

A Schur Complement Method for Optimum Experimental Design in the Presence of Process Noise^{*}

Adrian Bürger^{*,**} Dimitris Kouzoupis^{**}
Angelika Altmann-Dieses^{*} Moritz Diehl^{**,***}

^{*} Faculty of Management Science and Engineering,
Karlsruhe University of Applied Sciences, Germany

^{**} Department of Microsystems Engineering,
University of Freiburg, Germany

^{***} Department of Mathematics, University of Freiburg, Germany

Abstract: Optimization problems arising in Optimum Experimental Design (OED) applications require repeated covariance matrix evaluations of the underlying Parameter Estimation (PE) problems. The complexity of this task grows quickly with the problem dimensions, especially when process noise is considered. In this paper, a Schur complement method is proposed to alleviate this problem by translating the covariance matrix evaluation into the solution of a sparse linear system and the inversion of a small-scale matrix. The method is used as a building block for the open-source software package CASIOPEIA, a powerful and easy-to-use environment for OED and PE. The performance of the software and the proposed Schur complement method is assessed on a numerical example.

Keywords: Input and excitation design, Software for system identification

1. INTRODUCTION

In Parameter Estimation (PE) problems, the amount of information that can be extracted on the unknown parameters strongly depends on whether the conducted experiment was able to sufficiently excite the relevant system dynamics. A useful indicator for assessing the quality of the estimation results is the *covariance matrix* of the estimated parameters, see e. g. Walter and Pronzato (1997), Pronzato (2008). High values for a parameter's variance indicate that the setup was not optimal for determining its true value – even if the fitting to the given measurements is satisfactory – since either the inputs of the system were not well designed or the adopted model was inadequate. High covariance values may for instance occur when the values of some model parameters strongly depend on the values of others.

Since it is favorable not only to evaluate the information content of one experiment, but to ex ante determine meaningful experimental setups to reduce effort and cost, the covariance matrix is further used within Optimum Experimental Design (OED) to design experiments that result in a higher confidence on the estimated parameter values¹ (Pronzato, 2008), (Bock et al., 2013). In OED, the

covariance matrix is part of the objective of an optimization problem, which makes the efficient evaluation of the matrix crucial and every increase in computational effort problematic. Different approaches exist to calculate and use the covariance matrix in this context, see e. g. Bauer et al. (2000), Körkel (2002), Körkel et al. (2004), Bock et al. (2007), Kostina and Kostyukova (2012).

If a system is subject to noise that affects its dynamics and this noise is not explicitly considered within the PE problem, the estimation results may be inaccurate. As a remedy to this, additional degrees of freedom can be introduced in the PE problem formulation that capture such *process noise* and improve the estimation results. This approach is customary in the field of optimization based state and parameter estimation, see e. g. Rao et al. (2003). However, these additional degrees of freedom increase the effort to evaluate the covariance matrix, which can in turn have a negative impact on the solution times of OED problems.

Within this paper, a method to compute the covariance matrix based on the Schur complement is presented, which relies on the solution of a sparse linear system and the inversion of a small-scale matrix (with dimension equal to the number of estimated parameters). Although the method is presented for the case of process noise, it is deliberately kept generic and thus applicable to other types of noise as well. To facilitate direct use within PE and OED applications, the open-source software package CASIOPEIA that contains an implementation of the proposed method is introduced and its performance is assessed on a numerical example.

^{*} Support by the EU via ERC-HIGHWIND (259 166), ITN-TEMPO (607 957), and ITN-AWESCO (642 682), by DFG in context of the Research Unit FOR 2401 and by Ministerium fuer Wissenschaft, Forschung und Kunst Baden-Wuerttemberg (Az: 22-7533.-30-20/9/3) is gratefully acknowledged.

¹ Other OED approaches aim at minimizing the cost of the identification experiment while guaranteeing an acceptable control performance, see e. g. Annergren (2016).

The paper is organized as follows. Section 2 shows how the consideration of process noise increases the effort to compute the covariance matrix for both unconstrained and equality constrained parameter estimation problems. In Section 3 the Schur complement method for the covariance matrix evaluation is presented. The requirements for using this approach within OED are discussed in Section 4. Finally, CASIOPEIA is introduced in Section 5 and its performance is assessed in the numerical case study of Section 6.

1.1 Notation

In the following, the accent \bar{a} is used to denote a variable that is fixed within an optimization problem. The accent \tilde{a} means that the variable is polluted by noise and \hat{a} that it is the solution of an optimization problem.

2. COVARIANCE MATRICES IN PARAMETER ESTIMATION

In the following, we introduce the notion of covariance matrices in the context of both unconstrained and constrained least-squares PE. Least squares PE is a common method for formulation of linear and nonlinear PE problems, see e. g. Nocedal and Wright (2006). Further, we show how the consideration of process noise increases the computational effort to evaluate the covariance matrices.

