

# Input design and parameter estimation with open source tools<sup>★</sup>

Simon Mayr<sup>\*</sup> Gernot Grabmair<sup>\*\*</sup> Johann Reger<sup>\*\*\*</sup>

<sup>\*</sup> Upper Austria University of Applied Sciences, Stelzhamerstraße 23,  
4600 Wels, Austria (e-mail: simon.mayr@fh-wels.at).

<sup>\*\*</sup> Upper Austria University of Applied Sciences, Stelzhamerstraße 23,  
4600 Wels, Austria (e-mail: gernot.grabmair@fh-wels.at)

<sup>\*\*\*</sup> Control Engineering Group, Technische Universität Ilmenau,  
Helmholtzplatz 5, 98693 Ilmenau, Germany, (e-mail: reger@ieee.org)

---

**Abstract:** In this paper we present free tools for model-based optimal input design and parameter estimation. The discussed tool-chain is tailored for the needs of small- and medium sized companies. Its programming core is based on Scilab and the JModelica platform and features input design (DOE), optimal control problems (OCP), and parameter estimation. Finally, the entire tool-chain concept is tested via simulation of a cart and pendulum system.

*Keywords:* open source, software tools, optimal input design, optimal experiment design, parameter estimation

---

## 1. INTRODUCTION

Model-based design (MBD) is a common and wide-spread approach in various areas of application. Typical areas can be found in automotive and aerospace industry where motion control applications are prevalent. In course of time, MBD processes have received more and more attention in medium-sized and small businesses in order to improve the efficiency of the development processes. However, the pricing of existing commercial software-tools like Matlab/Simulink or Dymola may easily surpass necessary budgets, for small companies in particular. That is the rationale why we initiated to create a free alternative that may help small companies getting started in MBD processes. In Grabmair et al. (2014) an open source tool-chain for model-based design of embedded systems was presented. The focus of the contribution was on code generation. In contrast, to Grabmair et al. (2014), this work extends the free tool-chain by the generation of optimal excitation signals in combination with the solution of optimal control problems (OCP), and parameter estimation in order to efficiently adapt the simulation experiments to real plants.

It is very common that trajectories for mechanical systems are generated by solving a particular optimization problem. Customary approaches include time optimal, energy optimal, and other motion profiles. For decreasing mechanical wear in the plant these profiles need to provide, for example, smooth (rest-to-rest) movement in accordance with restrictions in jerk, acceleration, and velocity. However, smooth trajectories tend to contain insufficient excitation information for identification. To this end, we propose to exploit a measure for the information content concerning some unknown parameters in the optimal control problem.

---

<sup>★</sup> We gratefully acknowledge the support from the Austrian funding agency FFG in Coin-project ProtoFrame (project number 839074).

This paper is organized as follows: After this short introduction the considered class of models and necessary assumptions are noted. Section 3 briefly summarizes how to rephrase a parameter estimation problem as a solution of an optimization problem. In Section 4 the optimization problem for trajectory generation and optimal input design is discussed. In Section 5 the free tool-chain for physics-based modeling, OCP, input design and parameter estimation is presented. Finally, the functionality of the entire tool-chain is assessed and demonstrated on the well-known cart and pendulum model in Section 6. Conclusions are drawn in Section 7.

## 2. PRELIMINARIES

Within our paper we consider dynamical systems of the form<sup>1</sup>

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}) \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{p})\end{aligned}\quad (1)$$

with state vector  $\mathbf{x} \in \mathbb{R}^n$ , output vector  $\mathbf{y} \in \mathbb{R}^{n_y}$ , constant (unknown) parameter vector  $\mathbf{p} \in \mathbb{R}^{n_p}$  and input vector  $\mathbf{u} \in \mathbb{R}^{n_u}$ . These continuous-time systems are considered to be evaluated at various time points  $t_k, 1 \leq k \leq N$ , where  $N$  is the number of measured samples. Furthermore, we assume noise corrupted samples  $y_m(t_k)$  of the measured output

$$\mathbf{y}_m(t_k) = \mathbf{y}(t_k) + \mathbf{w}(t_k), \quad 1 \leq k \leq N \quad (2)$$

where  $\mathbf{y}_m(t_k)$  are the  $k$ -th samples of all measured outputs,  $\mathbf{y}(t_k)$  are the undisturbed outputs at time  $t_k$  and  $\mathbf{w}(t_k)$  is a sequence of uncorrelated white noise which is assumed to be normally distributed with zero mean and variance  $\sigma_i^2, i = 1, \dots, n_y$ . Further, we assume that the variances are known and constant with respect to time. The covariance matrix of the residuals is assumed to be a diagonal matrix with entries  $\sigma_i^2$  on the primary diagonal.

