

# MODEL PREDICTIVE CONTROL IN A PENDULUM SYSTEM

Pontus Giselsson  
Department of Automatic Control LTH  
Lund University  
Box 118, SE-221 00 Lund, Sweden  
pontusg@control.lth.se

## ABSTRACT

Model Predictive Control (MPC) is applied to a pendulum system consisting of a pendulum and a cart. The objective of the MPC-controller is to steer the system towards pre-calculated trajectories that move the system from one operating point to another. The sample time of the controller sets hard limitations on the execution time of the optimization routine in the MPC-controller. The optimization problem to solve is cast as a convex optimization problem that can be efficiently solved to allow for real time implementation. The control scheme is applied to a physical pendulum and cart system and the performance of the proposed controller is compared to optimal performance.

## KEY WORDS

Model Predictive Control, Optimal control.

## 1 Introduction

Model Predictive Control is a widely recognized control methodology for control of complex systems with state and control constraints. The idea of Model Predictive Control is to determine a control trajectory by minimizing a cost function based on predictions of future states over a finite time interval, starting with the current state of the system. The first control action in the obtained trajectory is applied to the system. When new state measurements become available, the optimization procedure is repeated with the new measurements used as initial values to the state predictions. There are hard timing constraints on the optimization routine before the control action must be applied. Solving an optimization problem can be a time-consuming task, which is why MPC has traditionally been considered a control methodology for systems with relatively slow dynamics that can be controlled with sample times measured in seconds or minutes. Thorough descriptions of MPC can be found in, e.g. [1] and [2] and examples of industrial processes, that have successfully been controlled using MPC can be found in [3]. Over the past decade, with faster computers and more efficient algorithms, systems with faster dynamics can be controlled using MPC. If the system dimensions are small, explicit MPC can be used, c.f. [4], [5] for linear systems and [6] for systems with non-linear dynamics. In [7] the structure and sparsity inherent in MPC optimization problems are exploited to greatly reduce the

execution time of the on-line optimization routine for systems with linear dynamics.

In this work we try to combine the flexibility of being able to control a non-linear system, with the efficiency of available solvers when the dynamics are linear which usually result in a convex optimization problem. This is obtained by linearizing the non-linear dynamics around a pre-calculated nominal trajectory. These pre-calculated nominal trajectories should, in absence of disturbances, move the system through the non-linear dynamics between two operating points. The objective of the MPC-controller is to steer the system towards the pre-calculated trajectories to achieve the original objective despite disturbances.

The MPC-controller is applied to a physical pendulum and cart system to show its applicability to a non-linear system. The work in this paper is a continuation of the work presented in [8] in which time-optimal trajectories for the pendulum and cart system are calculated. These trajectories are used as nominal feed-forward trajectories in this paper. This means that the objective of the MPC-controller is to control the system towards the nominal trajectories obtained in [8]. The MPC-controller is designed to help any feasible feed-forward trajectory to achieve its optimization objective. A special case of such feed-forward trajectory is (time-optimal) swing-up of the pendulum. There is an extensive literature on the subject of swing-up and control of a pendulum system, e.g. [9], [10]. Most of this literature use the rotary inverted pendulum to avoid problems with a limited track, and have a two-phase controller, one swing-up and one stabilizing controller close to the inverted position. However, in [11] a NMPC-controller is used to swing-up and stabilize a planar pendulum in the inverted position. The optimization need to be terminated after four iterations, before the optimum is reached, to not exceed the allowed execution time. The MPC-controller proposed in this paper makes use of pre-calculated feed-forward trajectories to linearize the non-linear dynamics around. The resulting time-varying linear model is used to state a convex quadratic optimization problem that is solved in each sample in the MPC-controller. Such problems can be solved very efficiently by standard solvers. The purpose of this work is to show that simple methods, which are easy to implement and has relatively low computational complexity, can be used to control highly non-linear systems as the pendulum system. Further the performance degradation com-

pared to what can optimally be achieved is analyzed and shown to be very small.

