2006) 1685–1690. doi:10.1021/ef0600457.
- [16] M. Hentschel, N. Page, Selection of descriptors for particle shape characterization, *Particle & Particle Systems Characterization* 20 (1) (2003) 25–38. doi:https://doi.org/10.1002/ppsc.200390002.
- [17] E. Olson, Particle shape factors and their use in image analysis part 1: theory, *J.GXP Compliance* 15 (3) (2011) 85–96.
- [18] F. Podczek, A shape factor to assess the shape of particles using image analysis, *Powder Technology* 93 (1) (1997) 47–53. doi:https://doi.org/10.1016/S0032-5910(97)03257-9.
- [19] I. Zimmermann, Teilchengrößenanalyse [Analysis of particle size], *Pharmazeutische Technologie [Pharmaceutical Technology]* 3 (2018) 245–300.
- [20] M. Zlatev, Beitrag zur quantitativen Kornformcharakterisierung unter besonderer Berücksichtigung der digitalen Bildaufnahmetechnik [Contribution to quantitative characterization of the particle shape with special consideration of digital imaging technology]Dissertation, Freiberg (2005).
- [21] F. P. Kuhl, C. R. Giardina, Elliptic fourier features of a closed contour, *Computer Graphics and Image Processing* 18 (3) (1982) 236–258. doi:https://doi.org/10.1016/0146-664X(82)90034-X.
- [22] B. G. M. Vandeginste, D. L. Massart, L. M. C. Buydens, S. de Jong, P. J. Lewi, J. Smeyers-Verbeke, Multivariate calibration, *Handbook of Chemometrics and Qualimetrics: Part B* By

- B. G. M. Vandeginste, D. L. Massart, L. M. C. Buydens, S. De Jong, P. J. Lewi, and J. Smeyers-Verbeke. (1998) 349–381.
- [23] D. L. Massart, B. G. M. Vandeginste, L. M. C. Buydens, S. de Jong, P. J. Lewi, J. Smeyers-Verbeke, Principal components, Handbook of Chemometrics and Qualimetrics: Part A By D. L. Massart, B. G. M. Vandeginste, L. M. C. Buydens, S. De Jong, P. J. Lewi, and J. Smeyers-Verbeke. (1998) 519–556.
- [24] M. Kuhn, K. Johnson, et al., Applied predictive modeling, Vol. 26, Springer, 2013.
- [25] S. P. Gundupalli, S. Hait, A. Thakur, A review on automated sorting of source-separated municipal solid waste for recycling, Waste Management 60 (2017) 56–74, special Thematic Issue: Urban Mining and Circular Economy. doi:<https://doi.org/10.1016/j.wasman.2016.09.015>.
- [26] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man, and Cybernetics 9 (1) (1979) 62–66. doi:10.1109/TSMC.1979.4310076.
- [27] S. Eddins, Feret properties – wrapping up, <https://blogs.mathworks.com/steve/2018/04/17/feret-properties-wrapping-up>, [Online; accessed April 23, 2020] (2018).
- [28] S. Eddins, Minimum feret diameter, <https://blogs.mathworks.com/steve/2018/02/20/minimum-feret-diameter>, [Online; accessed April 23, 2020] (2018).
- [29] S. Eddins, Feret diameters and antipodal vertices, <https://blogs.mathworks.com/steve/2017/10/24/feret-diameters-and-antipodal-vertices>, [Online; accessed April 23, 2020] (2017).
- [30] S. Dražić, N. Sladoje, J. Lindblad, Estimation of feret’s diameter from pixel coverage representation of a shape, Pattern Recognition Letters 80 (2016) 37–45. doi:<https://doi.org/10.1016/j.patrec.2016.04.021>.
- [31] J. D’Errico, A suite of minimal bounding objects, <https://www.mathworks.com/matlabcentral/fileexchange/34767-a-suite-of-minimal-bounding-objects>, [Online; accessed April 23, 2020] (2014).
- [32] T. Birdal, Maximum inscribed circle using distance transform, <https://www.mathworks.com/matlabcentral/fileexchange/30805-maximum-inscribed-circle-using-distance-transform>, [Online; accessed July 27, 2020] (2011).
- [33] L. Kandlbauer, Sensorische Messung der Trommelsieb-Korngrößenverteilung von gemischten Gewerbeabfällen [Sensor-based measurement of particle size distribution of mixed commercial waste] (2016).

- [34] S. de Jong, Simpls: An alternative approach to partial least squares regression, *Chemometrics and Intelligent Laboratory Systems* 18 (3) (1993) 251–263. doi:[https://doi.org/10.1016/0169-7439\(93\)85002-X](https://doi.org/10.1016/0169-7439(93)85002-X).
- [35] B. G. M. Vandeginste, D. L. Massart, L. M. C. Buydens, S. de Jong, P. J. Lewi, J. Smeyers-Verbeke, Relations between measurement tables, *Handbook of Chemometrics and Qualimetrics: Part B* By B. G. M. Vandeginste, D. L. Massart, L. M. C. Buydens, S. De Jong, P. J. Lewi, and J. Smeyers-Verbeke. (1998) 307–347.
- [36] K. Khodier, S. A. Viczek, A. Curtis, A. Aldrian, P. O’Leary, M. Lehner, R. Sarc, Sampling and analysis of coarsely shredded mixed commercial waste. part i: procedure, particle size and sorting analysis, *International Journal of Environmental Science and Technology* 17 (2) (2020) 959–972.
- [37] S. Möllnitz, K. Khodier, R. Pomberger, R. Sarc, Grain size dependent distribution of different plastic types in coarse shredded mixed commercial and municipal waste, *Waste Management* 103 (2020) 388–398. doi:<https://doi.org/10.1016/j.wasman.2019.12.037>.

