D.5.2.4 - Prototype of MPC toolchain in LMS Imagine.Lab Amesim

SubWP5.2 - Optimization Toolchain

WP5 - Optimized System Operations

MODRIO (11004)

Version	2.0
Date	September 30, 2014

Authors	Stijn De Bruyne,	Siemens Industry Software NV, Leuven, BE[1]
	Jan Anthonis,	Siemens Industry Software NV, Leuven, BE[1]
	Stijn Donders,	Siemens Industry Software NV, Leuven, BE[1]

[1] *formerly named* ***LMS International NV***

Accessibility : PUBLIC

Executive Summary

This document describes the prototype implementation of a toolchain for linear Model Predictive Control (MPC) design within `LMS Imagine.Lab Amesim`. This toolchain enables the usage of Amesim models in the context of predictive control. The implemented problem formulation is presented and some important design decisions are outlined. The usage of the prototype tool is demonstrated by means of a electro-mechanical servomechanism application.

Contents

1 Motivation

A major challenge in mechatronic system design is evaluating the potential performance of a proposed system concept [2, 10]. By definition, the performance of a mechatronic system relies on both the multi-physics hardware and the control software, as illustrated in Figure 1. Unfortunately, designing a control algorithm that handles multiple, often conflicting, objectives on a multi-input, multi-output (MIMO) system is not a straightforward task, which commonly results in a long and resource-intensive design process. Furthermore, mechatronic systems are inherently characterized by fundamental performance limitations, such as actuator constraints and safety limitations, which most often lead to conservative, and thus suboptimal, controller designs. As a consequence, engineers face the difficulty of having to differentiate between a deficient hardware design or an unsatisfactory controller, resulting in a real bottleneck in the mechatronic product development.

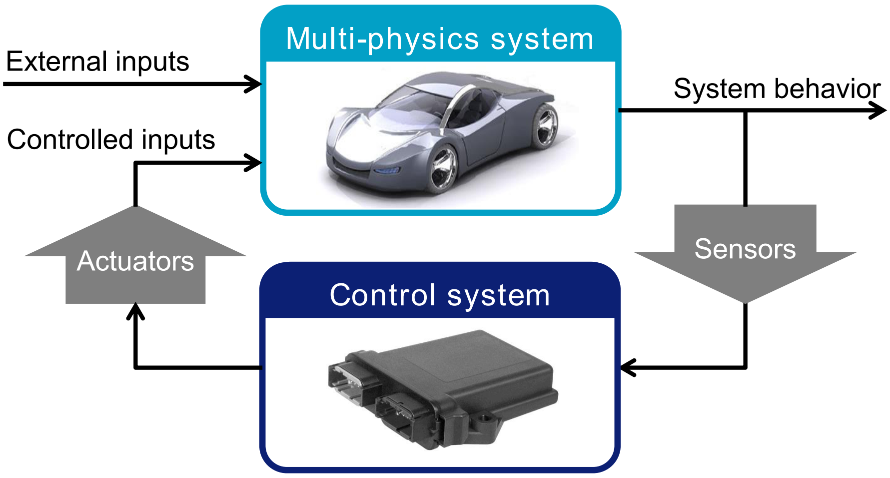

Figure 1: *A mechatronic system extends a multi-physics systems with sensors, actuators and a control system to create an intelligent system*

Model-based control techniques, such as Model Predictive Control (MPC) [8], offer the possibility to create a more systematic control design procedure. They explicitly incorporate a system model into the control algorithm. Moreover, they facilitate the translation of system requirements into control objectives and can even integrate system constraints into the control problem. Finally, by combining system models and numerical optimization algorithms, the performance of a mechatronic system can be assessed more accurately and earlier in the design cycle.