The article is organized as follows. In Section 2, the problem we are investigating is formulated. Section 3 describes the pendulum system. In Section 4 the minimum-time optimization problems from [8] are stated and the results are presented. The Model Predictive Controller is described in Section 5 and experimental results are presented. The performance of the closed loop system is analyzed in Section 6. Finally in Section 7 the paper is concluded.

## 2 Problem setup

The problem considered in this paper and in the companion paper, [8], is to achieve time-optimal transitions through the non-linear dynamics of the pendulum system. The dynamics of the system are relatively fast, which requires frequent updates of the control signal to be applied to the process. The objective is to develop a feedback solution that do not require too much on-line computational effort, or special purpose optimization algorithms.

The optimization problem described in [8] is of the form

$$\begin{aligned} \min_u \quad & t_f \\ \text{subject to} \quad & \dot{x} = f(x) + g(x)u \\ & x \in X \quad u \in U \\ & x(0) = x_0 \quad x(t_f) = x_{t_f} \end{aligned} \quad (1)$$

where  $f(x)$  and  $g(x)$  describes the non-linear dynamics of the pendulum system. The optimization objective is to minimize the transition time,  $t_f$ , between the initial state,  $x(0)$ , and the terminal state,  $x(t_f)$ , while satisfying state and control constraints. There are two problems considered in [8]. The first problem is concerns swing-up of the pendulum. The second problem is to move the cart from one side of the track to the other with the pendulum starting and stopping in the downward position, while the end-point of the pendulum should avoid a pre-specified obstacle.

The open-loop control trajectories that result from the optimization problems in [8] can be applied to the pendulum system as feed-forward control. The problem considered in this paper is to design an MPC-controller that controls the system towards the pre-calculated time-optimal feed-forward trajectories.

## 3 The Pendulum System

The pendulum system consists of a cart that is mounted on a track with a pendulum freely hanging from the cart. The cart is driven by a Faulhaber DC-motor and a rack and pinion to convert the rotating motion of the motor to linear motion of the cart. Further, the system is equipped with a Hall effect sensor to measure pendulum angle, a current sensor to measure motor current and a magnetic motor encoder that enables us to extract position measurements of the cart. There are also two programmable Atmel ATmega 16 micro

processors mounted on the cart for control purposes. The first micro processor is able to output motor voltage to the motor drive unit and receive current measurements. The second micro processor receives the motor encoder signals and the angle measurement signal. The two micro processors can communicate with each other and the second micro processor communicates with Matlab/Simulink on a PC via the serial interface.

### 3.1 Cart control

The motion of the cart is controlled in a cascaded control structure. See Figure 1 for a schematic view of the cascaded control structure. The innermost loop controls the

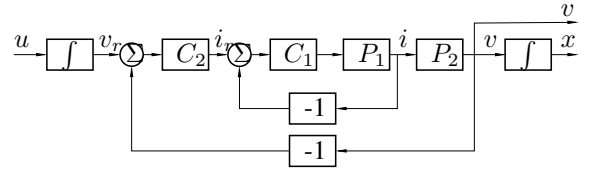


Figure 1. Cascaded control structure for the cart control.