2.1 Evaluation of covariance matrices

Consider a possibly nonlinear, unconstrained least-squares PE problem of the form

$$\hat{p} = \arg \min \|\tilde{y} - h(p; \bar{u})\|_{\Sigma_y^{-1}}^2, \quad (1)$$

with unknown parameters $p \in \mathbb{R}^{n_p}$ and fixed, noise-free inputs $\bar{u} \in \mathbb{R}^{n_u}$ that are applied to the system during a certain experiment.² The measurement values $\tilde{y} \in \mathbb{R}^{n_y}$ are polluted by additive, zero-mean Gaussian noise $\mathcal{N}(0, \Sigma_y)$ with $\Sigma_y \in \mathbb{R}^{n_y \times n_y}$ the (co-)variances of the measurements.³ The weighting matrix Σ_y^{-1} is the inverse of Σ_y . The model response $h(\cdot) \in \mathbb{R}^{n_y}$ may e. g. come from a single shooting discretization of a dynamic system, see e. g. Press (2007). It is assumed that the inputs are piecewise constant within one discretization interval and the measurements are taken at the discretization time points.

The covariance matrix can now be used for assessing the quality of the estimation results \hat{p} of (1) for the experimental setup determined by \bar{u} . An estimate for this matrix is

$$\Sigma_{\hat{p}} = \left(\left(\frac{\partial h}{\partial p}(\hat{p}; \bar{u}) \right)^\top \Sigma_y^{-1} \left(\frac{\partial h}{\partial p}(\hat{p}; \bar{u}) \right) \right)^{-1}, \quad (2)$$

² The vector \bar{u} contains the vectorized inputs of all time steps.

³ The vector \tilde{y} contains the vectorized measurements of all time points. The noise of the measurements is assumed to be uncorrelated between time points, but can be correlated for measurements at one time point, i. e., the matrix Σ_y can be diagonal or block-diagonal.

as in Bauer et al. (2000). In order to obtain an unbiased estimate for the covariance matrix, $\Sigma_{\hat{p}}$ has to be scaled by a factor

$$\beta_{\hat{p}} = \frac{\|\tilde{y} - h(\hat{p}; \bar{u})\|_{\Sigma_y^{-1}}^2}{(N - n_p)}, \quad (3)$$

where N is the number of measurements of the experiment. In case (1) is a linear PE problem, $\beta_{\hat{p}} \Sigma_{\hat{p}}$ is an unbiased estimator for the covariance matrix of (1) (Ljung, 1999). In case (1) is nonlinear, $\beta_{\hat{p}} \Sigma_{\hat{p}}$ is an approximation for the covariance matrix.

2.2 Covariance matrix evaluation under process noise

If the system is subject to process noise, additional degrees of freedom $w \in \mathbb{R}^{n_w}$ can be introduced in the optimization problem to yield more accurate results. For zero-mean Gaussian process noise $\mathcal{N}(0, \Sigma_w)$ with (co-)variances $\Sigma_w \in \mathbb{R}^{n_w \times n_w}$, the PE problem becomes

$$(\hat{p}, \hat{w}) = \arg \min \|\tilde{y} - h(p, w; \bar{u})\|_{\Sigma_y^{-1}}^2 + \|w\|_{\Sigma_w^{-1}}^2. \quad (4)$$

The covariance matrix of (4) is now obtained from

$$\Sigma_{\hat{v}} = \left(\left(\frac{\partial h^*}{\partial v}(\hat{v}; \bar{u}) \right)^\top \begin{bmatrix} \Sigma_y^{-1} & 0 \\ 0 & \Sigma_w^{-1} \end{bmatrix} \left(\frac{\partial h^*}{\partial v}(\hat{v}; \bar{u}) \right) \right)^{-1} \quad (5)$$

with $v^\top = [p^\top \ w^\top] \in \mathbb{R}^{n_v}$ and $h^*(\cdot)^\top = [h(\cdot)^\top \ w^\top]$. The scaling factor becomes

$$\beta_{\hat{v}} = \frac{\|\tilde{y} - h(\hat{p}, \hat{w}; \bar{u})\|_{\Sigma_y^{-1}}^2 + \|\hat{w}\|_{\Sigma_w^{-1}}^2}{(N - n_p)}. \quad (6)$$

The matrix $\Sigma_{\hat{p}}$ is now a submatrix of $\Sigma_{\hat{v}}$, located at the upper left corner due to the ordering of the optimization variables. The dimensions of $\Sigma_{\hat{v}}$ are indicated in (7).

$$\Sigma_{\hat{v}} = \left[\begin{array}{cc} \overbrace{\Sigma_{\hat{p}}}^{n_p} & \overbrace{\Sigma_{\hat{p}, \hat{w}}^\top}^{n_w} \\ \Sigma_{\hat{p}, \hat{w}} & \Sigma_{\hat{w}} \end{array} \right] \left. \begin{array}{l} \} n_p \\ \} n_w \end{array} \right\} \quad (7)$$

This implies that the computational effort for obtaining $\Sigma_{\hat{p}}$ has increased, despite the fact that the content of the other submatrices of $\Sigma_{\hat{v}}$ is typically not of interest.

