MODELLING AND MODEL PREDICTIVE CONTROL OF MAGNETIC
LEVITATION LABORATORY PLANT

Petr Chalupa
Jakub Novák
Martin Malý
Faculty of Applied Informatics
Tomas Bata University in Zlin
nam. T. G. Masaryka 5555, 760 01, Czech Republic
E-mail: chalupa@fai.utb.cz

KEYWORDS
State-space model, model predictive control, MATLAB, Simulink, magnetic levitation, CE152 model

ABSTRACT
The paper is focused on creating a mathematical model of a magnetic levitation plant and usage of the model for a design of a predictive controller. The magnetic levitation laboratory plant CE 152 by Humusoft Company is used to determine values of model parameters and for real time control experiments. From the control point of view, the CE152 represents a nonlinear and very fast system. Both the mathematical model and the model predictive controller are created using MATLAB / Simulink environment. This environment extended by Real time toolbox is used for real time experiments with the laboratory plant.

INTRODUCTION
Design techniques of current controllers are in most cases based on some kind of model of the real-time plant to be controlled (Bobál et al. 2005). There are many types of mathematical models of the real-time plants but linear models is the most popular category for the control design. More sophisticated nonlinear models can represent the plant in more details than linear models but the controller design is much more complicated in most cases. Moreover, controllers based on nonlinear models are in general more computationally demanding and thus are not suitable for fast systems.

This paper is based on previous work (Chalupa et al. 2016) where detailed nonlinear Simulink model of the CE152 Magnetic Levitation plant (Humusoft 1996) is presented. The model was derived using first principle modelling (Himmelblau and Riggs 2004) and values of the model parameters were specified by real-time measurements. This approach is often referred to as “grey-box modelling” (Tan and Li 2002). The final Simulink model is used to linearize the model around a suitable operating point in the proposed paper (Ljung 1999).

The linear model is used to design a model predictive controller (MPC) (Camacho and Bordons, 2004). The state space linear model is used and the receding horizon principle is applied to the process of minimization of the MPC criterion (Kwon and Han, 2005).

The main goal of the work presented in this paper was to design and verify a model predictive controller for the CE152 Magnetic levitation system. There are several challenges concerning the design process: instability of the controlled plant, nonlinearity of the controlled plant and very fast response of the plant.

CE152 MAGNETIC LEVITATION PLANT
A photo of the CE 152 magnetic levitation plant is presented in Figure 1.

The CE152 Magnetic levitation system is a nonlinear unstable dynamic system with one input and one output. The input signal affects the ball position while the output signal gives information about position of a steel ball. Both signal values are converted and scaled to the specific range of the machine unit [MU].

Structure of the CE152 Magnetic levitation system
The system consists of a model of the magnetic levitation system, power supply and a universal data acquisition card MF624. MF624 is a standard PCI card with A/D, D/A converters, analogue/digital inputs and outputs,
counters, timers and appropriate drivers. The model is connected to the PC via this card.

Figure 2: Scheme of the magnetic levitation plant

Simplified inner structure is shown in Figure 2. A steel ball levitates in magnetic field of the coil driven by power amplifier connected to D/A converter. Position of the steel ball is measured by inductive linear position sensor connected to A/D converter. Both control and measured parameters are sent and received by Simulink.

System behaviour

The CE152 Magnetic levitation system is a nonlinear unstable single input single output dynamic system. When an input control signal of certain value is sent to the system, the ball is lifted upwards to the magnetic core and it stops when it hits the core. This behaviour is caused by electromagnetic force of the magnetic core, which overcame the force of gravity. As the ball getting closer to the coil core, accelerating force grows. Because of an obstacle in the form of magnetic core, both the ball and accelerating force stops. Higher input signal means higher electromagnetic force and much more rapidly increasing acceleration. If control value decreases under a certain value, electromagnetic force is not big enough to overcome gravitational force and the ball falls down.