current through the DC-motor.  $P_1$  represents the current dynamics which can be modeled as a first order dynamical system with a time-constant of 0.17 ms.  $C_1$  represents the current-controller, which is a PI-controller, that controls the actual motor current,  $i$ , to its reference,  $i_r$ . This current controller runs at a sampling rate of 28.8 kHz on the first micro processor. The current reference,  $i_r$ , is set by the outer control loop that controls the cart velocity. The current dynamics are fast in comparison to the velocity dynamics, which makes  $i_r \approx i$  a good approximation. The transfer function from  $i$  to  $v$ , i.e.  $P_2$ , is ideally an integrator with a gain. The velocity dynamics are controlled with another PI-controller,  $C_2$ . There are no velocity measurements available. A velocity estimate is however obtained by differentiating the position measurement. This controller runs on the second micro processor at a sampling rate of 1 kHz. The reference to the velocity control loop,  $v_r$ , is received from Matlab/Simulink on the PC. The velocity reference is obtained by integrating the acceleration reference,  $u$ , on the PC-side. To avoid non-smooth behavior of the cart, the velocity reference needs to be updated at a higher frequency than 50 Hz. Therefore the acceleration reference,  $u$ , is also sent to the velocity controller from the PC. The velocity reference is updated in the micro processor according to  $v_r(t) = v_r(t_0) + u(t_0)(t - t_0)$  in every sample, where  $t_0$  is the time when the last references was received from the PC,  $t \in [t_0, t_0 + h]$  and  $h$  is the PC communication sampling time. These updates are consistent with the velocity reference in the next sample from the PC which is  $v_r(t_0 + h) = v_r(t_0) + u(t_0)h$ . This results in a smooth acceleration profile of the cart.

This cascaded control structure is suitable when fast closed loop dynamics from  $v_r$  to  $v$  is desired. In the eyes of

the slower MPC control loop on the PC,  $v_r = v$  is a good approximation. Due to this, the cart motion can accurately be modeled as a double integrator from control signal,  $u$ , to cart position,  $x$ .

### 3.2 System modeling

Due to the low level control previously described the cart position,  $p$ , depends on the control signal,  $u$ , according to

$$\ddot{p} = u. \quad (2)$$

The pendulum is attached to the cart. When the pendulum is swinging, reaction forces in the mounting point creates disturbances to the cart motion. These disturbances are attenuated by the cascaded control structure on the cart, making the double integrator model of the cart accurate despite disturbances. The pendulum is modeled as a simple gravity pendulum, in which the weight of the rod is neglected. The pendulum dynamics are well known, let  $\theta$  be the pendulum angle and the dynamics are described by

$$\ddot{\theta} = -\frac{g}{l} \sin \theta + \frac{u}{l} \cos \theta, \quad (3)$$

where  $\theta = 0$  is defined to be the pendulum downward position,  $g$  is the gravitational acceleration,  $l$  is the length of the idealized simple gravity pendulum which is chosen to match the pendulum frequency around the downward position and  $u$  is the cart acceleration. The full system dynamics are described by Equations (2) and (3). Note that since the cart acceleration is used as control signal, the cart and pendulum dynamics are decoupled. They can be seen as two separate dynamical systems that are driven by the same control signal.

The position of the cart and the pendulum angle are defined such that the pendulum end point in the horizontal direction,  $x_{pend}$ , and in the vertical direction,  $y_{pend}$ , are given by

$$\begin{aligned} x_{pend} &= p - l \sin \theta \\ y_{pend} &= -l \cos \theta. \end{aligned}$$

## 4 Optimal Feed Forward Trajectories

Two different minimum-time optimal control problems are considered in this paper. The first problem is a minimum-time swing-up problem with some additional constraints. The second problem is a path-constrained minimum-time problem. The optimal feed forward trajectories used in this paper are identical to and generated as in [8]. The optimization problems are solved using the JModelica.org platform, [12], which allow for solving dynamic optimization problems by specifying the dynamical model, the cost function and constraints on a high level. The problems and the solution from [8] are stated below.

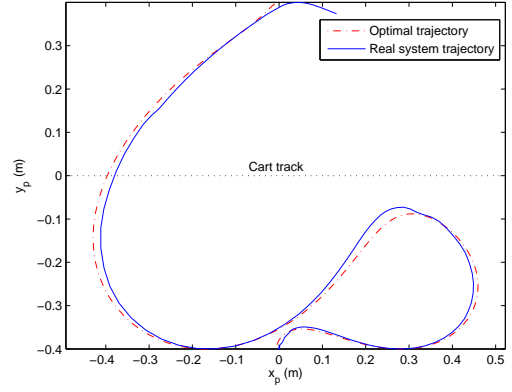


Figure 2. Experimental results for swing-up problem when control trajectory applied in open loop. The optimal pendulum end point trajectory is also plotted for comparison reasons.