2.3 Covariance matrices for equality constrained problems

For PE of dynamic systems, it is usually favorable to use multiple shooting instead of single shooting for discretization of the dynamics (Bock, 1983). This results in equality constrained PE problems.

Consider the equality constrained least-squares PE problem of the form

$$\begin{aligned} \underset{p, x, w}{\text{minimize}} \quad & \|\tilde{y} - h(p, x, w; \bar{u})\|_{\Sigma_y^{-1}}^2 + \|w\|_{\Sigma_w^{-1}}^2 \\ \text{subject to} \quad & g(p, x, w; \bar{u}) = 0, \end{aligned} \quad (8) \quad AX = C, \quad (13)$$

with $g \in \mathbb{R}^{n_g}$ coming from a multiple shooting discretization of a dynamic system. Vector x contains the optimization variables introduced for the system's states.

Let $z^\top = [p^\top \ x^\top \ w^\top]$ be the concatenation of the optimization variables and $B(\cdot)$ the Hessian approximation of the Generalized Gauss-Newton method (Bock, 1983). We define the Karush-Kuhn-Tucker (KKT) matrix K of (8) as

$$K(z, \bar{u}) = \begin{bmatrix} B(z; \bar{u}) & g_z(z; \bar{u})^\top \\ g_z(z; \bar{u}) & 0 \end{bmatrix}, \quad (9)$$

with $g_z(\cdot)$ the Jacobian of $g(\cdot)$ with respect to z . The covariance matrix $\Sigma_{\hat{z}}$ of the equality constrained least-squares PE problem in (8) is now equal to the upper left submatrix of K^{-1} , i. e.,

$$K^{-1}(\hat{z}; \bar{u}) = \begin{bmatrix} \overbrace{M_{11} \quad M_{21}^\top}^{n_z} \\ \overbrace{M_{21} \quad M_{22}}^{n_g} \end{bmatrix} \begin{matrix} \} n_z \\ \} n_g \end{matrix} \quad (10)$$

with $\Sigma_{\hat{z}} = M_{11}$ (Kostina and Kostyukova, 2012). Note that $\Sigma_{\hat{z}}$ contains all (co-)variances for all optimization variables z . Again, due to the ordering of the variables p , x , and w in z , only the upper left submatrix $\Sigma_{\hat{p}}$ of $\Sigma_{\hat{z}}$ contains relevant information on the quality of \hat{p} . For a multiple shooting problem (8) and a corresponding single shooting problem (4), the resulting covariance matrices $\Sigma_{\hat{p}}$ coincide.⁴

3. COVARIANCE MATRIX EVALUATION VIA A SCHUR COMPLEMENT METHOD

In this section, an efficient approach is presented to obtain $\Sigma_{\hat{p}}$ from K using the Schur complement.

3.1 Efficient matrix inversion using linear solvers

In principle, $\Sigma_{\hat{p}}$ can directly be obtained from K^{-1} as

$$\Sigma_{\hat{p}} = Z_p^\top (K^{-1} Z_p), \quad (11)$$

with Z_p being a projection matrix from K^{-1} to $\Sigma_{\hat{p}}$, i. e.,

$$Z_p = \begin{bmatrix} \overbrace{I}^{n_p} \\ 0 \end{bmatrix} \begin{matrix} \} n_p \\ \} n_z + n_g - n_p \end{matrix} \quad (12)$$

and $I \in \mathbb{R}^{n_p \times n_p}$ the identity matrix.

For this calculation to be efficient, it is crucial not to explicitly calculate K^{-1} , but to evaluate $(K^{-1} Z_p)$ directly. Consider a numerical (or symbolic) linear solver `solve()` that is able to solve a linear system of the form

⁴ In single shooting, w also contains degrees of freedom for estimation of the initial states.

with $A \in \mathbb{R}^{n \times n}$, $C \in \mathbb{R}^{n \times m}$ known and $X \in \mathbb{R}^{n \times m}$ unknown as in

$$X = \text{solve}(A, C). \quad (14)$$

Then, `solve()` can be used to directly compute $\Sigma_{\hat{p}}$ as

$$\Sigma_{\hat{p}} = Z_p^\top \cdot \text{solve}(K, Z_p). \quad (15)$$

without explicitly computing K^{-1} beforehand which reduces the computational effort significantly, see e. g. Golub and Loan (1996). The multiplication with Z_p^\top is only a selection of elements from `solve`(K, Z_p) and therefore it introduces no additional computational cost. Further gain in efficiency is obtained if `solve()` can exploit the sparsity of the involved matrices.