MODELLING OF THE CE 152 PLANT

The process of creating a mathematical model of the CE152 Magnetic Levitation laboratory plant is described in detail in (Chalupa et al. 2016). This chapter presents just results important for a design of a control system. The magnetic levitation plant can be divided into several parts and each of these parts can be modelled separately:

- D/A converter,
- power amplifier,
- ball and coil subsystem,
- position sensor,
- A/D converter.

D/A converter

The D/A converter converts digital signal \( u_{MU} \) from PC into an analogue voltage signal \( u \). The D/A converter can be described by a linear function (1):

\[
u = k_{DA} u_{MU}
\]

where \( u \) is D/A converter output signal/coil input voltage [V], \( u_{MU} \) is D/A converter input signal [MU] and \( k_{DA} \) is D/A converter gain [V/MU].

Power amplifier

The power amplifier represents a source of constant current, which is proportional to its control voltage:

\[
i = k_{i} u
\]

where \( k_{i} \) is a gain of the power amplifier.

Ball and coil subsystem

Lagrange’s method can be used for modelling ball and coil subsystem. The motion equation is based on the balance of all acting forces.

\[
m_{k} \ddot{x} + k_{fc} \dot{x} = \frac{i^2 k_{c}}{(x-x_{0})^2} - m_{k} g
\]

where:
- \( g \) - gravitational acceleration [m.s\(^{-2}\)]
- \( x \) - ball position [m]
- \( m_{k} \) - ball mass [kg]
- \( x_{0} \) - coil offset [m]
- \( k_{c} \) - coil constant [-]
- \( i \) - coil current [A]
- \( k_{fc} \) - dumping constant [N.s.m\(^{-1}\)]

Position sensor

An inductive position sensor is used to measure the ball position \( x \). The relation between ball position and voltage is approximately linear.

\[
y = k_{x} x + y_{0}
\]

where \( x \) is ball position [m], \( y_{0} \) position sensor offset [V], \( y \) position sensor voltage [V], \( k_{x} \) - position sensor gain [V/m]

A/D converter

The A/D converter converts analogue voltage signal \( y \) into a digital signal \( y_{MU} \). The A/D converter can be described by a linear function (5):

\[
y_{MU} = k_{AD} y
\]

where \( y \) represents A/D converter output signal/position sensor voltage [V], \( y_{MU} \) is A/D converter output signal [MU] and \( k_{AD} \) is A/D converter gain [MU/V].

Model of the whole system

The model of the whole system consists of joined models of individual parts – equations (1) – (5). A Simulink scheme of the whole model is presented in Figure 3.
Determination of matrices parameters is done based on selected operating (nominal) point. Linearization of following differential equations at parameters of these matrices are obtained through described as presented in Figure 4.

\[\frac{d\dot{x}}{dt} = f(x, u_{MU})\]  
\[y = g(x, u_{MU})\]

Where \(X\) is a state vector, \(u\) is an input signal and \(y\) represents the output signal. In case of CE 152 the state vector consists of two scalars:

\[X = [x, v]^T\]

The system described by equations (6) – (8) can be linearized around some operating point and system matrix (A), input matrix (B), output matrix (C) and feedthrough matrix (D) are calculated. Parameters of these matrices are obtained through linearization of following differential equations at selected operating (nominal) point.

\[\frac{dv}{dt} = \frac{(k_{DA}k_yu_{MU})^2k_c}{m_k(x-x_o)^2} - \frac{k_{fy}}{m_k}v - g\]  
\[\frac{dx}{dt} = v\]  
\[y_{MU} = k_{AD}k_x + k_{AD}y_o\]

Now final shape of the matrices is:

\[A = \frac{\partial f}{\partial x}(x, u_{MU})\]  
\[B = \frac{\partial f}{\partial u}(x, u_{MU})\]  
\[C = \frac{\partial g}{\partial x}(x, u_{MU})\]  
\[D = \frac{\partial g}{\partial u}(x, u_{MU})\]