<sup>1</sup> For brevity, we drop time-dependency of input, state, and output.

### 3. PARAMETER ESTIMATION

There are vast literature and numerous approaches for parameter estimation for continuous-time systems, e.g. evaluating certain system responses, Schmidt et al. (2010), specific filtering techniques and standard least square algorithm, see Slotine and Li (1991), modulating functions, see Rao and Unbehauen (2006), to name but a few. Continuous-time system identification can be done by minimizing the cost function

$$J = \frac{1}{t_e - t_0} \int_{t_0}^{t_e} \mathbf{e}^T(t) \mathbf{Q}(t) \mathbf{e}(t) dt \approx \sum_{k=1}^N \mathbf{e}^T(t_k) \mathbf{Q}(t_k) \mathbf{e}(t_k)$$

where  $\mathbf{e}(t) = \mathbf{y}_m(t) - \mathbf{y}(\mathbf{p}, t)$  (3)

with a symmetric, positive definite weighting matrix  $\mathbf{Q}$ . The abbreviation  $\mathbf{y}(\mathbf{p}, t)$  denotes the system output with true parameter vector  $\mathbf{p}$ .

For the moment, we assume a perfect model apart from unknown parameters and an input signal with enough ‘excitation’ or ‘information’ for identification. The cost function becomes zero in the case when the deviation between measurement  $\mathbf{y}_m$  and simulation signal  $\mathbf{y}$  is zero on  $[t_0, t_e]$  (undisturbed case). This is fulfilled when the model is parametrized with the correct parameter values. Minimizing the squared norm of deviation in the disturbed case leads to the well-known optimization problem

$$\min_{\hat{\mathbf{p}} \in \mathbf{P}} \sum_{k=1}^N (\mathbf{y}_m(t_k) - \mathbf{y}(\hat{\mathbf{p}}, t_k))^T (\mathbf{y}_m(t_k) - \mathbf{y}(\hat{\mathbf{p}}, t_k)) \quad (4)$$

with  $\hat{\mathbf{p}}$  as estimation of the nominal parameter vector  $\mathbf{p}$ .

### 4. OPTIMAL CONTROL PROBLEM AND INPUT DESIGN

To ensure an input signal with sufficient ‘information’ for identification we propose to combine the standard optimal control problem with an input design approach. We consider continuous-time systems (1) with outputs evaluated at certain discrete points of time. Each measurement consists of  $N$  samples and the initial guess of unknown parameters  $\hat{\mathbf{p}}_0 \approx \mathbf{p}$  is assumed to be chosen appropriate for the task of input design.

For analyzing the information content of a measurement we resort to the Fisher matrix  $\mathbf{F}(\mathbf{u}, \mathbf{p})$ . This matrix indicates how much information a number of  $n_y$  measurable system outputs contain about  $n_p$  unknown system parameters. This approach is assumed to be independent of the method which is used for identification.

The likelihood for given vectors  $\mathbf{e}(t_k)$  is given by

$$p_f(\mathbf{e}(t_1), \mathbf{e}(t_2), \dots, \mathbf{e}(t_k)) = (2\pi)^{-\frac{n_y N}{2}} \prod_{k=1}^N \frac{1}{\sqrt{\sigma_1^2 \cdot \sigma_2^2 \cdots \sigma_{n_y}^2}} e^{-\frac{1}{2} \sum_{k=1}^N \sum_{i=1}^{n_y} \frac{e_i(t_k)^2}{\sigma_i^2}} \quad (5)$$

The Fisher matrix is defined by Majer (1997)

$$\mathbf{F}(\mathbf{u}, \mathbf{p}) = E \left\{ \frac{\partial \ln p_f(\mathbf{u}, \mathbf{p})}{\partial \mathbf{p}} \bigg|_{\mathbf{p}, t_k} \frac{\partial \ln p_f(\mathbf{u}, \mathbf{p})}{\partial \mathbf{p}} \bigg|_{\mathbf{p}, t_k}^T \right\} \quad (6)$$

in terms of expectation  $E$  and the likelihood  $p_f(\mathbf{u}, \mathbf{p})$  (5). Taking into account that we take expectations and using

the assumption that the residuals have zero mean we may also write

$$\mathbf{F}(\mathbf{u}, \mathbf{p}) = \sum_{k=1}^N \frac{\partial \mathbf{y}}{\partial \mathbf{p}} \bigg|_{\mathbf{p}, t_k}^T \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{n_y}^2 \end{pmatrix}^{-1} \frac{\partial \mathbf{y}}{\partial \mathbf{p}} \bigg|_{\mathbf{p}, t_k}$$