3.2 Inversion of submatrices using the Schur complement

When K is partitioned as

$$K = \begin{bmatrix} \overbrace{K_{11} \quad K_{21}^\top}^{n_p} \\ \overbrace{K_{21} \quad K_{22}}^{(n_z - n_p + n_g)} \end{bmatrix} \begin{matrix} \} n_p \\ \} (n_z - n_p + n_g) \end{matrix} \quad (16)$$

the Schur complement can be used to compute the covariance matrix $\Sigma_{\hat{p}}$, which corresponds to the upper left block of the inverse of (16), as

$$\Sigma_{\hat{p}} = (K_{11} - K_{21}^\top (K_{22}^{-1} K_{21}))^{-1}. \quad (17)$$

Note that K_{22} is invertible if the Nonlinear Programming (NLP) problem that corresponds to the problem with all parameters p fixed, satisfies the Second Order Sufficient Condition for local optimality (Nocedal and Wright, 2006). This assumption corresponds to a well-posed parameter estimation problem and it is for example the case when the constraints come from a multiple shooting discretization and the disturbances are penalized with positive definite weighting matrices.⁵

Again, it is crucial to compute $(K_{22}^{-1} K_{21})$ at once using the previously introduced `solve()` operator. In this case, $\Sigma_{\hat{p}}$ is evaluated as

$$\Sigma_{\hat{p}} = (K_{11} - K_{21}^\top \cdot \text{solve}(K_{22}, K_{21}))^{-1}. \quad (18)$$

In order to obtain an unbiased estimator, $\Sigma_{\hat{p}}$ needs now to be scaled by the factor

$$\beta_{\hat{z}} = \frac{\|\tilde{y} - h(\hat{p}, \hat{x}, \hat{w}; \bar{u})\|_{\Sigma_y^{-1}}^2 + \|\hat{w}\|_{\Sigma_w^{-1}}^2}{(N - n_p)}. \quad (19)$$

⁵ The authors in Walter and Pronzato (1997) apply this method on covariance matrices of unconstrained PE problems where the inverse of K_{22} might not exist and mention possible remedies for that case.

A comparison of the evaluation times for the discussed approaches is provided in Section 6.5, using a numerical example.

4. APPLICATION TO OPTIMUM EXPERIMENTAL DESIGN

Within this section, an introduction to OED is provided and the relevant optimization problem is formulated. Based on this formulation, the requirements for a dedicated software implementation are derived.

4.1 Optimum experimental design problem formulation

The aim of OED is to generate optimized inputs \hat{u} for an experiment that eventually reveal more information on the unknown parameters p than a less educated guess \bar{u} .

These optimized inputs can be obtained from the solution of an optimization problem of the form

$$\begin{aligned} & \underset{x, u}{\text{minimize}} && \Phi(\Sigma_p(x, u; \bar{p}, \bar{w})) \\ & \text{subject to} && g(x, u; \bar{p}, \bar{w}) = 0, \\ & && x_{\min} \leq x \leq x_{\max}, \\ & && u_{\min} \leq u \leq u_{\max}, \end{aligned} \quad (20)$$

where the objective is a scalar information function $\Phi(\cdot)$ of the covariance matrix $\Sigma_p(\cdot)$ for the corresponding PE problem with free optimization variables x, u . The problem is also subject to the constraints $g(\cdot)$ imposed by the system dynamics (in form of, e.g., a multiple shooting discretization) and possibly upper and lower bounds on states x and inputs u . For further information on OED see e.g. Bock et al. (2013).

In (20), \bar{p} is an initial guess of sufficient quality for the true value of p , which can be obtained from the literature, previous experiments or other prior knowledge on the considered system. The process noise \bar{w} is assumed to be all 0.⁶

4.2 Software implementation requirements

When (20) is solved with a derivative-based NLP solver, $\Phi(\Sigma_p(\cdot))$ and its derivatives are evaluated frequently. This implies that an efficient method for evaluating Σ_p is crucial. Furthermore, it must be possible to efficiently compute up to second order derivatives for the components of (20) to high accuracy, despite the fact that $\Phi(\Sigma_p(\cdot))$ already contains first order derivatives of components of the PE problem that underlies the experimental design problem.