### 4.1 Pendulum Swing-up

The optimization objective is to reach the inverted position as fast as possible, starting from the downward position. Further constraints include that the cart should start and stop at the same position. The cart and angular velocities should be zero when the pendulum has reached the inverted position. The applied control signal, i.e., the cart acceleration,  $u$ , is limited to be in the interval  $\pm 5\text{m/s}^2$  and its derivative must satisfy  $-100\text{m/s}^3 \leq \dot{u} \leq 100\text{m/s}^3$ . Since the cart track is finite, the cart position must satisfy  $-0.5\text{m} \leq p \leq 0.5\text{m}$ . The optimization problem is stated mathematically in (4)

$$\begin{aligned} \min_u \quad & t_f \\ \text{subject to} \quad & \ddot{\theta} = -\frac{g}{l} \sin \theta + \frac{u}{l} \cos \theta \\ & \ddot{p} = u \\ & -0.5 \leq p \leq 0.5 \\ & |u| \leq 5 \quad |\dot{u}| \leq 100 \\ & \theta(t_f) = \pi \quad \dot{\theta}(t_f) = 0 \\ & p(t_f) = 0 \quad \dot{p}(t_f) = 0 \end{aligned} \quad (4)$$

where  $t_f$  is the final time. To analyze the plant-model accuracy the optimal feed-forward trajectory was applied to the physical plant with identical initial conditions as in the optimization, i.e. zero. The resulting pendulum end point trajectory, together with the optimal trajectory, is found in Figure 2.

### 4.2 Optimization with path-constraints

In this optimization problem the cart should move from one side of the track to the other side while the end point of the pendulum avoids a certain obstacle defined by

$$\left( \frac{x_{pend} - 0.5}{0.05} \right)^2 + \left( \frac{y_{pend} + 0.4}{0.3} \right)^2 = 1$$

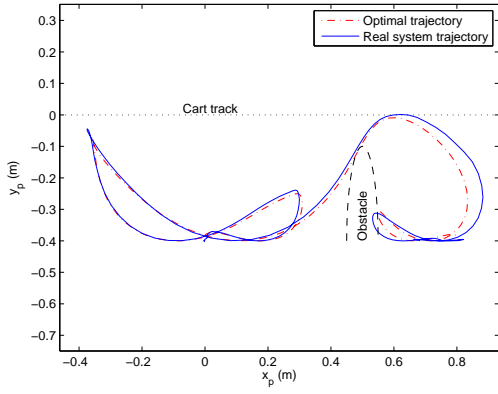


Figure 3. Experimental results for path constrained problem when control trajectory applied in open loop. The optimal pendulum end point trajectory is also plotted for comparison reasons.

The pendulum should start and stop at rest in the downward position. Track and control limitations are equivalent to in the swing-up problem. We get the following optimization problem

$$\begin{aligned}
 & \min_u \quad t_f \\
 & \text{subject to} \quad \ddot{\theta} = -\frac{g}{l} \sin \theta + \frac{u}{l} \cos \theta \\
 & \quad \quad \quad \ddot{p} = u \\
 & \quad \quad \quad x_{pend} = p - l \sin \theta \\
 & \quad \quad \quad y_{pend} = -l \cos \theta \\
 & \quad \quad \quad \left( \frac{x_{pend} - 0.5}{0.05} \right)^2 + \left( \frac{y_{pend} + 0.4}{0.3} \right)^2 \geq 1 \\
 & \quad \quad \quad -0.1 \leq p \leq 0.9 \\
 & \quad \quad \quad |u| \leq 5 \quad |\dot{u}| \leq 100 \\
 & \quad \quad \quad \theta(t_f) = 0 \quad \dot{\theta}(t_f) = 0 \\
 & \quad \quad \quad p(t_f) = 0.8 \quad \dot{p}(t_f) = 0
 \end{aligned} \quad (5)$$

where  $t_f$  again is the final time. Optimization results, as well as the trajectory obtained when applying the control action to the physical system, again with initial conditions zeros, are found in Figure 3.