A Digital Twin for Deep Vibro Ground Improvement

Negin Khalili-Motlagh-Kasmaei¹, Dimitar Ninevski¹, Paul O’Leary¹
Vincent Winter², Christopher J. Rothschedl¹, Alexander Zöhrer²

¹University of Leoben, A8700 Leoben, Austria

automation@unileoben.ac.at

²Keller Grundbau Ges.mbH, Guglgasse 15, BT4a / 3. OG,

1110 Wien

{vincent.winter, alexander.zoehrer}@keller.com

Abstract

This paper presents the implementation of a digital twin for the vibro - replacement ground improvement process, as part of a digital construction site. The goal is to implement an evidence-based quality assurance for each produced element (column or point). The system uses a combination of planning data and real-time machine data collected automatically from construction equipment. In addition to the quality assurance, feedback is provided in a number of areas, e.g., condition monitoring of the equipment. The digital twin builds upon the real-time sensor and machine data, fused with metadata, to implement a digital model for the process. A hierarchical data evaluation, using rule-based Key Performance Indicators (KPIs), permits viewing of the spatial variation of data over the construction site. This permits the modelling of systematic variations in subsurface properties across the site, enabling the detection of anomalous individual elements, the diagnosis of their cause and the prediction of properties as construction proceeds over the site. All time-series data and KPIs are represented by standard classes of objects, leading to a generic coding of the data analysis; this minimizes the effort required to add additional sites and/or new KPIs. Examples are presented for a number of sites, where the system was used to identify anomalous columns.

Keywords: Deep vibro, Ground improvement, Digital twin, Time series analysis, Key performance indicators

1 Introduction

This paper addresses the issues involved in the implementation of a digital twin of the vibro-replacement ground improvement process. The digitalization of the construction sector is becoming an increasingly popular topic. Recently, COVID-19 has motivated various contributions, which have sped up this process significantly. Additionally, there is an even more important change ongoing in the field of construction, namely the digitalization of the construction site itself. Focusing now on the geotechnical sector, design data can be transmitted to sites in a digital format [1]. This can be seen in the context of the Building Information Modelling (BIM), which has been a crucial aid to improve construction quality during its whole life cycle [2], [3]. The approach here is to extend the BIM concept with real-time machine data from the construction site [4]. In this manner, the digital twin can combine both site design data with real-time operational data, collected from a wide range of sensors mounted on various equipment across the site. This permits a validation of the execution of the planned elements (columns or points). This relies on the quality of data acquisition, which has improved remarkably during the last years. The time-series data can now be used to assist the rig operator during production, helping to detect anomalous behavior [5]. The large volume of data, now collected from a construction site, necessitate the development of tools to evaluate the quality and performance of the process [6], [7]

The execution of ground improvement techniques is especially dependent on the availability of process data. Simple quality control methods, such as visual inspections or sample-taking are not suitable to check the correct installation of foundation elements. Simple geometrical parameters, such as length or inclination of columns, can be checked easily, using the data acquired by the rig. Installation reports show the basic installation parameters, which are generated for every element. However, a typical characteristic of soil improvement techniques is the execution procedure and therefore the properties of the elements depend on the soil conditions. Unfortunately, soil exploration campaigns are often not sufficiently precise, so the installation of every element is supposed to deliver additional information on layering and mechanical properties of the soil. A new approach, enabled by the digital twin, is to use rule-based algorithms to compute Key Performance Indicators (KPI) and to view their spatial distributions over a complete site. The KPI can be used to detect, diagnose [7], [8] and predict subsurface properties.

2 Current State of Ground Improvement Monitoring

The Vibro-Replacement is a common ground improvement process; hereby, the ground is penetrated by a deep-vibrator, then during retraction of the vibrator, gravel is pressed into the open space below the tip of the vibrator and vibrated to obtain a higher gravel stiffness. For the purposes of monitoring, the process can be segmented into two phases (a portion of such real-time data set is shown in Fig. 2):

Penetration: In this phase, the vibrator penetrates the ground. During penetration, the in-situ

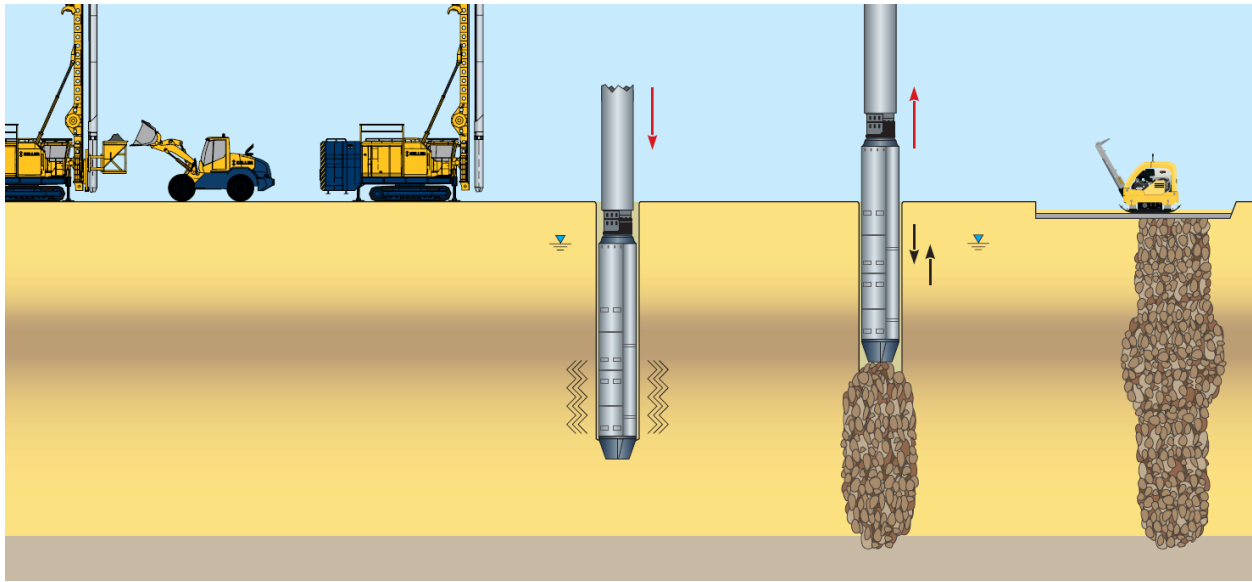


Figure 1: Vibro-Replacement production process as an example for ground improvement.

soil is displaced with the help of flushing with compressed air. The vibratory motion is produced by an eccentric weight powered by an electric motor, where both the frequency and power of the vibrator are monitored. The vertical penetration is driven by the pull-down force / activation force. The vibratory power and pull-down force result in a penetration rate that is related to the stiffness of the ground. Consequently, these parameters are also logged in real-time. Additionally, the temperature of the vibrator is monitored, since the work performed to compact the ground results in heating. It must be ensured that the vibrator does not exceed a specific temperature limit. The penetration process continues to a defined design depth, which in most cases is a bearable soil layer. However, there are additional abort criteria, based on vibration power and activation force. These are triggered when a sufficiently strong bearing level is reached. A series of KPIs is computed specifically to characterize the penetration phase.