Accurate derivatives can be obtained efficiently by Algorithmic Differentiation (AD) (Griewank, 2000). However, this technique is only applicable when all expressions used within the construction of the optimization problem are differentiable and the differentiation rules of every expression – including calls to sophisticated linear solvers and numerical integrators – are known.

⁶ Even though \bar{w} is 0, the consideration of process noise in the underlying PE problem influences the construction of Σ_p as shown in the previous sections. Therefore, w still has to be included in the construction of the objective.

5. SOFTWARE IMPLEMENTATION

Based on the reasoning of the previous section, the choice of the software framework is motivated and the software package CASIOPEIA is introduced.

5.1 Choice of software framework

A framework that satisfies the requirements formulated in Section 4.2 and with this facilitates the use of (18) for formulation and solution of (20) is CASADI, introduced by Andersson (2013). It does so by providing the possibility to automatically generate derivative expressions using AD principles for complex symbolic expressions as well as for calls to external solvers that have been coupled to CASADI in a way that supports AD. These expressions can be used to formulate and solve optimization problems, while all derivatives that are needed by the NLP solvers that interact with CASADI are generated automatically.

5.2 Introduction of the software package CASIOPEIA

The implementation of the presented methods is included in the open-source Python module CASIOPEIA (Bürger, 2016). In addition to PE and OED features, a collection of tools is provided in order to help the user to assess the quality of the estimation results. The implemented methods can be used via a unified user interface. Furthermore, multiple shooting and collocation discretizations for dynamic systems are generated automatically. A detailed documentation including introductory examples can be found in (Bürger, 2016).

For the `solve()` command introduced in Section 3.1, the numerical linear solver CSPARSE (Davis, 2006) is used, which is available as an infinitely many times differentiable function through CASADI and is also able to exploit the sparsity structure of the involved matrices.

6. NUMERICAL CASE STUDY

Within this section, the methods implemented in CASIOPEIA are applied to a numerical example and their performance is assessed in terms of solution quality and computational effort. The code for the example can be found in (Bürger, 2016).

6.1 Model description

The model used within this example describes the dynamics of a quarter vehicle, a system that is used in analysis of road performance and driving characteristics of vehicles. Within the model, a spring-damper-system depicts a vehicle's chassis that is coupled to another spring that reflects the behavior of the tire. A schematic depiction of the system and a description of the symbols used within the model is given in Figure 1. The dynamics of the quarter vehicle can be described by a system of ordinary differential equations (Mitschke, 2014)

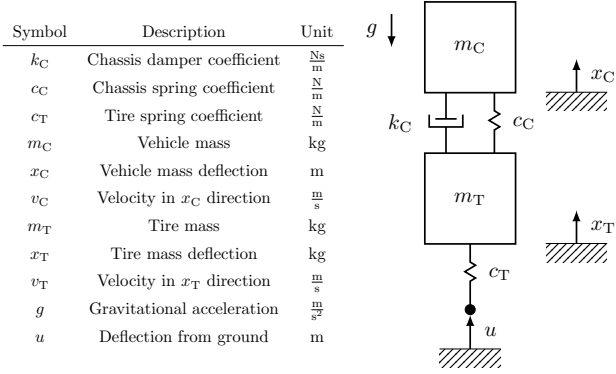


Fig. 1. Schematic depiction of the quarter vehicle system model and description of symbols.

$$\begin{aligned}
\dot{x}_T(t) &= v_T(t) \\
\dot{v}_T(t) &= \frac{k_C}{m_T}(v_C(t) - v_T(t)) + \frac{c_C}{m_T}(x_C(t) - x_T(t)) \\
&\quad - \frac{c_T}{m_T}(x_T(t) - u(t)) \\
\dot{x}_C(t) &= v_C(t) \\
\dot{v}_C(t) &= -\frac{k_C}{m_C}(v_C(t) - v_T(t)) - \frac{c_C}{m_C}(x_C(t) - x_T(t)).
\end{aligned} \tag{21}$$

The system states $x^\top = [x_T, v_T, x_C, v_C]$ describe the positions of the involved masses and their current velocities. The parameters $p^\top = [k_C, c_C, c_T]$ describe spring and damper coefficients to be determined experimentally. The input u describes the deflection of the tire bottom from the ground, which is either introduced by random road undulation during driving, or applied by an apparatus in order to run certain experiments for the system.