where  $\frac{\partial \mathbf{y}}{\partial \mathbf{p}} \bigg|_{\mathbf{p}, t_k} = \begin{pmatrix} \frac{\partial y_1}{\partial p_1} \bigg|_{\mathbf{p}, t_k} & \cdots & \frac{\partial y_1}{\partial p_{n_p}} \bigg|_{\mathbf{p}, t_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_{n_y}}{\partial p_1} \bigg|_{\mathbf{p}, t_k} & \cdots & \frac{\partial y_{n_y}}{\partial p_{n_p}} \bigg|_{\mathbf{p}, t_k} \end{pmatrix}$  (7)

describes the partial derivatives of the system outputs with respect to some system parameters evaluated at a certain discrete point in time  $t_k$ .

For optimization purposes the above matrix has to be considered in some meaningful metric. For this purpose, various real-valued functions have been suggested as a measure for size of the variance-covariance or the information matrix. The most common are the so-called A-criterion (trace of the matrix), D-criterion (determinant of the matrix) or E-criterion (eigenvalue of the matrix). The interested reader is kindly referred to Franceschini and Macchietto (2008), also for a geometrical interpretation of these criteria.

$$\max_{\mathbf{u}} \|\mathbf{F}(\mathbf{u}, \mathbf{p})\|_{A/D/E\text{-criterion}}$$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p})$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{p})$$

$$\mathbf{x}(t_0) = \mathbf{x}_0$$

$$\mathbf{x}(t_e) = \mathbf{x}_e$$

$$\mathbf{u} \in \mathbf{U}$$

$$\mathbf{x} \in \mathcal{X}$$

$$\dot{\mathbf{S}}_{\mathbf{x}} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p})}{\partial \mathbf{x}} \cdot \mathbf{S}_{\mathbf{x}} + \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p})}{\partial \mathbf{p}}, \quad t > 0$$

$$\mathbf{S}_{\mathbf{x}}(0) = \frac{\partial \mathbf{x}_0}{\partial \mathbf{p}}$$

$$\mathbf{S}_{\mathbf{y}} = \frac{\partial \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{p})}{\partial \mathbf{x}} \cdot \mathbf{S}_{\mathbf{x}} + \frac{\partial \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{p})}{\partial \mathbf{p}}, \quad t \geq 0 \quad (8)$$

The above-stated optimization problem with (forward) state sensitivities  $\mathbf{S}_{\mathbf{x}}$  and output sensitivities  $\mathbf{S}_{\mathbf{y}}$  may in practice be highly nonlinear. Necessarily, we thus shall be satisfied with a suboptimal result.

### 5. THE TOOL-CHAIN

The tool-chain aims to cover plant modelling (*Step 1*), basic model verification (*Step 2*), optimal excitation signal calculation (*Step 3*) and parameter estimation (*Step 4*), see Fig. 1.

*Step 1: System modeling.* System modeling can be either done by implementing equations or by connecting physical blocks in a system modeler. From an engineering point of view it is a nice feature to build mechatronic plants by connecting physical blocks (e.g., mass, spring, damper, ...).

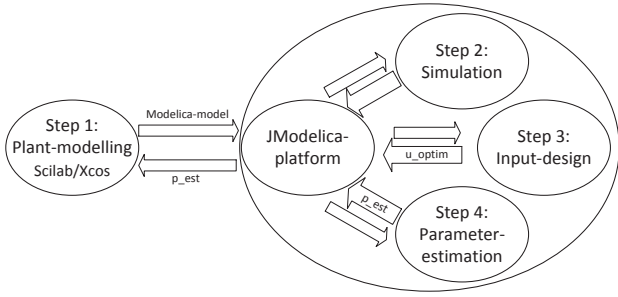


Fig. 1. Tool-chain for MBD process with focus on input design and parameter estimation

*Step 2: Simulation.* Model simulation for principle verification of system behavior.

*Step 3: Input Design.* For an accurate estimation of the parameters an input signal with ‘high information content’ is necessary. Therefore, the nonlinear optimization problem for input design (8) has to be solved. To apply powerful optimization algorithms the system equations are required.

*Step 4: Parameter estimation.* Choose one of the different possibilities to estimate unknown system parameters, e.g. solve the parameter identification optimization problem (4). It may be highly nonlinear optimization problem, however, there is no need to look for a formulation linear in the parameters, what would be very hard to automate for general engineering applications.

For *Step 1* we use the open source software Scilab/XCos. It serves as model editor. The platform for *Step 2–4* is JModelica. It is mainly used to solve the optimization problems (8) and (4).