Compaction: The start of the compaction phase is taken as the moment the final depth has been reached. The vibrator string is filled with gravel, so a vertical reciprocating process is started. This reciprocating motion consists of a cyclic process of retraction and compaction. Each reciprocation introduces gravel into the hole and compacts it to the required stiffness; this is determined from the vibrator power and pull-down force. Thus, the actual diameter of the column as a function of depth depends on the local density and the load-carrying capacity of the soil, see Fig. 1. The reciprocation steps continue until the column is completed, with pauses needed to reload gravel. Consequently, the monitoring permits the computation of built-in volume of gravel per meter as a function of depth. This is one of a number of quality-relevant KPIs computed for the compaction phase.

The result of the penetration and compaction phase is an improvement of the ground load-carrying capacity. The monitoring permits an evidence-based verification of the correct execution

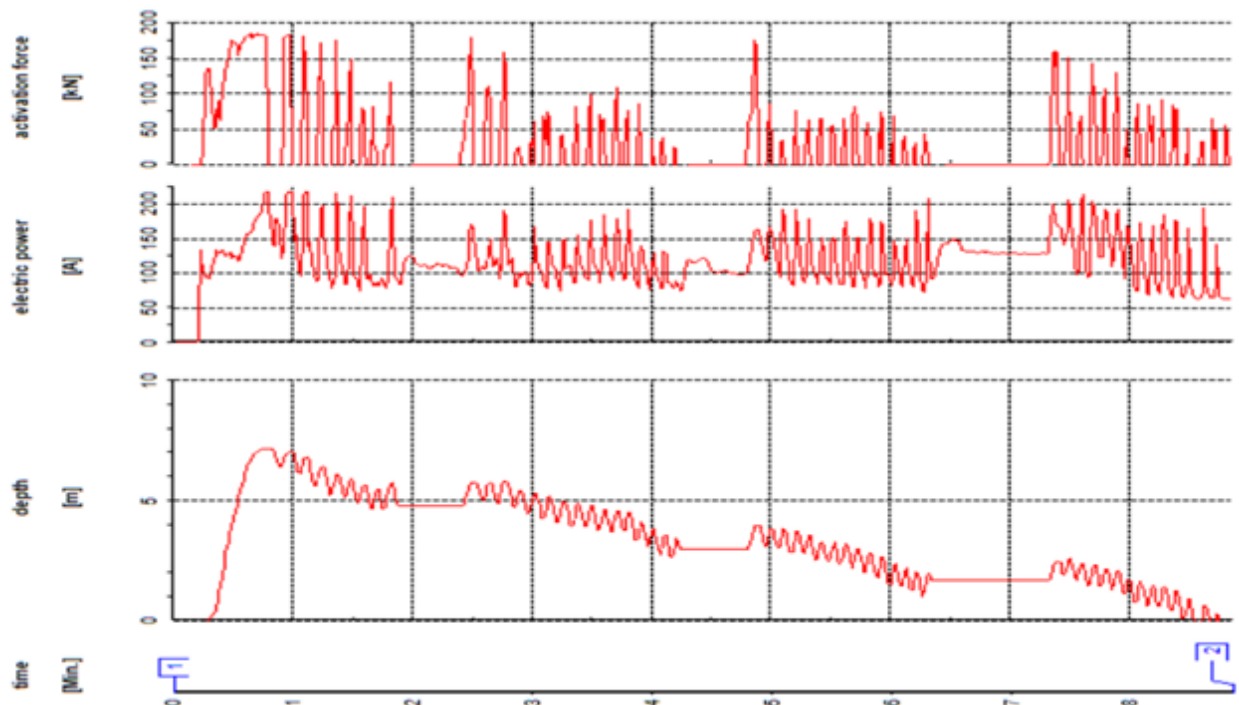


Figure 2: Example for an element protocol of a vibro-replacement column

of each column and the achievement of the required stiffness. Additionally, specific KPIs are computed for the complete element, which permit an evaluation of the total efficiency of the process.

The fusion of the real-time machine data, with geo-referencing data for the machine at each column, enables a location-based viewing and evaluation of the KPI. In this manner, systematic changes over the site can be determined and prediction of behavior can be made.

This data is displayed on the data monitor inside the rig during production for the rig operator, as well as recorded and documented in the element protocols. A typical element protocol of a vibro-replacement column shows - besides the production time and depth - also the electric power used by the vibrator (current in Ampere) and/or the activation force.

The real-time machine data is logged internally on an industrial PC and transmitted to a web server for external recording and evaluation. This server provides data services so that the data can be searched or accessed for further evaluations. A protocol is produced for each executed element, resulting in 20 to 70 element logs per rig and day which need to be evaluated to ensure the quality of each and every element. Of course, the quality of every column can be ensured in this manner; however, getting a good overview of the columns of the whole site, small deviations from column to column or abnormalities in a certain area of the site is difficult. Due to variations in sub-surface soil properties across the site, there is no average element that shows the whole site's characteristics. Consequently, no standard valid column can be detected via simple cross-correlation. Considering the data represented in Fig. 4, it can be seen that there is a systematic change in the maximum depth of columns required across the site. Therefore, there is no standard