The result from the optimization problems (4) and (5) are continuous time state and control trajectories. The discrete time counterparts, that will be used in the sequel, are obtained by taking the values from the corresponding continuous trajectories at each sampling point. The discrete time variables are denoted  $p^0(t)$ ,  $\dot{p}^0(t)$ ,  $\theta^0(t)$ ,  $\dot{\theta}^0(t)$ ,  $u^0(t)$  and  $x^0(t) = [p^0(t) \dot{p}^0(t) \theta^0(t) \dot{\theta}^0(t)]^T$ .

## 5 Model Predictive Control

The open-loop control trajectories from the previous section gives close to optimal state trajectories when applied to the physical pendulum system, see Figures 2 and 3. This behavior cannot be expected when disturbances are present. In the optimization problems in the previous section it is specified that the pendulum should start at rest in

the downward position. When there are disturbances in the initial state of the pendulum, the trajectories do not match very well. This can be seen in Figure 4 for the swing-up and in Figure 5 for the path constrained problem. The experiments were started when the pendulum was swinging back and forth with a magnitude of approximately  $45^\circ$ . In this section we introduce MPC-feedback to cope with such disturbances. The objective is to design a well performing MPC-feedback which is easy to implement and satisfies the real-time requirements of the physical pendulum system.

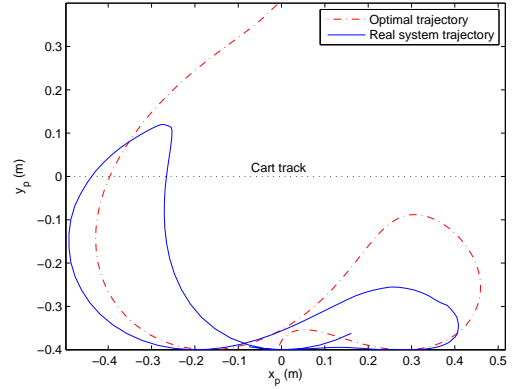


Figure 4. Pendulum end point trajectory for the real system when pendulum is swinging initially and no feedback is used.

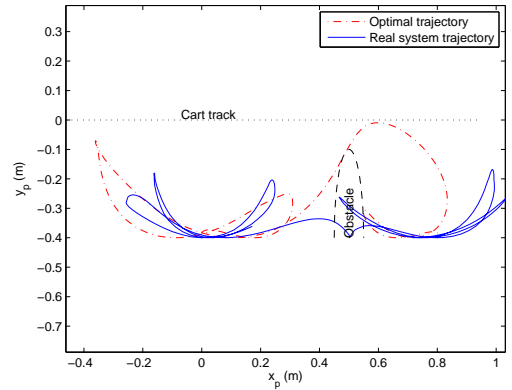


Figure 5. Pendulum end point trajectory for the real system when pendulum is swinging initially and no feedback is used.

### 5.1 Discrete time pendulum model

The continuous dynamics of the pendulum system must be discretized and linearized to be used in this Model Predictive Control context. In each time instant,  $t$ , the system, (2)-(3), is linearized around the nominal states,  $x^0(t)$ , and nominal control,  $u^0(t)$ , which are obtained from the optimization problems in the previous section. Since the cart

dynamics are linear, we only need to linearize the pendulum dynamics around the nominal states. To achieve this, introduce  $\Delta z_\theta(t)$  which is the deviation from the linearization point for the pendulum states and  $\Delta v(t)$  which is the control signal for this linearized model. This gives the following linearized pendulum dynamics