An open source software that may be used for plant modeling with physical blocks is Scilab/XCos, see Scilab Enterprises (2012). Scilab is a free software for numerical computations similar to MATLAB from Mathworks and is published and developed by Scilab Enterprises. The corresponding dynamic system modeler and simulator is called XCos, a counterpart to Mathwork’s SIMULINK. In Scilab version 5.4.1 the software has accomplished a major development step and may be considered a useful application for laboratory or smaller industrial projects. Systems can be implemented via its differential equations (ODEs) or by referring to the Modelica based XCos extension *Coselica*, see Reusch and Jofret (2013). *Coselica* comprises an implementation of standard Modelica libraries (e.g., electrical, heat transfer, planar mechanical, rotational mechanical, translational mechanical) and therefore, may serve a sufficient number of models for most applications. A Modelica file is generated automatically during the simulation process in Scilab which also grants access to the symbolic system equations as DAE’s. Even if the system is converted into DAE’s in Modelica, we restrict ourselves to systems of the form (1) for easy calculation of the necessary sensitivities, see Remark in Section 6.3.

Solving optimization problems for input design, optimal control, and parameter estimation requires powerful optimization algorithms. Unfortunately so far, optimization algorithms included in Scilab do not use the system equations of XCos/*Coselica* to solve the highly nonlinear optimization problems that occur within input design and

parameter estimation. That is the reason why we use additionally the open source platform JModelica. JModelica is a Modelica-based open source platform for simulation, optimization and analysis of dynamic systems with the focus on optimization of models. It is developed by the Department of Automatic Control, Lund University, see JModelica.org (2014). JModelica supports optimization of dynamic state models. While the system is described in Modelica code, the optimization specifications are described in the Optimica extension. JModelica encapsulates a number of other free software packages. JModelica provides different optimization algorithms. The automatic differentiation based optimization algorithm relying on the CasADi framework is used for solving optimal control problems and parameter estimation problems. The applied CasADi package is an open source software library for symbolic automatic differentiation of functions. For solving the nonlinear program (NLP) the IPOPT (Interior Point OPTimizer) algorithm is used. IPOPT is a state-of-the-art solver for large-scale NLP problems.

Plant modeling (*Step 1*) is done in XCos. The systems are built from *Coselica* blocks. For simulation in XCos a Modelica file of the model is generated, which can be used in JModelica for simulation and optimization. *Step 2* is already done in JModelica. The generated Modelica file is compiled and loaded into JModelica. Simulation options (input, solver properties, simulation time...) can be defined. Then the model is simulated and the results can be plotted for verification. *Step 3* includes input design and is also done in the JModelica environment. For calculation of system sensitivities which are necessary for input design (8) and (7), the sensitivity equations must be known. Up to now this equations have to be added manually.<sup>2</sup>

## 6. EXPERIMENT: CART AND PENDULUM SYSTEM

The well-known cart and pendulum example in gantry crane configuration shall serve as a reference experiment, see Fig. 2. The cart with mass  $m_1$  is driven by a dc-drive with external excitation, see Fig. 3. In the following, we assume a very small electrical time constant ( $\tau_{el} = \frac{L_A}{R_A}$  with armature inductance  $L_A$  and resistance  $R_A$ ) compared to the mechanical one. By means of model reduction, i.e.  $L_A \rightarrow 0$ , we introduce equivalent parameters (10) incorporating the whole drive-train (drive constant  $k_m$ , inertia  $J_G$ , transmission ratio  $n$ , gear pinion radius  $r$ , and some mechanical damping  $d_1$ ) into the mathematical model of the cart. For further parameters refer to Tab. 1.

Table 1. System parameters

Name	Description
$m_2$	mass pendulum
$l$	pendulum length
$d_1$	friction coefficient cart
$d_2$	friction coefficient pendulum
$g$	acceleration due to gravity

The nonlinear system equations for this plant can be computed using the Lagrange formalism, see Grabmair et al. (2014), and leads to

<sup>2</sup> JModelica has also implemented the IDAS solver that can compute sensitivities, see remark in Section 6.3