This deliverable presents a prototype implementation of a toolchain for linear MPC design within `LMS Imagine.Lab Amesim`. The overall workflow for designing an MPC controller is outlined Section 2. Section 3 presents the problem formulation that is currently implemented in the prototype, while Section 4 outlines some details on the actual implementation. Finaly, Section 5 will demonstrate the usage of the prototype on a servomechanism application.

2 MPC Design Workflow

The real challenge in MPC control design consists in developing a prediction model (or *control model*) that describes the main system behavior sufficiently well, while being still suitable for online, dynamic optimization [3]. It is clear that the selected prediction model has a critical influence on the resulting performance of the MPC control algorithm. However, in the majority of engineering applications, it is non-trivial to decide which modeling formalism is best suited for solving the task with minimum computational and engineering effort. Consequently, it is common engineering practice to iteratively refine the prediction model, the objective function and the employed numerical solution methods. This workflow for designing an MPC control algorithm is illustrated in Figure 2:

Model analysis and reduction: Typically, detailed design models for the considered system are available. The control engineer should analyze the model, identify the main dynamics and reduce the complexity and structure of the model to a level that makes the control model suited for dynamic optimization.

Control design: Based on the characteristics of the prediction model, the control engineer can select an appropriate MPC algorithm and numerical solution strategy. Preferably, a high-level description is used to capture the control objective, the system constraints and the employed prediction model, whereas the actual implementation of the numerical algorithm should build on automation technologies.

Closed-loop performance analysis: Finally, the developed MPC control algorithm can be evaluated by coupling it to the original, detailed system model. Closed-loop performance, stability and robustness can be evaluated in a normal simulation analysis, while the computational efficiency should be validated by using real-time simulation technology.

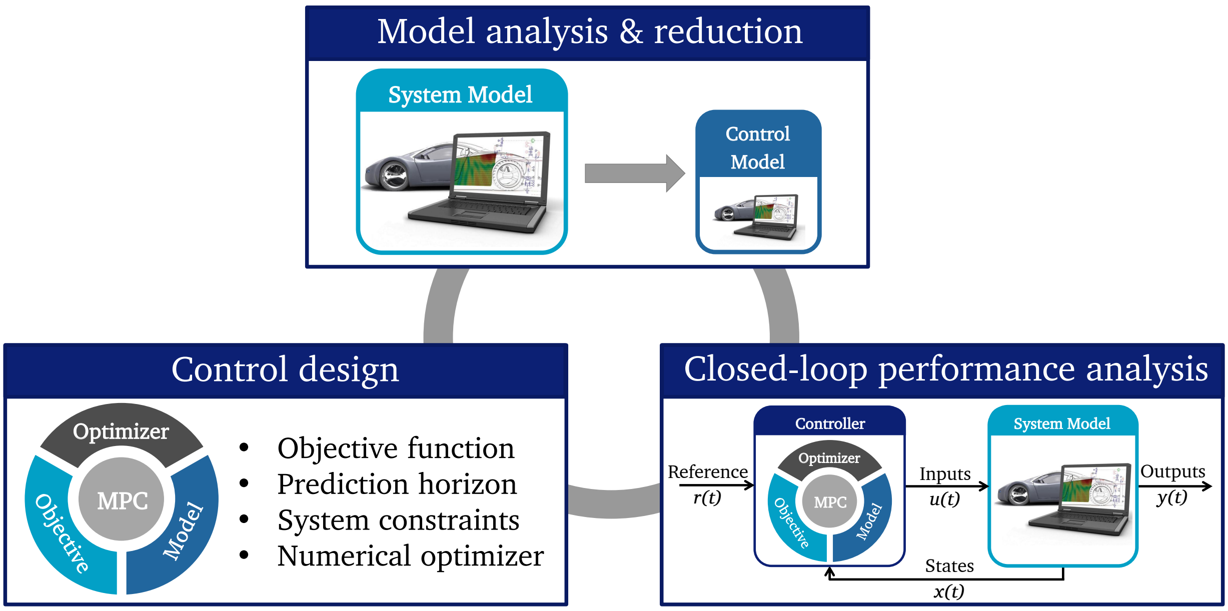

Figure 2: *Workflow for designing and evaluating an MPC control algorithm.*

The developed prototype enables the design of a linear MPC algorithm within the `LMS Imagine.Lab Amesim` environment [9]. It follows the same steps as shown in Figure 2:

- Develop a predictive control model
- Specify the control problem, i.e. prediction horizon, system constraints, objective weights, etc.
- Analyze the closed-loop system performance

3 Linear Model Predictive Control

A linear MPC problem consists of linear system dynamics for the prediction model, linear constraints and a convex quadratic objective function [7, 8]. In this case, the resulting Optimal Control Problem (OCP) can be formulated as a static Quadratic Program (QP), for which efficient solvers exist. This section discusses the control problem formulation under consideration and outlines the design decisions that were made for the prototype implementation.

3.1 Problem Formulation

In many applications, a control engineer is interested in making the system output $y(\cdot)$ track a given reference signal $r(\cdot)$, while keeping the inputs $u(\cdot)$ close to a reference value $u^{\text{ref}}(\cdot)$. Expressed in discrete-time, the resulting MPC problem can be formulated as (2), where:

- A, B, C, D determine the discrete-time linear dynamics at a sampling time t_s,
- r_k, u_k^{ref} and x_N^{ref} are the reference signals,
- N is the prediction horizon (in samples),
- P, Q, R, R_Δ are the weighting matrices of the objective function,
- $\mathbb{X}, \mathbb{X}_f, \mathbb{U}, \mathbb{U}_\Delta, \mathbb{Y}$ are polytopic sets, i.e. sets defined by affine constraints ($\mathbb{X} = x \in \mathbb{R}^{n_x} : F_x x \leq f_x$).

The current prototype only allows to formulate box constraints:

$$\begin{aligned}
u_{\min} \leq u_k \leq u_{\max}, &\qquad \forall k \in \{0, 1, \dots, N-1\} \\
\Delta u_{\min} \leq \Delta u_k \leq \Delta u_{\max}, &\qquad \forall k \in \{0, 1, \dots, N-1\} \\
y_{\min} \leq y_k \leq y_{\max}, &\qquad \forall k \in \{0, 1, \dots, N-1\} \\
x_{\min} \leq x_k \leq x_{\max}, &\qquad \forall k \in \{0, 1, \dots, N\}.
\end{aligned} \tag{1}$$

Linear Tracking MPC

$$
\begin{aligned}
\underset{\substack{\{x_0,\ldots,x_N\},\{y_0,\ldots,y_{N-1}\}\\ \{u_{-1},..,u_{N-1}\},\{\Delta u_0,..,\Delta u_{N-1}\}}}{\text{minimize}} \quad V = & \sum_{k=0}^{N-1}\left(||y_k - r_k)||_Q^2 + ||u_k - u_k^{ref}||_R^2 + ||\Delta u_k||_{R_\Delta}^2\right) \\
& + ||x_N - x_N^{ref}||_P^2 && \text{(2a)}\\
\text{subject to} & \\
x_0 &= x(t) && \textit{Initial State} \quad \text{(2b)}\\
u_{-1} &= u(t - t_s) && \textit{Previous Input} \quad \text{(2c)}\\
x_{k+1} &= Ax_k + Bu_k\ , \forall k \in \{0, 1, \ldots, N-1\} && \textit{Linear Dynamics} \quad \text{(2d)}\\
y_k &= Cx_k + Du_k\ , \forall k \in \{0, 1, \ldots, N-1\} && \textit{Output Equation} \quad \text{(2e)}\\
u_k &= u_{k-1} + \Delta u_k\ , \forall k \in \{0, 1, \ldots, N-1\} && \textit{Control Increment} \quad \text{(2f)}\\
x_k &\in \mathbb{X} \quad , \forall k \in \{0, 1, \ldots, N-1\} && \textit{State Constraints} \quad \text{(2g)}\\
u_k &\in \mathbb{U} \quad , \forall k \in \{0, 1, \ldots, N-1\} && \textit{Input Constraints} \quad \text{(2h)}\\
\Delta u_k &\in \mathbb{U}_\Delta\ , \forall k \in \{0, 1, \ldots, N-1\} && \textit{Input Rate Constr.} \quad \text{(2i)}\\
y_k &\in \mathbb{Y} \quad , \forall k \in \{0, 1, \ldots, N-1\} && \textit{Output Constraints} \quad \text{(2j)}\\
x_N &\in \mathbb{X}_f && \textit{Terminal Constraint} \quad \text{(2k)}
\end{aligned}
$$