$$\begin{aligned}\dot{\Delta z}_\theta(t) &= \begin{pmatrix} 0 & 1 \\ -\frac{g}{l} \cos \theta^0(t) & 0 \end{pmatrix} \Delta z_\theta(t) + \begin{pmatrix} 0 \\ \frac{1}{l} \cos \theta^0(t) \end{pmatrix} \Delta v(t) \\ &= \underbrace{\begin{pmatrix} 0 & 1 \\ -\omega^0(t)^2 & 0 \end{pmatrix}}_{A(t)} \Delta z_\theta(t) + \underbrace{\begin{pmatrix} 0 \\ \frac{\omega^0(t)^2}{g} \end{pmatrix}}_{B(t)} \Delta v(t)\end{aligned}$$

where  $\omega^0(t)^2 = \frac{g}{l} \cos \theta^0(t)$ . The resulting linear time-varying dynamics depend only on the nominal pendulum angle,  $\theta^0(t)$ . To obtain a discrete time model, the linearized model is discretized using zero-order-hold according to

$$\Delta x_\theta(t+h) = e^{A(t)h} \Delta x_\theta(t) + \int_{s=0}^h e^{A(t)(h-s)} B \Delta u ds$$

where  $\Delta u$  is a piece-wise constant control signal and

$$\begin{aligned}e^{A(t)h} &= I + A(t)h + \frac{(A(t)h)^2}{2!} + \frac{(A(t)h)^3}{3!} + \dots \\ &= I + \begin{pmatrix} 0 & 1 \\ -\omega^0(t)^2 & 0 \end{pmatrix} h - \begin{pmatrix} \omega^0(t)^2 & 0 \\ 0 & \omega^0(t)^2 \end{pmatrix} \frac{h^2}{2!} \\ &\quad + \begin{pmatrix} 0 & -\omega^0(t)^2 \\ \omega^0(t)^4 & 0 \end{pmatrix} \frac{h^3}{3!} + \begin{pmatrix} \omega^0(t)^4 & 0 \\ 0 & \omega^0(t)^4 \end{pmatrix} \frac{h^4}{4!} \\ &\quad + \begin{pmatrix} 0 & \omega^0(t)^4 \\ -\omega^0(t)^6 & 0 \end{pmatrix} \frac{h^5}{5!} - \begin{pmatrix} \omega^0(t)^6 & 0 \\ 0 & \omega^0(t)^6 \end{pmatrix} \frac{h^6}{6!} \\ &\quad + \begin{pmatrix} 0 & -\omega^0(t)^6 \\ \omega^0(t)^8 & 0 \end{pmatrix} \frac{h^7}{7!} + \dots \\ &= \begin{pmatrix} \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} (\omega^0(t)h)^{2k} & \dots \\ -\omega^0(t) \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} (\omega^0(t)h)^{2k+1} & \dots \\ \dots & \dots \\ \frac{1}{\omega^0(t)} \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} (\omega^0(t)h)^{2k+1} & \dots \\ \dots & \dots \\ \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} (\omega^0(t)h)^{2k} & \dots \end{pmatrix} \\ &= \begin{pmatrix} \cos \omega^0(t)h & \frac{1}{\omega^0(t)} \sin \omega^0(t)h \\ -\omega^0(t) \sin \omega^0(t)h & \cos \omega^0(t)h \end{pmatrix}\end{aligned}$$

The last equality comes from the Taylor series expansion of cosine and sine. When  $e^{A(t)h}$  is known, the integral can

easily be calculated

$$\begin{aligned}\int_{s=0}^h e^{A(t)(h-s)} B \Delta u ds &= \Delta u \int_{s=0}^h \begin{pmatrix} \frac{\omega^0(t)}{g} \sin \omega^0(t)(h-s) \\ \frac{\omega^0(t)^2}{g} \cos \omega^0(t)(h-s) \end{pmatrix} ds \\ &= \Delta u \begin{pmatrix} \frac{1}{g} \cos \omega^0(t)(h-s) \\ -\frac{\omega^0(t)}{g} \sin \omega^0(t)(h-s) \end{pmatrix}_{s=0}^h \\ &= \begin{pmatrix} \frac{1}{g}(1 - \cos \omega^0(t)h) \\ \frac{\omega^0(t)}{g} \sin \omega^0(t)h \end{pmatrix} \Delta u\end{aligned}$$