Steady state is assumed and then the derivatives in equations (9), (10) are set to be zero, which leads to

\[v_{ss} = 0\]

\[u_{MUSs} = \pm \sqrt{\frac{m_kg(x-x_o)^2}{k_c}}\]

State-space representation can be converted to transfer function form. For this case of a SISO model, the resulting transfer function is only one in the form:

\[G(s) = \frac{b_0}{s^2+a_1s+a_0}\]

where partial coefficients are:

\[b_0 = -\frac{2k_{DA}k_yk_xk_c}{\sqrt{m_k(x_{ss}-x_o)}}\]

\[a_0 = -\frac{2g}{x_{ss}-x_o}\]

\[a_1 = \frac{k_{fy}}{m_k}\]

\[a_2 = 1\]

The transfer function gain is a static variable and it doesn’t depend on the chosen operating point.

**Selection of an operating point**

In general, not all operating points are necessarily suitable for linearization. It is a question, how adequate is the linearized approximation in comparison with the real system dynamics. First step is to specify value of the operating point parameters, in this case only one parameter is needed and this is to specify a position of the ball \(x_{ss}\). The chosen operating point corresponds to the position of a ball approximately in the middle of the space between the magnetic core and the head of the inductive sensor. Then the result value is:

\[x_{ss} = -\frac{1}{2} = 2.85 \cdot 10^{-3}[m]\]
Verification and validation of the linearized model

It is assumed that nonlinear model is sufficiently accurate and so given linearized model should be too, but it must be considered, that linear model will be accurate only, when the system state is near the operating point.

DISCRETIZATION OF THE LINEAR MODEL

The continuous-time linear model presented in previous chapter is not suitable for controller design because many MPC algorithms are based on discrete-time model of the controlled system. Therefore, continuous-time model was transformed to its discrete-time version. Sampling period $T_0=0.001$ s was used for all experiments. It is assumed, that discrete-time state space system looks as follows:

$$x_{k+1} = \Phi x_k + \Gamma u_k$$

$$y_k = C x_k + D u_k$$

where

$$\Phi = e^{\Delta T_0} \quad \Gamma = \int_0^{T_0} e^s ds B$$

There are more solutions, how get the matrices $\Phi$ and $\Gamma$. Some of them are presented in (Kwon and Han, 2005) and MATLAB offers the $c2d$ function to process this problem.

STATE ESTIMATION

State variables are mostly unmeasurable and they need to be estimated if the state space form is used. For observation (prediction) of state variables are used so-called filters because a part of this estimator works in the same way as classic filter. Other part estimates state space system looks as follows:

$$x_{k+1} = \Phi x_k + \Gamma u_k$$

$$y_k = C x_k + D u_k$$

Matrices $H$ and $L$ are derived from the state space description as given by equations (26) and (27) and length of the filter horizon $N_f$.

MODEL PREDICTIVE CONTROL

Main strategy of the MPC can be described as follows (Orukpe 2012), (Camacho and Bordons, 2004), (Kwon and Han, 2005):

- Step 1: Model is an explicit part of controller and it is used to predict future outputs $y(t)$. Predictions are calculated at each time step based on available information and the unknown future control signal course $u(t)$.
- Step 2: The sequence of future control signals is computed as a result of optimization of the performance criterion. Performance criterion consists of cost function and constraints. Cost functions include future output predictions, reference trajectory and future control signals. Constraints can be applied on both the output and the input of the process.
- Step 3: Only the first control action $u_0$ is transmitted to the process. At the next sampling time, a new output value $y_{k+1}$ is measured and whole sequences is repeated. This strategy is known as the receding horizon concept.