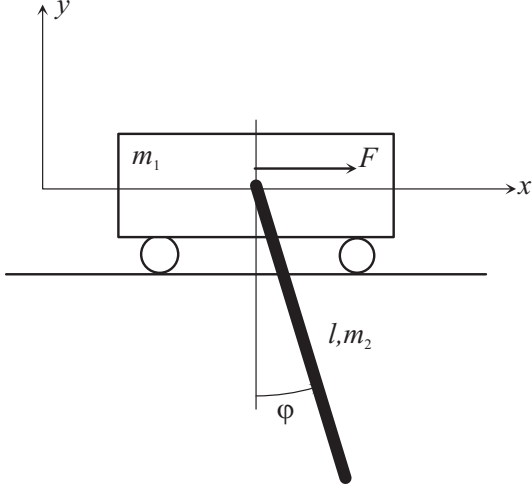


Fig. 2. Cart and pendulum system

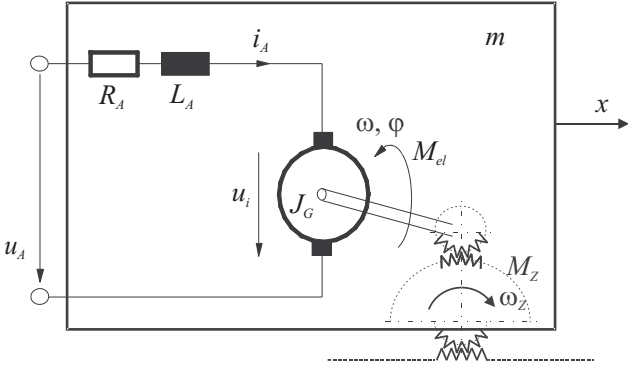


Fig. 3. Armature equivalent circuit and power conversion of the dc-drive with external excitation

$$\begin{pmatrix} \dot{v} \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} \tilde{m}_1 + m_2 & \frac{1}{2}m_2l \cos(\varphi) \\ \frac{1}{2}m_2l \cos(\varphi) & \frac{1}{3}m_2l^2 \end{pmatrix}^{-1} \begin{pmatrix} \frac{1}{2}m_2l \sin(\varphi)\omega^2 - \tilde{d}_1v + F \\ -\frac{1}{2}m_2gl \sin(\varphi) - d_2\omega \end{pmatrix} \quad (9)$$

with system states cart position  $x$ , pendulum angle  $\varphi$  and the corresponding velocities  $(v, \omega)^T$ . Further, we introduce some abbreviations

$$\begin{aligned} \tilde{m}_1 &= m_1 + J_G \left(\frac{n}{r}\right)^2 \\ \tilde{d}_1 &= d_1 + \frac{n^2 k_m^2}{r^2 R_A} \end{aligned} \quad (10)$$

We assume an equivalent cart mass  $\tilde{m}_1 = 2$  kg, an equivalent linear friction coefficient  $\tilde{d}_1 = 15 \frac{\text{Ns}}{\text{m}}$ , a pendulum mass  $m_2 = 1$  kg and a rotational friction coefficient  $d_2 = 0.5 \frac{\text{sNm}}{\text{rad}}$  for all following simulations. Note that the ordinary differential equations (ODEs) from (9) are not used in JModelica. Modelica works with DAEs obtained automatically from XCos. Even if the tool-chain does not require the model from (9), at the moment, we need it to compute the sensitivities  $\mathbf{S}_x$  in an extended model, see (11).

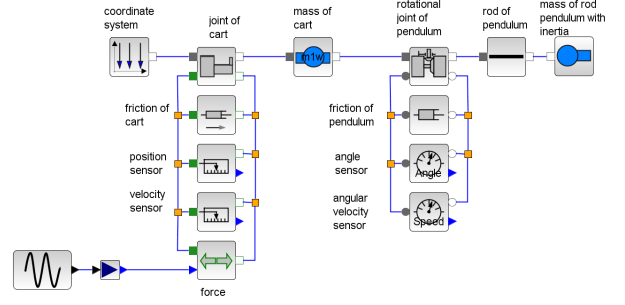


Fig. 4. Cart and pendulum plant built from Coselica blocks

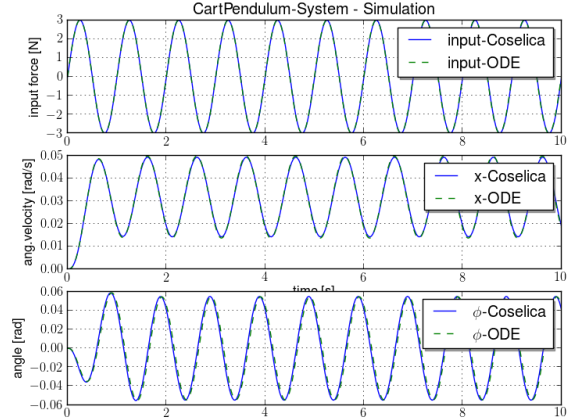


Fig. 5. Cart and pendulum plant simulation in JModelica; compare ODEs and Coselica

### 6.1 Plant modeling in XCos

As mentioned before, the plant is built by connecting blocks from Coselica libraries, see Fig. 4. For the description of the blocks, also see Fig. 4.