In order to solve this optimization problem, this MPC problem is reformulated as a Quadratic Program (QP).

3.2 Equivalent QP Formulation

The standard form of a QP is as follows:

Quadratic Problem

$$\underset{w \in \mathbb{R}^{n_w}}{\text{minimize}} \quad \frac{1}{2} w^T H w + g^T w \tag{3a}$$

$$\text{subject to} \quad Gw \leq b, \tag{3b}$$

$$G_{eq} w = b_{eq} \tag{3c}$$

Two main paradigms exist for reformulating the original MPC tracking problem (2) into the standard QP form (3).

- The **sparse reformulation** expresses all difference and constraint equations as (in)equality constraints and considers all intermediate states and outputs as optimization variables, together with the input actions:

$$w = [y_0, u_0, \Delta u_0, x_1, y_1, u_1, \Delta u_1, x_2, ..., y_{N-1}, u_{N-1}, \Delta u_{N-1}, x_N]^T \tag{4}$$

  This leads to a large, but structured and sparse QP problem. [4, 11]

- In an alternative reformulation, all intermediate states and outputs are explicitly expressed as a function of the initial state x_0 and the sequence of control inputs. This approach is often referred to as **condensing**. [1, 5]

$$u_k = u_{-1} + \sum_{i=0}^{k} \Delta u_i, \tag{5a}$$

$$x_k = A^k x_0 + \sum_{i=0}^{k-1} \Big(A^i B u_{k-1-i} \Big), \tag{5b}$$

$$y_k = C\Big(A^k x_0 + \sum_{i=0}^{k-1} (A^i B u_{k-1-i})\Big) + D u_k; \tag{5c}$$

  It reduces the number of optimization variables significantly:

$$w = [\Delta u_0, \Delta u_1, \ldots, \Delta u_{N-1}]^T \tag{6}$$

  resulting in a small, but dense QP problem.

The current prototype implements a ***condensing strategy***.

4 Implementation Details

4.1 System Linearization

`LMS Imagine.Lab Amesim` offers a Linear Analysis simulation mode, that allows defining the system inputs, states and outputs. The corresponding analysis results in the continuous-time representation of the linearized dynamics. For computing the discrete-time equivalent, the current MPC prototype uses the GNU Scientific Library (GSL) [6] .

4.2 QP Solution

The MPC-QP problem (3) is solved on-line, using an iterative optimization algorithm. Dedicated QP-solvers have been developed to exploit the specific structure that characterizes MPC problems. One solver that is particularly suited for solving condensed MPC problems is `qpOASES` [5]. The link between `LMS Imagine.Lab Amesim` and `qpOASES` is realized through a DLL interface.

5 Demo Example: Servomechanism

To demonstrate the usage of this prototype, it will be employed to design a control algorithm for a electro-mechanical servomechanism, illustrated in Figure 3. It consists of a DC-motor coupled through a gearbox and a flexible shaft to a mechanical load. Table 1 lists the main model parameters.