Cost function

The cost function (objective function) is necessary part of MPC. Cost function of linear models is mostly quadratic. Basic task is to minimize this function, which results in a sequence of input control samples. For better results, penalization constant can be added. The cost function is often expressed in matrices and is computed by simulation in the prediction horizon for all sequences of the manipulated variable and the manipulated variable sequence is calculated by a numerical algorithm such as gradient method or some even more sophisticated algorithms. Duration of the calculation is extended after addition of constraints (Haber et al., 2011), (Camacho and Bordons, 2004). Typical cost function look as follows:

$$J_k(N_d, N_y, N_u) = \sum_{j=N_d}^{N_y} (\hat{y}(k+j)|k) - w(k+j))^2 + \lambda \sum_{j=1}^{N_u-1} \Delta u(k+j-1)$$

where:

$$\hat{y}(k+1|k)$$ - sequence of future output values

$$w(k+j)$$ - sequence of desired values

$$\Delta u(k+j-1)$$ - sequence of differences of future control efforts

Cost function contains some optional parameters. Parameters $N_d$ and $N_u$ represents minimal, maximal prediction horizon and they determine the future interval, when a reference signal trajectory should have been followed. It is assumed that reference trajectory is
known. That helps to make a timely reaction, before any changes have been effectively made. Parameter \( N_u \) represents control horizon, which does not necessarily have to coincide with the maximum prediction horizon. Lower value of \( N_u \) brings fewer computations. The coefficient \( \lambda \) represents a weight of the future control signal differences. All these parameters are assumed as tuning parameters (Haber et al., 2011).

**Constraints**

In practice, we often encounter with constraints. It can be constraints of sensor, actuator or some technological limitations. Usually input variables are constrained because they operate only in a certain range of values. In addition, there are also some constraints for the process output variables (with respect to the environment or the safety of workplaces). Ability to work with constraints is one of the main advantages of predictive control, which had an impact on MPC expansion in the industry, where large number of industrial processes is controlled to values close to restrictive conditions (Orukpe 2012). Constraints can be categorized to hard constraints or soft constraints.

Hard constraints can never be exceeded (they must be satisfied). Typical examples are:

- constraint of control input variable: \( u_{\text{min}} \leq u(k) \leq u_{\text{max}} \)
- constraint of output variable: \( y_{\text{min}} \leq y(k) \leq y_{\text{max}} \)

Soft constraints are allowed to exceed the established limits on certain limit tolerance \( \varepsilon \). Typical examples are:

- constraint of control input: \( u_{\text{min}} + \varepsilon \leq u(k) \leq u_{\text{max}} + \varepsilon \)
- constraint of output: \( y_{\text{min}} + \varepsilon \leq y(k) \leq y_{\text{max}} + \varepsilon \)

**Control law**

For the control of the CE152 magnetic levitation system a GPC algorithm was used. Prediction of output values is given by equation (33).

\[
Y_k = PX_k + HU_k + Ld_k
\]  

(33)

where parameter \( d_k \) represents compensation parameter, which gives value of prediction error and can be computed as:

\[
d_k = y_k - CX_k
\]  

(34)

The cost function (32) can be rewritten as follows:

\[
J_k = E_k^TE_k + \Delta U_k^T \lambda \Delta U_k
\]  

(35)

where \( E_k \) represents a vector of predicted future control errors. Sequence of optimal control actions is computed by solving equation (35). This can be done using MATLAB function `quadprog`.

The MATLAB Embedded function block in Simulink environment was used for implementation of the MPC controller. Control law is then computed in this function. Input values are current state estimates \( x_k \), previous control action \( u_{k-1} \), compensation constant \( d_k \) and reference trajectory represented by vector of values of desired future positions \( W_k \). Output value is new action value \( u_k \).

**REAL TIME EXPERIMENTS**

This section presents several real-time control experiments that were performed using CE 152 magnetic levitation plant. The following tuning parameters of the MPC were used for all presented experiments:

\[
N_d = 1, \quad N_r = 15, \quad N_u = 15, \quad N_f = 15, \quad \lambda = 1 \quad (36)
\]

Figure 5 presents control loop courses of the model predictive control of the CE 152 Magnetic Levitation plant:

![Figure 5: MPC control of the real plant](image)

**Comparison with simple controllers**

Performance of the MPC controller was tested by comparing its control circuit courses with two PID-based controllers:

- **PID-DEMO** – pure PID controller with parameters defined as optimal by Humusoft – the manufacturer of the CE 152 plant.
- **PID-Humusoft** – advanced controller based on PID algorithm designed by the Humusoft company

All experiments were performed under the same conditions (same starting point). Figures 6 and 7 presents controlled outputs and control actions respectively. The course of reference trajectory contains only step changes.