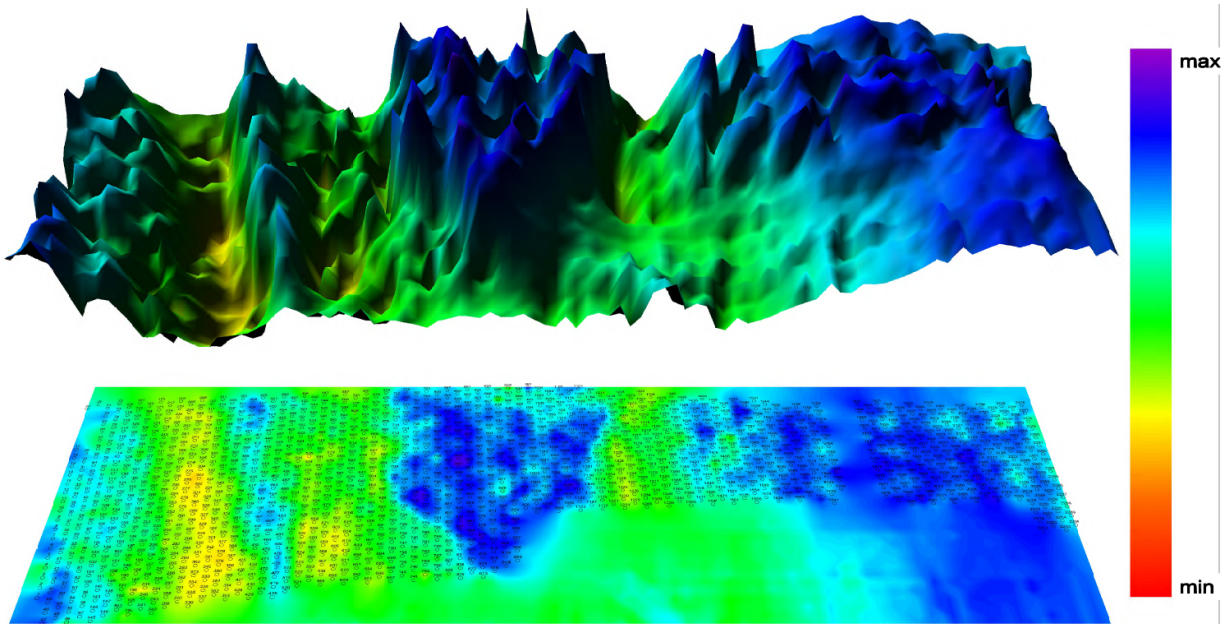


Figure 3: 2D- and 3D- evaluation of the parameter "built-in gravel material per column" with the VibroScan-tool.

valid column detectable for the whole site. Furthermore, coherences between different parameters sometimes get lost, when parameters are changing during the site, due to the large amount of data that would need to be viewed.

Evaluation tools, such as VibroScan [9], [10], help to preserve the overview of execution quality of a whole site. But such tools are only capable to show one single parameter at a time, such as the depth of all columns of a site, the built-in material per column-meter or the needed electric energy, for instance, in the lowest meter of a column, where the bearable soil is penetrated. These evaluations result in a 2D- or 3D-plan view of the chosen parameter.

However, coherences between different parameters in certain areas of the site are likely to be missed. For example, the production time of the columns can increase in an area of the site, due to idle times during the column production as a result of over-temperature of the vibrator. Probably this is caused by changed soil conditions – possibly a harder soil layer in this area? High ampere values or a high activation force as well as less volume of built-in gravel at a certain depth of all of these columns may confirm this suspicion.

The aim of the toolbox and collection of apps is to give a structure handling of data and meta-data, with the goal of improving the monitoring of the process as a whole. Fusing the data from multiple sources and providing an overview of different parameters on a site leads to an even better information about the in-situ soil conditions, the production process and the execution quality. It also yields insights into possible efficiency improvements that can be achieved. Furthermore, the data-centric nature of the implementation enables the creation of multiple data views as required

by different personnel, see for example Fig. 4, Fig. 5 and Fig. 6 for different views. The evidence-based validation or negation of a hypothesis is enabled by the structured availability of all the real-time machine data for a complete site.

3 Conceptual Design of the Toolbox

The concept behind this development is to provide a data-centric framework, which enables multiple hierarchical and structured views of real-time machine data, relating to the vibro ground improvement process. The aim is to provide an overview of all the points on any selected construction site, while maintaining a direct link to the details of the real-time machine data. The goal is to identify points which are possibly anomalous, enabling the timely initiation of mitigating actions. Additionally, the overview reveals systematic spatial changes in ground behavior over the site. The collection of applications is designed to be used by a site manager and does not need specialist knowledge of data analysis techniques. The system does, however, enable process and analysis experts to perform in-depth analysis of the data for a site or a specific point.



Figure 4: Geo referenced simultaneous presentation of two KPIs - one as color, one as symbol size for each point (column) of a construction site. Note: in this example, a series of planned but not executed points can be seen at the top (white circles). Furthermore, a systematic variation of the KPI (color) can be seen from the top left towards the bottom right; this reveals a systematic change in underground properties. The panels on the right permit the definition of filters for the KPIs, with the aim of detecting and marking outlier points. Additionally, points manually flagged as being exceptional can be indicated.

The hierarchical concept starts with the real-time machine data, which is segmented into the penetration and compaction phases using a rule-based approach. Then a series of KPIs is calculated for each point and used to form an in-memory hierarchical index into the point data files. This index provides the data required to produce both spatial and heat-map overviews for a site.