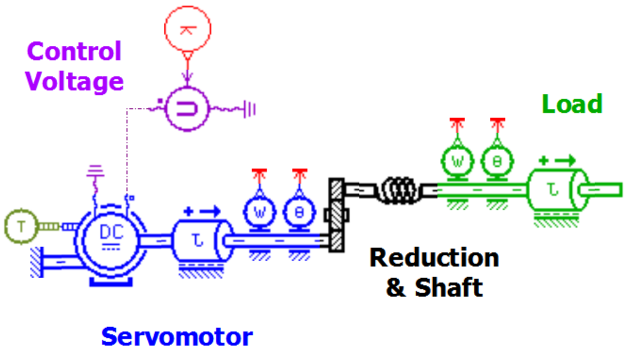

Figure 3: *Servomechanism model in LMS Imagine.Lab Amesim*

Parameter	Value	Unit
Motor inertia	0.5	kgm^2
Load inertia	10	kgm^2
Reduction ratio	20	$-$
Motor torque constant	10	$\mathrm{V \cdot s/rad}$
Armature resistance	20	Ω
Shaft stiffness	1280	$\mathrm{Nm/rad}$
Motor viscous friction	0.1	$\mathrm{Nm \cdot s/rad}$
Load viscous friction	25	$\mathrm{Nm \cdot s/rad}$

Table 1: *Servomechanism model parameters*

5.1 Prediction Model

In a first step, the model is linearized with the motor voltage as input signal, and the shaft torque and load angle as output signals.

5.2 Control Problem

The following system constraints hold:

- Shaft torque is limited to 40 Nm
- Control voltage is limited to 200 V

The prediction model is discretized at $t_s = 0.1s$ and the output tracking weights are chosen as follows:

$$Q = \begin{bmatrix} 0.1 & 0 \\ 0 & 30 \end{bmatrix} \tag{7}$$

Figure 4 shows the closed-loop system and illustrates how these problem specifications can be set in the prototype MPC block within `LMS Imagine.Lab Amesim`.