A discrete time model of the double integrator, (2), is well known to be

$$\Delta x_p(t+h) = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix} \Delta x_p(t) + \begin{pmatrix} \frac{h^2}{2} \\ h \end{pmatrix} \Delta u(t)$$

This gives the following full model that is linearized around the nominal feed-forward trajectories

$$\Delta x(t+1) = \Phi(t) \Delta x(t) + \Gamma(t) \Delta u(t) \quad (6)$$

where

$$\begin{aligned}\Phi(t) &= \begin{pmatrix} 1 & h & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos \omega^0(t)h & \frac{1}{\omega^0(t)} \sin \omega^0(t)h \\ 0 & 0 & -\omega^0(t) \sin \omega^0(t)h & \cos \omega^0(t)h \end{pmatrix} \\ \Gamma(t) &= \begin{pmatrix} \frac{h^2}{2} \\ h \\ \frac{1}{g}(1 - \cos \omega^0(t)h) \\ \frac{\omega^0(t)}{g} \sin \omega^0(t)h \end{pmatrix}\end{aligned}$$

The resulting model is a time-varying discrete time linear dynamical system which is to be used in the Model Predictive Controller.

## 5.2 Linear MPC

The model developed in the previous section is unstable for certain pendulum angles. Due to this, predicting future states directly with (6) would result in poor predictions. To avoid that, a LQ-feedback term that depends on the pendulum angle is introduced,  $u_{fb}(t) = -L(t)\Delta x(t)$ . For accurate predictions, the feedback vector  $L(t)$  must be known for all  $t$  in the prediction horizon. This can be achieved by calculating the LQ-optimal feedback in every time instant in the horizon. If the feedback is applied with the same sample period as the MPC-controller,  $h$ , the prediction model to be used in the MPC-controller for each time instant,  $t$ , is

$$\begin{aligned}\Delta x(\tau+1|t) &= (\Phi(\tau) - \Gamma(\tau)L(\tau))\Delta x(\tau|t) + \Gamma(\tau)\Delta u(\tau|t) \\ &= \Phi^L(\tau)\Delta x(\tau|t) + \Gamma(\tau)\Delta u(\tau|t)\end{aligned} \quad (7)$$

with initial condition  $\Delta x(t|t) = \Delta x(t)$  and  $\Delta u(t|t) = \Delta u(t)$  which is applied to the process. This model is

stable for all time instants. In the MPC-problem we are dealing with state and control signal deviations from the nominal trajectories. The constraints on the system deals with actual state and control limitations. In every sample,  $u(t) = u^0(t) + \Delta u(t) + u_{fb}(t)$ , is sent as control signal to the system. The maximal allowed acceleration is  $\pm 7m/s^2$ . This is chosen since the inner control loops do not saturate for that choice. The control constraint set is defined as

$$U = \{\Delta u \in \mathbb{R} \mid |\Delta u + u^0 + u_{fb}| \leq 7\}. \quad (8)$$

The only state constraint present is due to track limitations. The track where the cart is moving is slightly longer than 1 meter. This gives the following state constraint set

$$X = \{\Delta x \in \mathbb{R}^4 \mid \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} (\Delta x + x^0) \leq 1 - p_0 \\ \begin{bmatrix} -1 & 0 & 0 & 0 \end{bmatrix} (\Delta x + x^0) \leq p_0\}, \quad (9)$$

where the track length is set to 1 meter and  $p_0$  is the initial position of the cart on the track.