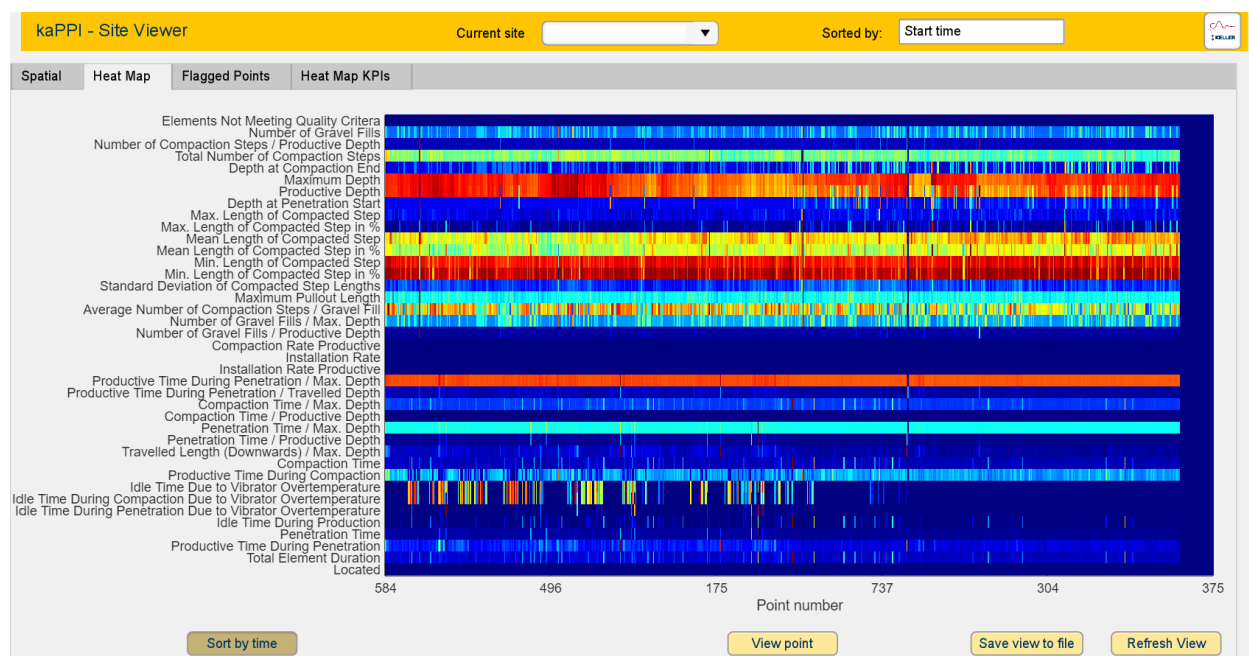


Figure 5: Heat map representation of selected KPIs for all points of the construction site (same data as in Fig. 4); sorted according to when they were executed. The non executed points can be seen on the right of this table. The heat map supports the visual detection of patterns over multiple KPIs.

A geo-referenced view of the points on a specific site is shown in Fig. 4. With this view, two out of a large collection of KPIs can be viewed simultaneously; one represented by the color and the other by the size of the point symbol. The KPIs to be viewed are dynamically selectable, using simple drop-down menus. An automatic detection of anomalous points is provided based on statistical definitions for an outlier; however, it is also possible to filter the KPIs to investigate other structure within the site data. In this view both outliers and flagged points can be marked automatically. Furthermore, a point can be selected and an additional application started to view the real-time machine data associated with that point, see Fig. 6 for an example.

A heat-map view of the same site as in Fig. 4 is shown in Fig. 5. This view provides an overview of all points on the site for multiple selectable KPIs. Correlations between KPIs and clustering of behavior are revealed, sorted either by the time when the point was executed or according to the point name. For example, see the clustering in the KPI idle time due to vibrator over-temperature.

A time-series of the machine data for a specific point is shown in Fig. 6; here a subset of channels has been selected for visualization. Additionally, the rule-based, automatic segmentation



Figure 6: Visualization of a selected sub set of the real time machine data; here for point 190. The real time machine data for each point in Fig. 4 and Fig. 5 are available in the system. They are directly linked to support the visualization of detailed machine behavior. The data is automatically segmented to indicate the penetration and compaction phases. This example shows a pre-drilled point, identified by the fact that the vibrator enters the ground without significant pull down force while the electric motor was even turned off. Anomalous points can be flagged and commented in this view.

into penetration and compaction phases can be seen in this figure. Specific characteristics of a point can be identified in the real-time machine data. In the case shown here this is a pre-drilled point in the first 4 m; identifiable by the fact that the vibrator enters to a significant depth without any significant pull-down force while the electric motor is even turned off.

Separate applications are provided to enable parameterization of the metadata for a site, the segmentation parameters and the KPIs. The structured management of multiple types of metadata proved to be a very important issue in the implementation of the framework. It is essential if the framework is to be generic and extendable with minimal resources.

4 Implementation Strategy

The implementation is based on the data, computation and visualization paradigm; whereby, it is data-centric. This means the real-time machine data is ingested into a pool of binary files. The computations analyze the real-time data to provide segmentation and KPI values for each point. These results are stored with the real-time data or in a point index, whichever is appropriate. The visualization applications can trigger computations; however, they access results exclusively via the data index or from the implemented data services. This ensures a consistent access to data and makes the implementation of multiple views simpler. It is akin to a publish and subscribe mechanism; whereby, one visualization may trigger a computation and the data services publish the results, so that they are simultaneously available for all views.

An object model has been implemented for the data, which consists of three objects:

1. An object to handle multi-channel time series; here it is called a *timetable*. For example, the real-time machine data is maintained in such timetables, as are events relating to the machine data.
2. A *table* object to handle tabular data, such as the point index which contains a table of KPIs.
3. A *structured type*, primarily used to manage metadata.

These objects are serialized to binary files in the Hierarchical Data Format (HDF5). The hierarchical data format (see [11]) is a set of file formats designed to organize and store large amounts of scientific data. The format supports the partial reading of data from the binary files on disk using B-trees. B-trees are a tree data structure suitable for handling large blocks of data; they enable search, access, insertion, and deletion of data and are proper for indexing table objects [12]. The use of B-trees to index table objects makes the format faster than SQL when dealing with time-series data and data summarized in tables. The use of objects to represent the time-series data permits the addition of functionality via the methods defined for the class. Each data set is stored with its associated metadata; consequently, when the object is passed to a method, both the data and metadata are available. Furthermore, the generic concept of events and segments are attached to the time-series object. This simplifies the handling of the phases and segmentation of the data for the vibro process. The use of standard objects, independent of specific content, permits the implementation of generic functions, e.g., a heat-map view. The goal is to enable a minimal resource implementation and extension of applications.