### 6.2 Plant simulation in JModelica

When simulating the model in XCos a Modelica file is generated, which is compiled and loaded in JModelica. For tool-chain verification the simulation of the Coselica-model is compared to the signals of the ODE model. As input force we assume a sine wave with amplitude  $A = 3$  N and frequency  $f = 1$  Hz. For the simulation results, see Fig. 5. It is visible that the system behaviors match perfectly.

### 6.3 Optimal input design

The optimization problem for input design (8) shall be specified in more detail for this example. We assume that all system states are zero at time  $t_0 = 0$  s. After the movement at  $t_e = 5$  s the cart is located at position  $x_e$  with pendulum oscillation suppressed. Furthermore there are constraints concerning input signal  $[u_{LO}, u_{HI}]$  and system states  $[\mathbf{x}_{LO}, \mathbf{x}_{HI}]$  as common in engineering tasks, see (12). The pendulum mass is assumed unknown and constant. Position  $x$  and pendulum angle  $\varphi$  are the measured outputs.

*Remark:* At the moment, the sensitivity equations (11) which are used in (8) for input design are added manu-

ally. In later versions the sensitivities will be computed automatically by the IDAS solver implemented in JModelica. The parameter vector including all physical plant parameters is  $\mathbf{p} = (\tilde{m}_1, \tilde{d}_1, m_2, d_2, l)^T$  and the state vector  $\mathbf{x} = (x, \varphi, v, \omega)^T$ . The forward sensitivity differential equation system has the dimension  $n \times n_p$  and can be written as

$$\begin{pmatrix} \dot{S}_{x_1, p_1} & \cdots & \dot{S}_{x_1, p_5} \\ \vdots & \ddots & \vdots \\ \dot{S}_{x_4, p_1} & \cdots & \dot{S}_{x_4, p_5} \end{pmatrix} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \begin{pmatrix} S_{x_1, p_1} & \cdots & S_{x_1, p_5} \\ \vdots & \ddots & \vdots \\ S_{x_4, p_1} & \cdots & S_{x_4, p_5} \end{pmatrix} + \frac{\partial \mathbf{f}}{\partial \mathbf{p}} \quad (11)$$

with  $S_{x_i, p_j} = \frac{\partial x_i}{\partial p_j}$ , the number of system states  $n$  and  $n_p$  the number of system parameters.

For subsequent engineering purposes, all possible parameter sensitivities are already calculated using (11). In this simulation experiment the output sensitivity matrix  $\frac{\partial \mathbf{y}}{\partial \mathbf{p}}$ , see (7), consists of various elements of (11) depending on measurable outputs and parameters of interest. Note that we assume to measure  $x$  and  $\varphi$ , and identify  $m_2$ .

The optimization problem can be phrased as

$$\begin{aligned} \max_{\mathbf{u}} & \|\mathbf{F}(\mathbf{u}, \mathbf{p})\|_{A\text{-criterion}} \\ \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}) \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{p}) \\ \mathbf{x}(t_0) &= \mathbf{0} \\ \mathbf{x}(t_e) &= (0.5 \text{ m}, 0, 0, 0)^T \\ u &\in [-5 \text{ V}, 5 \text{ V}] \\ x &\in [0, 0.5 \text{ m}] \\ \varphi &\in [-0.5 \text{ rad}, 0.5 \text{ rad}] \\ v &\in [0, 0.5 \frac{\text{m}}{\text{s}}] \\ \omega &\in [-0.5 \frac{\text{rad}}{\text{s}}, 0.5 \frac{\text{rad}}{\text{s}}] \\ \dot{\mathbf{S}}_{\mathbf{x}} &= \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p})}{\partial \mathbf{x}} \cdot \mathbf{S}_{\mathbf{x}} + \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p})}{\partial \mathbf{p}}, \quad t > 0 \\ \mathbf{S}_{\mathbf{x}}(0) &= \frac{\partial \mathbf{x}_0}{\partial \mathbf{p}} \\ \mathbf{S}_{\mathbf{y}} &= \frac{\partial \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{p})}{\partial \mathbf{x}} \cdot \mathbf{S}_{\mathbf{x}} + \frac{\partial \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{p})}{\partial \mathbf{p}}, \quad t \geq 0 \end{aligned} \quad (12)$$

with  $t_0 = 0$  and  $t_e = 5 \text{ s}$ .