6.2 Parameter estimation

Sample data \bar{y} of all system states are obtained by simulation of (21) for a time horizon of $t_f = 5$ s at a sampling rate of $f_s = 20$ Hz with assumed true parameters

$$p_{\text{true}} = \begin{pmatrix} k_{C,\text{true}} \\ c_{C,\text{true}} \\ c_{T,\text{true}} \end{pmatrix} = \begin{pmatrix} 4.0 \cdot 10^3 \frac{\text{Ns}}{\text{m}} \\ 4.0 \cdot 10^4 \frac{\text{N}}{\text{m}} \\ 1.6 \cdot 10^5 \frac{\text{N}}{\text{m}} \end{pmatrix} \tag{22}$$

and all initial states $x(t=0)$ set to 0. Noise is applied to the generated simulation results as in

$$\tilde{y} = \bar{y} + \mathcal{N}(0, \Sigma_y) \tag{23}$$

to reflect noisy measurements. In general, Σ_y is a block-diagonal matrix with the (co-)variances of the measurements σ_y^2 at each time point. For simplicity, we assume that the noise of the several measurements is uncorrelated here, such that σ_y only contains the diagonal entries

$$\text{diag}(\sigma_y) = [0.01 \text{ m}, 0.01 \frac{\text{m}}{\text{s}}, 0.01 \text{ m}, 0.01 \frac{\text{m}}{\text{s}}]. \tag{24}$$

As a result, Σ_y is also a diagonal matrix. The inputs \tilde{u} that are sent to the system are

$$\tilde{u}(t) = 0.05 \text{ m} \cdot \sin(2\pi t). \tag{25}$$

These inputs are assumed to be subject to noise, so that the inputs that actually excite the system and therefore are used for simulation are

$$\tilde{u} = \bar{u} + \mathcal{N}(0, \Sigma_w) \tag{26}$$

with Σ_w a diagonal matrix with entries $\sigma_w^2 = 0.005^2 \text{ m}^2$. This input noise is a special kind of process noise. Following these steps, data are generated for $L = 100$ repeated estimations of \hat{p} with the PE routine of CASIOPEIA. To compensate for the noise on \tilde{u} , additional degrees of freedom $w \in \mathbb{R}^{n_w}$ with $n_w = n_u$ are introduced to the PE problem. Since the expected order of magnitude for the values in \hat{p} is assumed to be known, the parameters are scaled accordingly within the estimation. The initial guess for \hat{p} used within the estimation is

$$p_{\text{init}} = \begin{pmatrix} k_{C,\text{init}} \\ c_{C,\text{init}} \\ c_{T,\text{init}} \end{pmatrix} = \begin{pmatrix} 1.0 \cdot 10^3 \frac{\text{Ns}}{\text{m}} \\ 1.0 \cdot 10^4 \frac{\text{N}}{\text{m}} \\ 1.0 \cdot 10^5 \frac{\text{N}}{\text{m}} \end{pmatrix}. \tag{27}$$

The scaled mean values \hat{p}_{mean} of all estimates \hat{p}_i and the corresponding standard deviations are then computed and compared to the result of the last estimation \hat{p}_L and the standard deviations obtained from the evaluation of $\beta_{z_L} \Sigma_{\hat{p}_L}$ in CASIOPEIA. The results are shown in Table 1.

Table 1. Comparison of parameter values and standard deviations, initial experiments.

Parameter	k_C ($10^3 \frac{\text{Ns}}{\text{m}}$)	c_C ($10^4 \frac{\text{N}}{\text{m}}$)	c_T ($10^5 \frac{\text{N}}{\text{m}}$)
$\hat{p}_{\text{mean}} = L^{-1} \sum_{i=1}^L \hat{p}_i$	4.082	3.973	1.616
$\sqrt{L^{-1} \sum_{i=1}^L (\hat{p}_i - \hat{p}_{\text{mean}})^2}$	0.1245	0.1455	0.0264
\hat{p}_L	4.141	3.965	1.650
$\sqrt{\text{diag}(\beta_{z_L} \Sigma_{\hat{p}_L})}$	0.1143	0.1284	0.0254

Table 1 shows that the estimated parameter values lie in the neighborhood of p_{true} , but with comparatively high standard deviation, which is caused by the insufficient system excitation introduced by \tilde{u} . Also note that the standard deviations computed by CASIOPEIA are close to the standard deviations of the repeated experiments.

6.3 Optimum experimental design

Based on the initial input trajectory \bar{u} and using the estimated parameters \hat{p}_L from Section 6.2, optimized inputs \hat{u} are obtained using the OED routine of CASIOPEIA. Upper and lower bounds on the state values and the input values are introduced as follows

$$x_{\text{min}} = - \begin{pmatrix} 0.1 \text{ m} \\ 0.4 \frac{\text{m}}{\text{s}} \\ 0.1 \text{ m} \\ 0.4 \frac{\text{m}}{\text{s}} \end{pmatrix}, \quad x_{\text{max}} = \begin{pmatrix} 0.1 \text{ m} \\ 0.4 \frac{\text{m}}{\text{s}} \\ 0.1 \text{ m} \\ 0.4 \frac{\text{m}}{\text{s}} \end{pmatrix}, \tag{28}$$

$$u_{\text{min}} = -0.05 \text{ m}, \quad u_{\text{max}} = 0.05 \text{ m}, \tag{29}$$