The structural implementation is centered around a data index. This is an in-memory object containing references to the binary files for each point and a table of KPIs for all points. The KPIs are organized into groups and members. This is similar to the concept of Google's Bigtable, whereby, the data in the table is in column format arranged into families (groups here) and addressable by name-value pairs. All KPI computations return their results to this index, from where the visualizations obtain their data and references to binary files. Consequently, additional views can

be implemented with minimal resources, since they only interact with the index, through a defined structure. The information and references required to populate the drop-down menus are automatically extracted from this index. This avoids the necessity to hard-code the content of the menus; furthermore, they automatically extend when new results are published in the index. To support multi-language development, the machine data channels and KPIs are assigned aliases and display names in the metadata. Consequently, a machine may use local languages to support the operator, while an automatic mapping to the alias is performed to simplify development of the applications.

5 Case Study

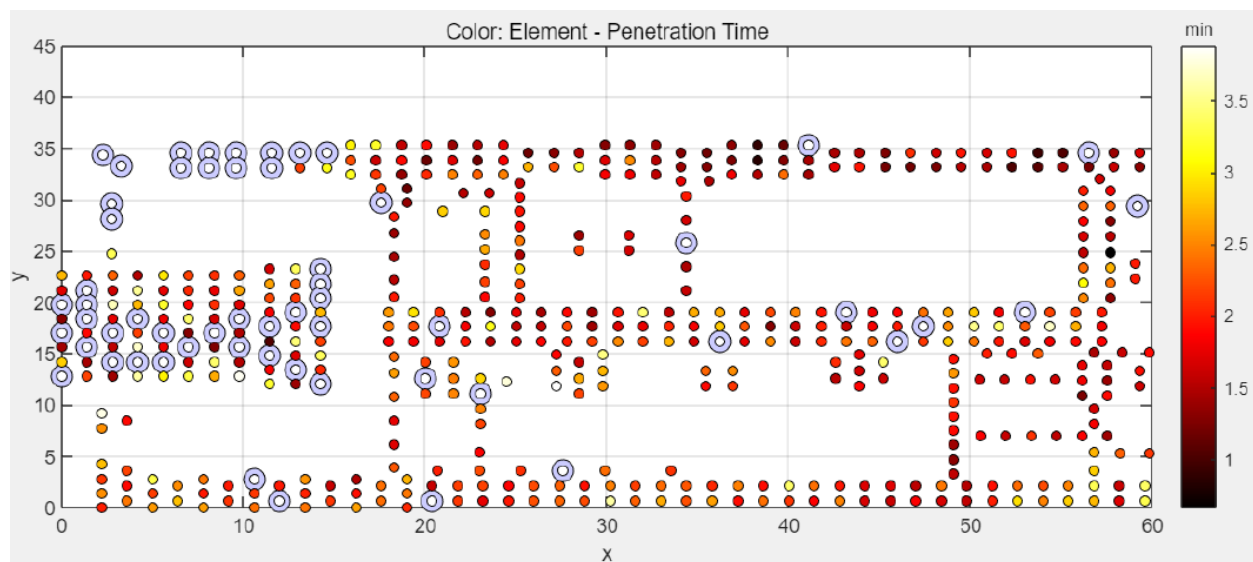


Figure 7: Outliers in the KPI penetration time of the executed points ; the concentration on the left part of the site

One example for the application of the framework to a test site is shown in the following figures. A poorer performance was observed for one production shift, with respect to the performance estimated used to calculate the site. The evaluation of the *penetration time* of the points, see in Fig. 7, revealed that the outliers are concentrated on the left part of the site.

Another KPI for the travelled length of the vibrator in the penetration process in Fig. 8 leads to similar results. Outliers in Fig. 7 are concentrated in the same area of the site as the outliers in Fig. 8. The travelled length of the vibrator during the penetration phase is high, when the vibrator has to be pulled back and pushed into the soil again several times as a result of a high soil density.

Comparing the element protocols of a typical point on the test site in Fig. 9 with one of the outlier-points No. 465 in Fig. 10 shows the longer penetration phase of point 465, as well as the larger travelled length of the vibrator in the Depth-chart as a result of the modified execution procedure due to the stiffer soil conditions.

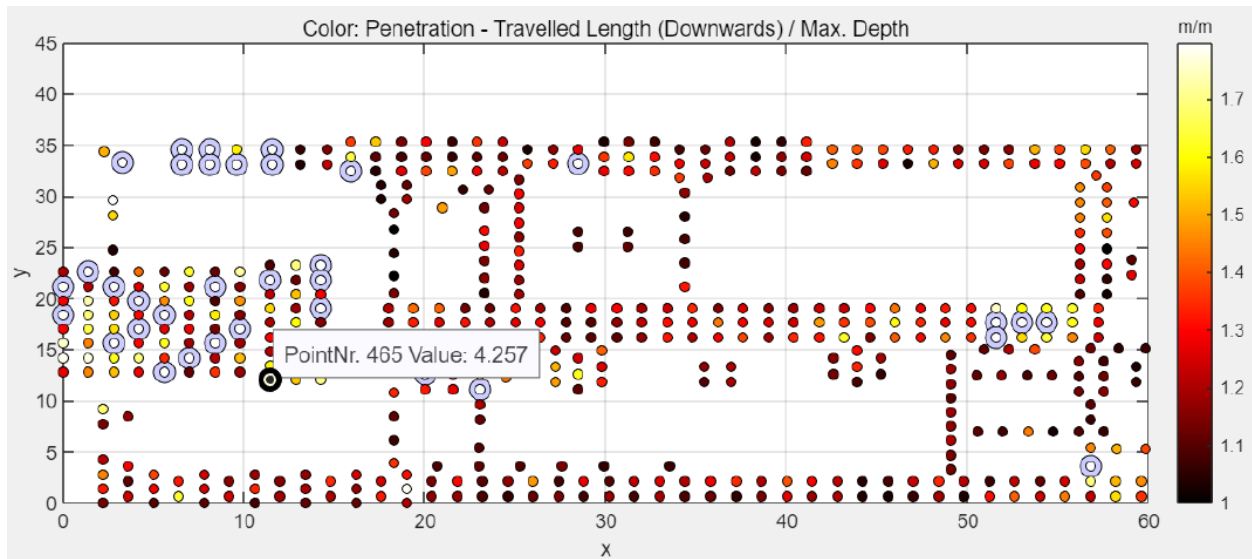


Figure 8: Penetration phase KPI: ratio of traveled length to max. penetration depth of the vibrator . This is a result for a reciprocating motion being required to penetrate the ground; with this, the cause for the loss in efficiency as seen in Fig. 7 has been identified.