The Fisher matrix may be detailed as per

$$\begin{aligned} \mathbf{F}(\mathbf{u}, \mathbf{p}) &= \sum_{k=1}^N \frac{\partial \mathbf{y}}{\partial \mathbf{p}} \Big|_{p, t_k}^T \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}^{-1} \frac{\partial \mathbf{y}}{\partial \mathbf{p}} \Big|_{p, t_k} \\ \frac{\partial \mathbf{y}}{\partial \mathbf{p}} \Big|_{p, t_k} &= \begin{pmatrix} \frac{\partial x}{\partial m_2} \\ \frac{\partial \varphi}{\partial m_2} \end{pmatrix} \Big|_{p, t_k}. \end{aligned} \quad (13)$$

We assume  $\sigma_1^2 = \sigma_2^2$ . After adding the sensitivity equations to the auto-generated Modelica model the optimization

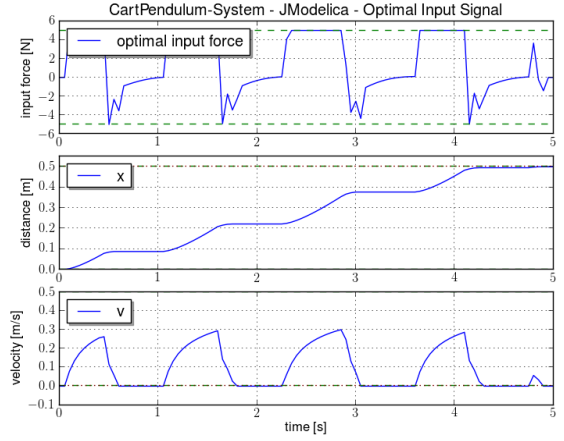


Fig. 6. Cart and pendulum - simulation with optimal input

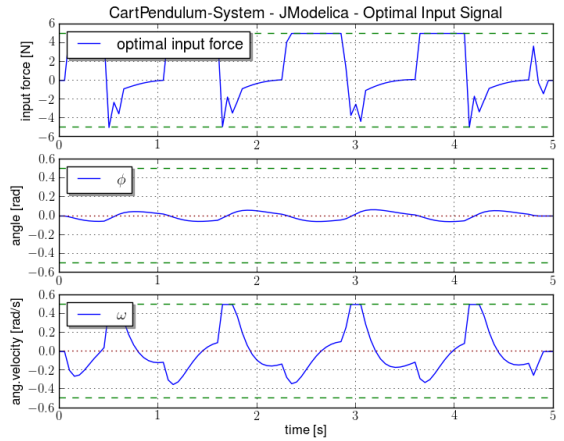


Fig. 7. Cart and pendulum - simulation with optimal input

problem (12) is implemented in JModelica, subsequently the solution may be computed. The dynamic optimization problem (12) is discretized by collocation. The dynamic model variable profiles are approximated by piecewise polynomials. This method of approximation corresponds to a fixed step implicit Runge-Kutta scheme, where  $n_e$  describes the number of steps and  $n_{cp}$  describes the number of collocation points within each step. The resulting nonlinear program (NLP) is solved by the nonlinear IPOPT solver. For more details, the reader is referred to Modelon AB (2014). The optimization result is given in Fig. 6 and Fig. 7, where the number of elements  $n_e = 100$  and the number of collocation points per element  $n_{cp} = 1$  are set as options for the CasADi- and collocation-based optimization algorithm.

In the figures, optimization constraints are represented by green dashed lines and the desired final-states in red dotted-dashed lines. As expected, the car reaches its final position at final-time. Cart speed, angle and angular velocity are forced to zero.

#### 6.4 Parameter identification

The optimization problem (12) is solved under the assumption that the pendulum mass is well-excited with



constraints for our laboratory experiment and therefore, can be identified with high accuracy. For testing the identification of the pendulum mass by simulation, the solution of the optimization problem from (4) is employed. The measurement  $\mathbf{y}_m$  is modeled in the simulation experiment with the true system parameters and additional noise.  $\mathbf{y}(\hat{\mathbf{p}})$  denotes the simulation with estimated parameters.

The identification optimization problem for this particular experiment is given by

$$\min_{\hat{m}_2} \sum_{k=1}^N (x_m(t_k) - x(\hat{m}_2, t_k))^2 + (\varphi_m(t_k) - \varphi(\hat{m}_2, t_k))^2$$

subject to constraint  $\hat{m}_2 \in [0.1 \text{ kg}, 2 \text{ kg}]$

and  $n_e = 500$  and  $n_{cp} = 1$  as options for the CasADi- and collocation-based optimization algorithm.

In Fig. 8 and Fig. 9 a comparison between measurement (simulation with nominal parameters and noise) and simulation signals with the estimated pendulum mass are shown. Neglecting output noise we get an estimated value  $\hat{m}_2 = 1.0 \text{ kg}$ . For pure testing purpose, we choose noisy measurement signals (white noise with zero mean,  $\sigma_1 = 0.01 \text{ m}$  and  $\sigma_2 = 0.01 \text{ rad}$ ) for identification. The value of the identified parameter deteriorated to  $\hat{m}_2 = 1.017 \text{ kg}$ .