![Figure 6: Comparison of outputs – step changes](image)
The control outputs presented in Fig. 7 and Fig. 9 are values produced directly by the controllers. These values are saturated to range <0 MU; 1 MU> by DA converter before the signal enters magnetic levitation laboratory plant. The limits of control signal were incorporated directly into MPC design and therefore it is ensured that MPC output is always within given limits. On the other hand, both PID produced control values out of the saturation range.

Criterions of control quality for both steps and ramp courses of reference signal are summarized in Table 1 where $e_k$ represents control error in step $k$ and $\Delta u_k = u_k - u_{k-1}$ represents control signal difference in step $k$. The sums presented in Table 1 are calculated over the whole time range presented in Fig 6 -9. As sample time $T_0 = 1$ ms was used for all experiments, all compared sums have the same number of summands.

<table>
<thead>
<tr>
<th></th>
<th>PID DEMO</th>
<th>PID Humusoft</th>
<th>GPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steps</td>
<td>$\sum_{k=1}^{N} e_k$</td>
<td>3.3705</td>
<td>0.7387</td>
</tr>
<tr>
<td></td>
<td>$\sum_{k=2}^{N} \Delta u_k$</td>
<td>1.7822</td>
<td>1.8641</td>
</tr>
<tr>
<td>Ramps</td>
<td>$\sum_{k=1}^{N} e_k$</td>
<td>2.2693</td>
<td>0.3836</td>
</tr>
<tr>
<td></td>
<td>$\sum_{k=2}^{N} \Delta u_k$</td>
<td>1.3445</td>
<td>1.1175</td>
</tr>
</tbody>
</table>

It can be seen that performance of MPC is significantly better then performance of classical PID based controllers.

**CONCLUSION**

The CE152 Magnetic levitation system was investigated and its first principle model was derived. Consequently, this model was linearized to obtain a model suitable for control design. A model predictive controller was derived using the state space model and verified by real-time control experiments.

The designed MPC was compared with two more simple PID-based controllers. Comparison of control courses led to the conclusion that the performance of the MPC is significantly better than the performance of the PID-based controllers. On the other hand, computational demands of the MPC are much higher comparing to PID controllers.

Further work will be focused on more detailed examination of the proposed MPC. Special attention will be paid to the role of tuning parameters.
ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme project No. LO1303 (MSMT-7778/2014).

REFERENCES


Humusoft. 1996. CE 152 Magnetic levitation model educational manual


AUTHOR BIOGRAPHIES

Petr Chalupa was born in Zlin in 1976 and graduated from Brno University of Technology in 1999 and received the Ph.D. degree in Technical cybernetics from Tomas Bata University in Zlin in 2003.

He worked as a programmer and designer of an attendance system and as a developer of a wireless alarm system. He was a researcher at the Centre of Applied Cybernetics. Currently he works as a researcher at the Faculty of Applied Informatics, Tomas Bata University in Zlin as a member of CEBIA-Tech team. His research interests are adaptive and predictive control and modelling of real-time systems.

Jakub Novak was born in 1976 and received the Ph.D. degree in chemical and process engineering from Tomas Bata University in Zlin in 2007.

He is a researcher at the Faculty of Applied Informatics, Tomas Bata University in Zlin under a CEBIA-Tech project. His research interests are modeling and predictive control of the nonlinear systems.

Martin Maly graduated from Tomas Bata University in Zlin; Faculty of Applied Informatics in 2015. Nowadays he works as an engineer in TES Vsetin, Czech Republic.