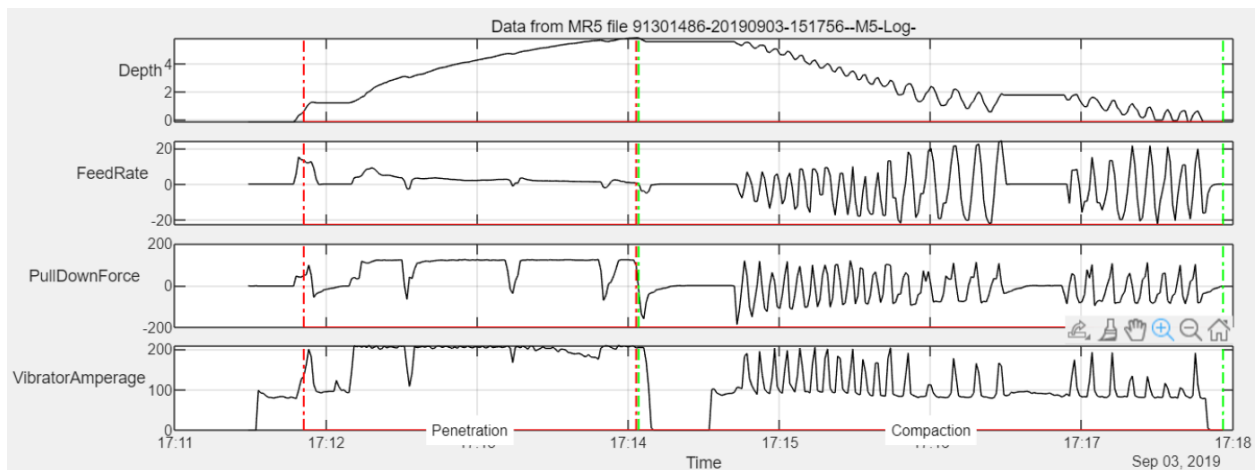


Figure 9: Element protocol of a standard point on the test site shown in Fig. 7

The automatic analysis of the site data led to the information, that approx. 10% of the executed points needed abnormal additional time for the penetration process. The outlier points are marked automatically for the site manager for further, more detailed inspection. The site manager noticed from a detailed observation of these points, the result of the lower shift performance on this site - in that case it was an in-situ soil of a higher density - as expected in one area of the site.

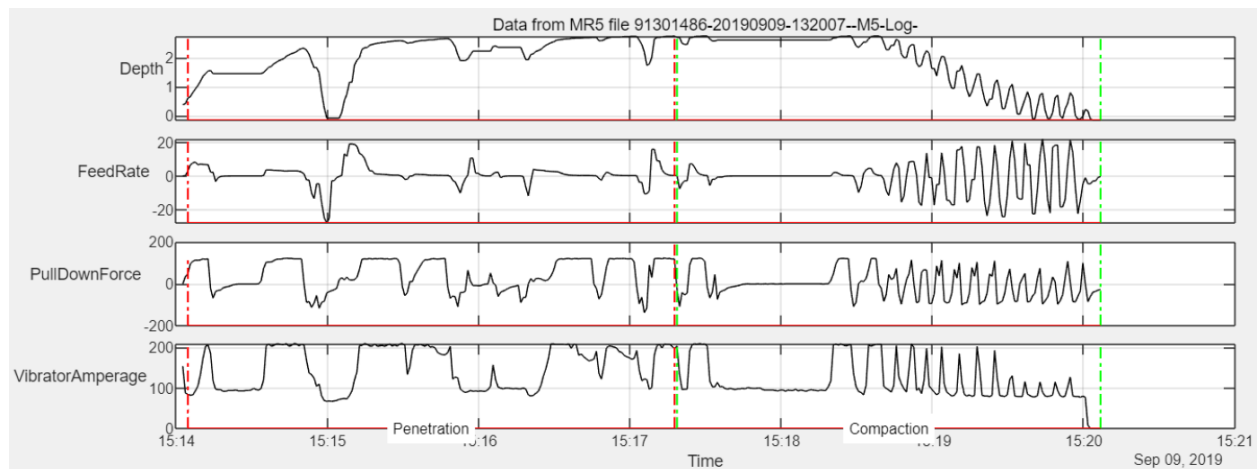


Figure 10: Element protocol of an outlier in penetration time of Fig. 7 and a high ratio in travelled length to max. penetration depth of Fig. 8 (point No. 465). Note: the vibrator needed to be extracted during the penetration phase at time 15:15, this is due to encountering soil with an unexpectedly high stiffness. This is also evidenced in the pull down force and vibrator amperage, both indicating that a very high stiffness had been encountered in the soil.

6 Conclusions and Outlook

The toolbox and framework presented in this paper are a significant extension of the current partial digitalization of the construction site. The fusion of data from different sources into a structured availability of all real-time machine data for a complete site, has made new analysis approaches possible. For example, it has been demonstrated that the systematic spatial variation in process values over a site can be efficiently viewed. The data-centric approach also enables the viewing of multiple KPIs for all points as a heat-map; providing a good overview of the performance over the site as a whole. The structured management of data and the required corresponding metadata enables the extraction of a higher degree of added value from the available data.

The automatic segmentation of the data, together with the computation of a large number of KPIs, relieves the site manager from arduous tasks; this also avoids the errors often occurring during tedious work. Hence, the time dedicated to quality control can either be reduced or spent more effectively. Correlations between several KPIs can reveal specific characteristics like a change of sub-soil conditions that might have been overlooked with classical quality control procedures.