## 7. CONCLUSION

We have sketched an open-source tool-chain for input design, optimal control and parameter identification. Starting with the physics-based modeling in XCos we export the entire model to JModelica. JModelica supports simulation and optimization of models which are written in Modelica code. Based on high-performant optimization algorithms (CasADi with IPOPT solver) an information rich input signal is computed. Parameter identification is rephrased in solving an optimization problem. Up to now, the sensitivity equations, necessary for optimal input design, are added manually. In the near future, the JModelica implemented IDAS solver will be used. This solver computes already the sensitivities automatically. So far in our work, these steps lived on the assumption that for the input design the system parameters are known well. Therefore, an iterative process of input design and

parameter identification will be developed for relaxing this assumption.

## ACKNOWLEDGEMENTS

We gratefully acknowledge the support from the Austrian funding agency FFG in Coin-project ProtoFrame (project number 839074).

## REFERENCES

- Franceschini, G. and Macchietto, S. (2008). Model-based design of experiments for parameter precision: State of the art. *Chemical Engineering Science*, 63(19), 4846–4872. doi:10.1016/j.ces.2007.11.034.
- Grabmair, G., Mayr, S., Hochwallner, M., and Aigner, M. (2014). Model based control design - a free tool-chain. In *European Control Conference (ECC)*. Strasbourg, France.
- JModelica.org (2014). *JModelica.org - Open Source Modelica Platform For Modeling, Simulation And Optimization*. JModelica.org. URL <http://www.jmodelica.org>.
- Majer, C.P. (1997). *Parameterschaetzung Versuchsplanung und Trajektorienoptimierung fuer Verfahrenstechnische Prozesse*. Ph.D. thesis, Stuttgart University, Stuttgart.
- Modelon AB (2014). *JModelica.org User Guide - 1.14*. Modelon AB. URL <http://www.jmodelica.org/page/236>.
- Rao, G. and Unbehauen, H. (2006). Identification of continuous-time systems. *IEEE Proceedings Control Theory and Applications*, 153, 185–220.
- Reusch, D. and Jofret, B. (2013). Atoms: Coselica. URL <https://atoms.scilab.org/toolboxes/coselica>.
- Schmidt, A.P., Bitzer, M., Imre, A.W., and Guzzella, L. (2010). Experiment-driven electrochemical modeling and systematic parameterization for a lithium-ion battery cell. *Journal of Power Sources*, 195, 5071–5080.
- Scilab Enterprises (2012). *Scilab: Free and Open Source software for numerical computation*. Scilab Enterprises, Orsay, France. URL <http://www.scilab.org>.
- Slotine, J.J.E. and Li, W. (1991). *Applied Nonlinear Control*. Prentice Hall.

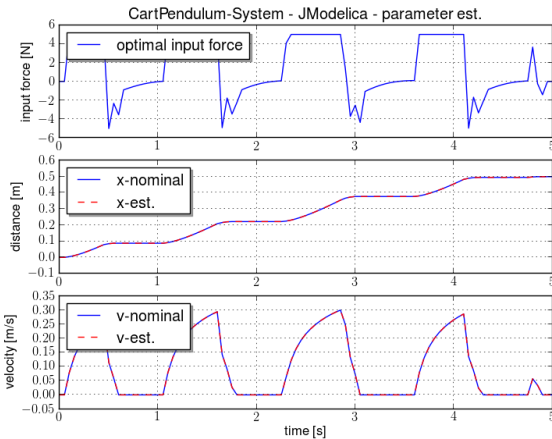


Fig. 8. Cart and pendulum - quality of the parameter estimation

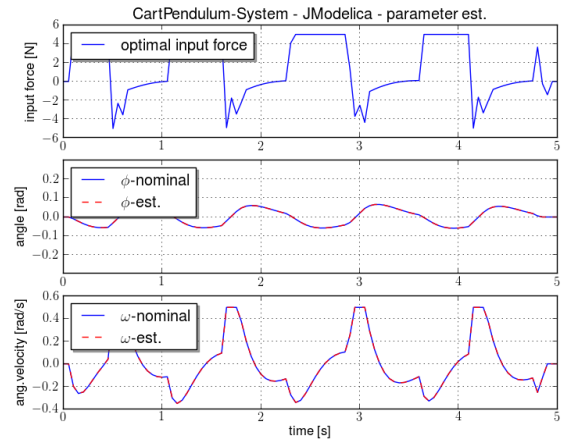


Fig. 9. Cart and pendulum - quality of the parameter estimation