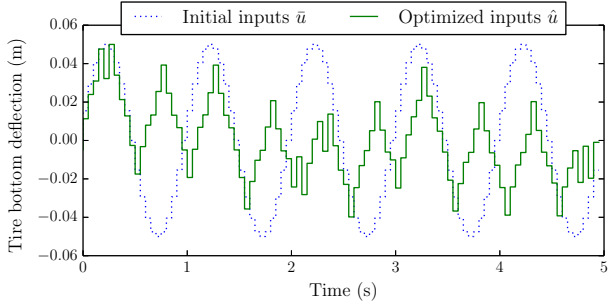


Fig. 2. Comparison of initial and optimized inputs.

in order to generate experiments only within the system's (assumed) operation range. As information function Φ the A-criterion, i. e. the trace of Σ_p , is used.

Figure 2 shows the optimized inputs \hat{u} in comparison to the initial inputs \bar{u} , depicting how the optimized inputs are used to intentionally excite different parts of the system and increase the information content. The increase in confidence achieved by the experimental design can be visualized by a comparison of the confidence ellipsoids that result from the initial and optimized inputs, which is shown in Figure 3 for parameters k_C and c_T .

6.4 Parameter estimation from optimized experiments

Using \hat{u} , optimized measurement data \hat{y} are generated according to Section 6.2 and the estimations are repeated with otherwise unchanged settings. The results are shown in Table 2.

Table 2. Comparison of parameter values and standard deviations, optimized experiments.

Parameter	k_C ($10^3 \frac{Ns}{m}$)	c_C ($10^4 \frac{N}{m}$)	c_T ($10^5 \frac{N}{m}$)
$\hat{p}_{\text{mean}} = L^{-1} \sum_{i=1}^L \hat{p}_i$	4.027	3.985	1.604
$\sqrt{L^{-1} \sum_{i=1}^L (\hat{p}_i - \hat{p}_{\text{mean}})^2}$	0.0592	0.0784	0.0177
\hat{p}_L	3.992	4.013	1.596
$\sqrt{\text{diag}(\beta_{\hat{z}_L} \Sigma_{\hat{p}_L})}$	0.0559	0.0814	0.0172

Table 2 shows that it is now possible to retrieve confident estimates for all parameter values that lie very close to p_{true} . This shows that not only more reliable estimates can be obtained based on the optimized experimental data, but also less experiments would be necessary in order to obtain this level of certainty.

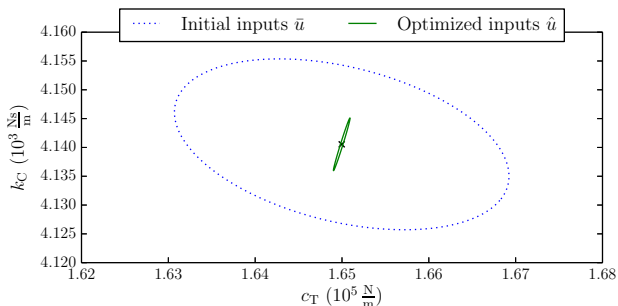


Fig. 3. Comparison of confidence ellipsoid of parameters k_C and c_T for initial and optimized inputs.

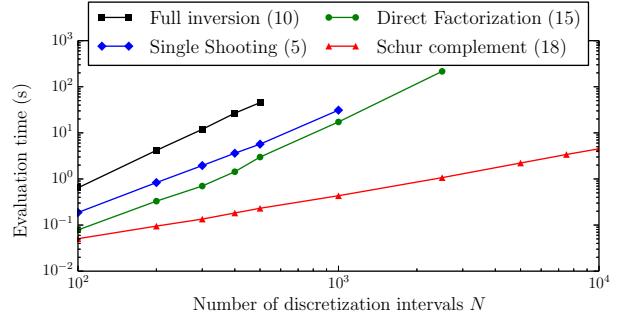


Fig. 4. Comparison of covariance matrix evaluation time on an Intel Core i5-4570 3.20 GHz CPU.

In case further improvement was desired, the design and estimation procedures could be repeated based on the improved estimation results. Furthermore, CASIOPEIA could be used to design multiple experiments within one optimization problem to obtain complementary experiments that deliberately excite different system parts.

6.5 Performance of the Schur complement method

To assess the performance of the method in (18), $\Sigma_{\hat{p}}$ is computed for different time horizons t_f which result in a different number of discretization intervals. The computation times are compared to those of the other approaches shown in the previous sections, i. e., evaluating $\Sigma_{\hat{p}}$ as in (5) for a single shooting implementation of the PE problem according to (1), to the direct evaluation of K^{-1} given in (10) and to the direct factorization approach in (15). To simplify the comparison, a fourth-order Runge-Kutta method with fixed step size is used for discretization of the dynamic systems.