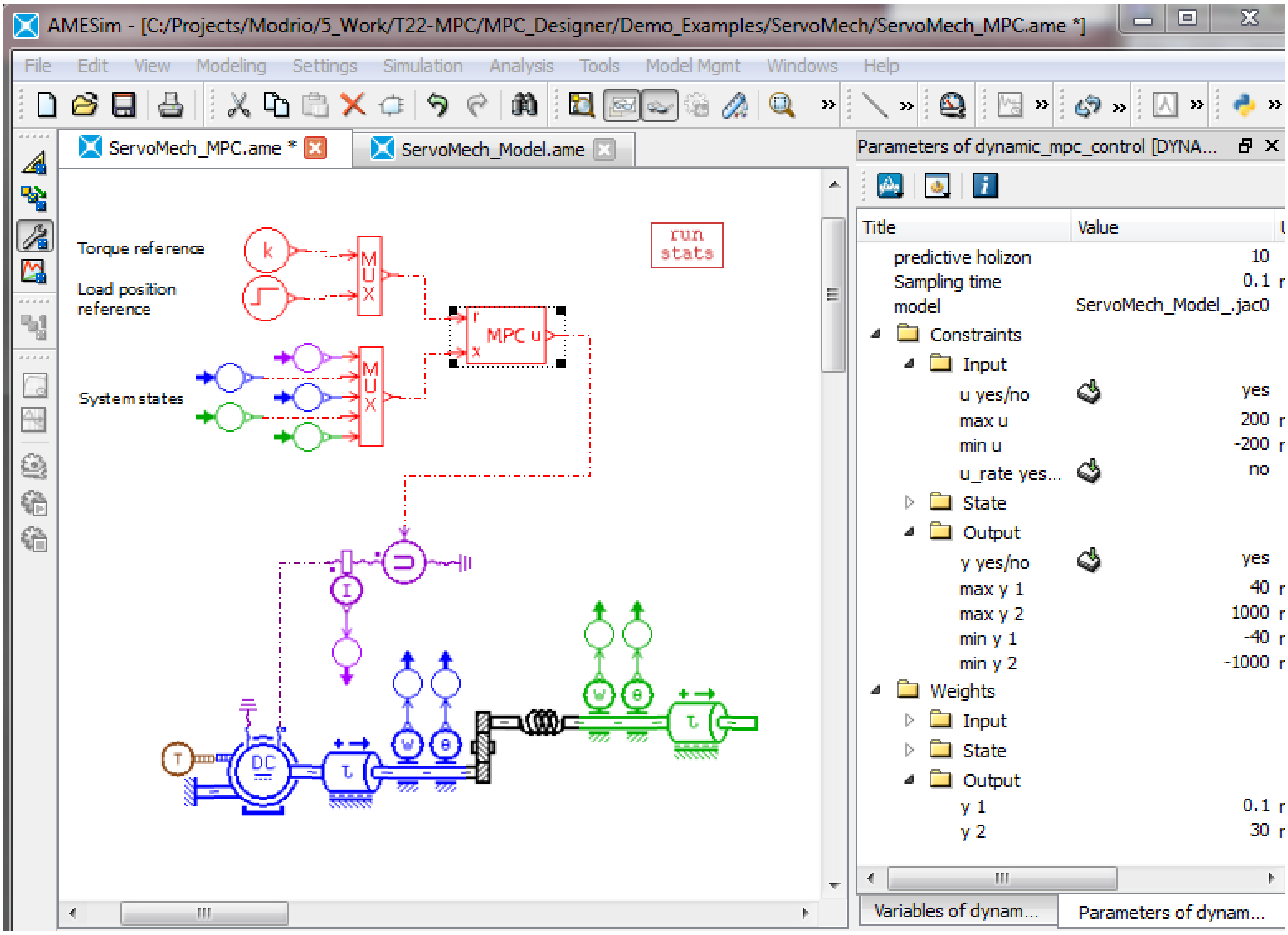

Figure 4: *MPC Control Design in LMS Imagine.Lab Amesim*

Figure 5 shows the closed-loop system reaction to a 30° step change in the reference load angle. Both the constraint on the input voltage and the torque through the shaft are satisfied, and the load is quickly brought to the new reference position. Thanks to the efficient implementation of the QP reformulation and the `qpOASES` solver, this 3s-simulation was computed in 750ms.