Future work will focus on extending the functionality of the system. For example, to implement automatic operations-recognition, so that human behavior can be separated from sub-surface properties which can only be observed indirectly. Currently deep learning is being investigated as a means of learning the implicit structures in the processes reflected in the data, this is facilitated by the structured interface to the complete data. The combination of classical descriptive statistics for the KPI, with the results of variational auto-encoders should enable a more complete description

of the process from the data. In particular, it will reduce the number of false positives, i.e., judging an element as being positive in terms of quality, although it is not; similarly, for false negatives.

Bibliography

- [1] M. Bilal, L. O. Oyedele, J. Qadir, K. Munir, S. O. Ajayi, O. O. Akinade, H. A. Owolabi, H. A. Alaka, and M. Pasha, “Big data in the construction industry: A review of present status, opportunities, and future trends,” *Advanced Engineering Informatics*, vol. 30, no. 3, pp. 500–521, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474034616301938>
- [2] N. Konstantopoulos, P. Trivellas, and P. Reklitis, “A conceptual framework of strategy, structure and innovative behaviour for the development of a dynamic simulation model,” *AIP Conference Proceedings*, vol. 963, no. 2, pp. 1070–1073, 2007. [Online]. Available: <https://aip.scitation.org/doi/abs/10.1063/1.2835927>
- [3] A. Schweigkofler, G. P. Monizza, E. Domi, A. Popescu, J. Ratajczak, C. Marcher, M. Riedl, and D. Matt, “Development of a digital platform based on the integration of augmented reality and bim for the management of information in construction processes,” in *Product Lifecycle Management to Support Industry 4.0*, P. Chiabert, A. Bouras, F. Noël, and J. Ríos, Eds. Cham: Springer International Publishing, 2018, pp. 46–55.
- [4] S. H. Khajavi, N. H. Motlagh, A. Jaribion, L. C. Werner, and J. Holmström, “Digital twin: Vision, benefits, boundaries, and creation for buildings,” *IEEE Access*, vol. 7, pp. 147 406–147 419, 2019.
- [5] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, “A review on outlier/anomaly detection in time series data,” *ACM Comput. Surv.*, vol. 54, no. 3, apr 2021. [Online]. Available: <https://doi.org/10.1145/3444690>
- [6] T. chung Fu, “A review on time series data mining,” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0952197610001727>
- [7] V. Runkana, *Digital Twins for Manufacturing*, 12 2018, p. 97.
- [8] “Digital Twins in Logistics. a DHL perspective on the impact of digital twins on the logistics industry,” <https://www.dhl.com/content/dam/dhl/global/core/documents/pdf/glo-core-digital-twins-in-logistics.pdf>, accessed 02-06-2021.
- [9] E. Falk, A. Zöhrer, and G. Strauch, “Entwicklung der Tiefenverdichtung - Von der händischen Datenaufzeichnung zur automatischen Visualisierung,” 2011.

- [10] A. Zöhrer and J. Wondre, “Neue Entwicklungen und Anwendungen von VibroScan,” 2012.
- [11] “HDF5,” <https://www.hdfgroup.org/>, accessed 12-06-2021.
- [12] “2011 IEEE 27th International Conference on Data Engineering,” *2011 IEEE International Conference on Data Engineering (ICDE 2011)*, 2011.

Chapter 6

Summary and Future Work

1 Summary

The methods presented in this thesis summarize the tools commonly used in evidence based change detection from discrete data. The following methods were developed in this thesis:

1. An algorithm for discontinuity detection of any non-negative integer order. The method proposed uses double-sided Taylor approximations of data as a measure of the difference of the left and right derivatives at a point. It bears similarities to the definition of Lipschitz continuity, where the first derivatives are bound by a constant. The algebraic formulation using matrix algebra allows for a neat expression of the solution and an easy implementation of the algorithm in MATLAB .
2. A framework for solving generalized constrained inverse problems, emanating from cyber-physical systems. It includes the implementation of both pseudo-spectral and finite-difference method constraints, which allow for solving a very wide array of problems.
3. New algebraic formulations for discretizing problems in optimal control. The method used avoids the Hamiltonian approach and provide accurate approximations to solutions for numerically and physically stiff systems.
4. Modeling periodic functions without spectral leakage or Gibbs error using the variable projection method. This method provides highly stable and numerically efficient solutions, since the non-linear problem needed to be solved is only one dimensional.
5. Consequent use of mathematical methods to extract desired information from large volumes of industrial data. This shows the interdisciplinary nature of data science in industrial applications.

2 Future Work

The following list contains open problems and directions for future work.

1. The discontinuity detection algorithm can be extended to allow for different definitions of continuity. Discrete data is always discontinuous in the classical mathematical sense. Consequently, new definitions that have significance need to be determined. Initial research results indicate that there is a great significance in data science for detecting locations in time series where, e.g. $y(t)$ is discontinuous but $y'(t), y''(t)$ could be continuous. Such cases are not covered by the classical C^n continuity definition. An additional open question is, whether data sets from industrial applications can be found demonstrating these kinds of discontinuities? If so, these generalizations would become more applicable.
2. There are many methods for approximating data with rational functions, which are quotients of polynomials. Can the variable projection method be used to do this sensibly? Will it perform better than current methods?
3. Solving optimal control problems for systems with a proportional-integral-derivative (PID) controller. The methods presented here can be used to determine the optimal coefficients k_p, k_i, k_d for the proportional, integral and derivative terms respectively. This is done by introducing more regularization terms in the cost function which needs to be minimized, also called Tikhonov regularization. The issue currently is, that there are three regularization terms (one for each coefficient) which need to be adjusted accordingly, to determine the optimal coefficients, and this translates into a multi-dimensional problem which requires a lot of computational power to be solved in this manner.
4. Possibly the theoretically most demanding proposal for future research is to extend the work presented in the paper "A Computational Framework for Generalized Constrained Inverse Problems" to deal with discrete calculus of variations in combination with machine learning. Some mechanical problems have been addressed and solved in this manner. *Physics Informed Autoencoders* have been implemented in collaboration with my colleagues at the University of Leoben and the Know-Center Graz. This opens the door to the current hot research topic of hybrid learning.