Figure 4 shows that the Schur complement method in (18) is the most efficient approach within this comparison. The evaluation time increases almost linearly with the number of discretization intervals N (which here is equivalent to an increase in measurements, see Section 2.1). The methods (5), (10) and (15) show a much more rapid increase in evaluation time. Missing entries in Figure 4 indicate that the evaluations either did not finish in reasonable time, or the limitations of approximately 14 GB of available memory were exceeded.

7. CONCLUSIONS AND FUTURE WORK

Within this paper, a Schur complement method for OED has been presented that is particularly well suited for PE problems with process noise. The method has been implemented in the open-source software package CASIOPEIA based on the symbolic framework CASADI and tested on a numerical example.

While the advantages over other approaches have been investigated, it is subject to further research how the evaluation time can be further decreased to facilitate OED problems of larger size. Since the Schur complement approach to compute the covariance matrix is rather generic, the method can easily be extended to directly support other types of noise as well.

REFERENCES

- Andersson, J. (2013). *A General-Purpose Software Framework for Dynamic Optimization*. PhD thesis, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, KU Leuven.
- Annergren, M. (2016). *Application-Oriented Input Design and Optimization Methods Involving ADMM*. PhD thesis, KTH Royal Institute of Technology.
- Bauer, I., Bock, H.G., Körkel, S., and Schlöder, J.P. (2000). Numerical methods for optimum experimental design in DAE systems. *Journal of Computational and Applied Mathematics*, 120(1-2), 1–25.
- Bock, H.G. (1983). Recent advances in parameter identification techniques for ODE. In *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, 95–121. Birkhäuser.
- Bock, H.G., Körkel, S., Kostina, E., and Schlöder, J.P. (2007). *Robustness Aspects in Parameter Estimation, Optimal Design of Experiments and Optimal Control*, 117–146. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-540-28396-6_6. URL http://dx.doi.org/10.1007/978-3-540-28396-6_6.
- Bock, H.G., Körkel, S., and Schlöder, J.P. (2013). *Parameter Estimation and Optimum Experimental Design for Differential Equation Models*, 1–30. Springer, Berlin, Heidelberg.
- Bürger, A. (2016). *casiopeia – Casadi Interface for Optimum experimental design and Parameter Estimation and Identification Applications*. <https://github.com/adbuenger/casiopeia>.
- Davis, T.A. (2006). *Direct methods for sparse linear systems*. Fundamentals of algorithms. SIAM, Society for Industrial and Applied Mathematics, Philadelphia.
- Golub, G. and Loan, C. (1996). *Matrix Computations*. Johns Hopkins University Press, Baltimore, 3rd edition.
- Griewank, A. (2000). *Evaluating derivatives : principles and techniques of algorithmic differentiation*. Frontiers in applied mathematics ; 19. SIAM, Philadelphia.
- Körkel, S. (2002). *Numerische Methoden für Optimale Versuchsplanungsprobleme bei nichtlinearen DAE-Modellen*. PhD thesis, Interdisciplinary Center for Scientific Computing, Heidelberg University.
- Körkel, S., Kostina, E., Bock, H.G., and Schlöder, J.P. (2004). Numerical methods for optimal control problems in design of robust optimal experiments for nonlinear dynamic processes. *Optimization Methods and Software*, 19(3-4), 327–338.
- Kostina, E. and Kostyukova, O. (2012). *Computing Covariance Matrices for Constrained Nonlinear Large Scale Parameter Estimation Problems Using Krylov Subspace Methods*, 197–212. Springer Basel, Basel.
- Ljung, L. (1999). *System identification : theory for the user*. Prentice-Hall information and system sciences series. Prentice Hall, Upper Saddle River, NJ [et al.], 2. edition.
- Mitschke, M. (2014). *Dynamik der Kraftfahrzeuge*. Springer Vieweg, Wiesbaden, 5. edition.
- Nocedal, J. and Wright, S.J. (2006). *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition.
- Press, W.H. (ed.) (2007). *Numerical recipes : the art of scientific computing*. Cambridge University Press, Cambridge [et al.], 3. edition.
- Pronzato, L. (2008). Optimal experimental design and some related control problems. *Automatica*, 44, 303–325.
- Rao, C.V., Rawlings, J.B., and Mayne, D.Q. (2003). Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE Transactions on Automatic Control*, 48(2), 246–258.
- Walter, E. and Pronzato, L. (1997). *Identification of parametric models from experimental data*. Communications and control engineering. Springer [et al.], Berlin.