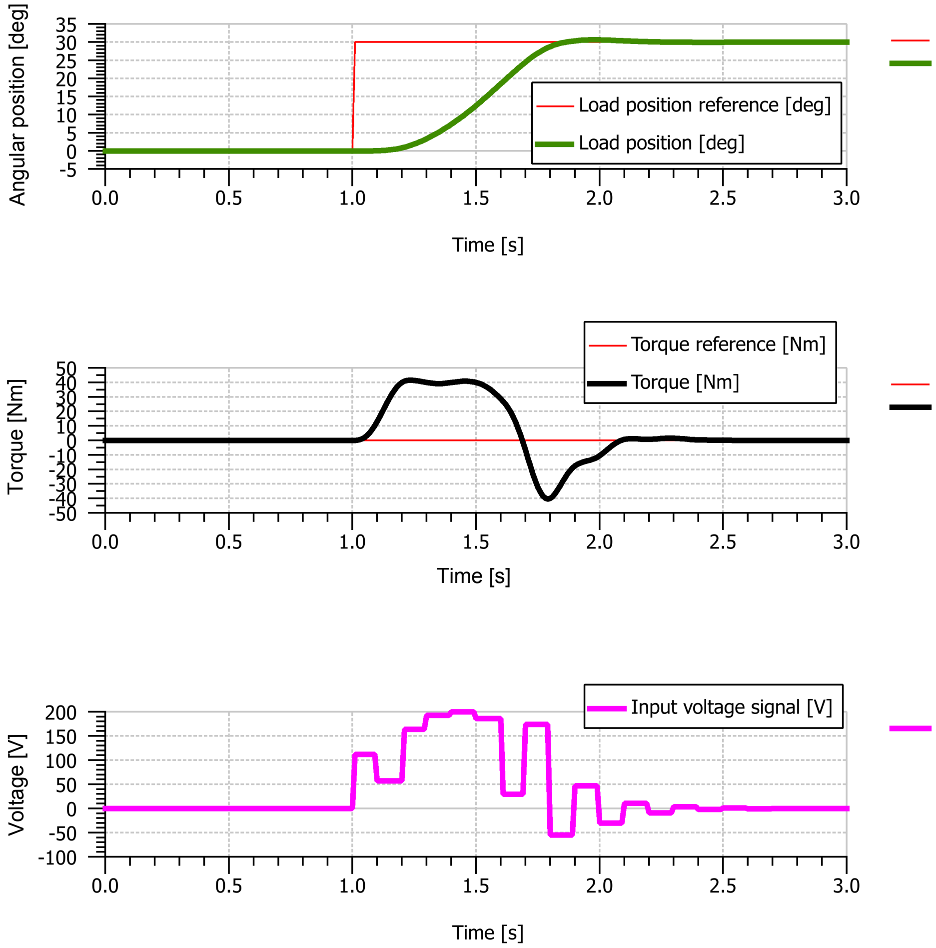

Figure 5: *Closed-loop response to step change in reference position.*

6 Conclusion

This document presented a prototype toolchain for the development and analysis of a linear Model Predictive Control algorithm within the `LMS Imagine.Lab Amesim` modeling environment. A prediction model can be derived using the system analysis tools that are available in the software package. This prediction model can then be integrated in a linear MPC formulation for reference tracking problems. The effectiveness of the prototype tool is demonstrated in an electro-mechanical servomechanism application.

References

[1] Bock H.G. and Plitt K.J. *A multiple shooting algorithm for direct solution of optimal control problems*. In *Proceedings of the 9th IFAC World Congress (IFAC 1984)*, 243—247. Budapest, Hungary (1984).

[2] Coelingh E., de Vries T. and Koster R. *Assessment of mechatronic system performance at an early design stage*. IEEE/ASME Transactions on Mechatronics, 7(3):269—279 (2002).

[3] Del Re L. *Automotive Model Predictive Control: Models, Methods and Applications*. Springer (2010).

[4] Domahidi A., Zgraggen A.U., Zeilinger M.N., Morari M. and Jones C.N. *Efficient interior point methods for multistage problems arising in receding horizon control*. In *Proceedings of the 2012 IEEE 51st Annual Conference on Decision and Control (CDC 2012)*, 668—674. Maui, HI, USA (2012).

[5] Ferreau H.J. *Online active set strategy for fast solution of parametric QP with application on predictive engine control*. Master thesis, KU Leuven (2006).

[6] Gough B. *GNU Scientific Library Reference Manual - Third Edition*. Network Theory Ltd. (2009).

[7] Maciejowski J. *Predictive Control with Constraints*. Prentice Hall, 1st edition (2000).

[8] Rawlings J.B. and Mayne D.Q. *Model Predictive Control Theory and Design*. Nob Hill Pub. (2009).

[9] Siemens Industry Software NV. *LMS Imagine.Lab Amesim* (2013). http://www.plm.automation.siemens.com/en_us/products/lms/imagine-lab/amesim/.

[10] Tate E.D., Sasena M., Gohl J. and Tiller M. *Model embedded control: A method to rapidly synthesize control-lers in a modeling environment*. In *Proceedings of the 6th International Modelica Conference*, 493—502. Modelica Association, Bielefeld, Germany (2008).

[11] Wang Y. and Boyd S. *Fast model predictive control using online optimization*. IEEE Transactions on Control Systems Technology, 18(2):267—278 (2010).