The MPC problem to be solved in each time sample,  $t$ , is

$$\min_{\Delta u(\cdot)} \sum_{\tau=t}^{t+N} \Delta x(\tau|t)^T Q \Delta x(\tau|t) + \Delta u(\tau|t)^T R \Delta u(\tau|t)$$

with  $Q \succeq 0$  and  $R \succ 0$ , subject to (7), (8) and (9) and for  $\tau = t, \dots, t + N$ . Since the objective function is quadratic, the dynamics linear and the control and state constraint sets are linear in  $\Delta u$  and  $\Delta x$  the resulting optimization problem is a quadratic program.

The outcome of the optimization is twofold. Firstly, the control signal to be applied,  $\Delta u(t)$ , is calculated. Secondly, a prediction of the deviation from the optimal trajectory is obtained. This information can be used to improve the estimated pendulum angle trajectory to linearize around in subsequent samples.

All state variables cannot be measured directly, only cart and pendulum positions are measured. The cart velocity is estimated by a derivative filter on the second micro processor which is updated in 1 kHz. The most recent estimate is sent to the PC when it is needed by the MPC-controller. The pendulum angular velocity is estimated by a derivative filter in Simulink which is updated ones in every MPC-sample. This gives accurate enough estimates since the pendulum dynamics are quite slow.

### 5.3 Implementational aspects

The MPC controller described previously is implemented in Simulink and communicates with one of the micro-processors on the cart via serial interface. The control horizon is chosen to  $N = 50$ , which gives 50 control variables and 200 state variables to decide in each optimization. The sampling time, which is chosen to  $h = 0.02s$ , sets hard limitations on the allowed execution time of the MPC-controller before the control signal must be applied. The software used to solve the QP-problem in each sample

is a solver written in C with a Matlab interface, [13]. The execution time of one MPC-cycle with this setup is close to 6ms on a standard PC with a 2.6 GHz processor. This gives a processor load at around  $6ms/20ms = 30\%$  when running this MPC-controller on the physical pendulum system.

### 5.4 Experimental results

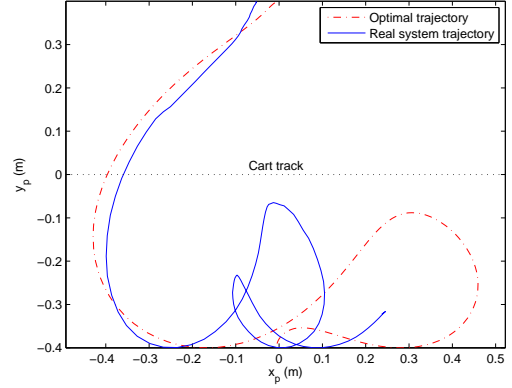


Figure 6. Pendulum end point trajectory for the real system when pendulum is swinging initially and feedback is used.

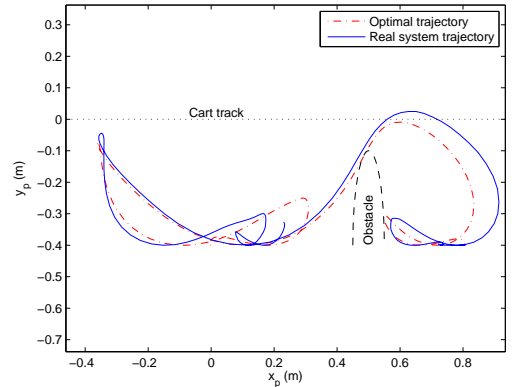


Figure 7. Pendulum end point trajectory for the real system when pendulum is swinging initially and feedback is used.

Resulting pendulum end point trajectories, when applying the MPC-feedback to the physical system, are visualized in Figures 6 and 7. The weight matrix,  $Q$  is chosen to penalize position and pendulum angle error more than the corresponding velocities. Further the  $R$ -matrix is chosen rather small to not penalize control action too much. The feedback gain vector  $L$  is calculated by LQ-methods using unit weights on both states and with initial pendulum swings whose magnitude are comparable to the control. The experiments are performed with initial pendulum swings whose magnitude are comparable to the magnitude of the initial swings when no feedback was present

in Figures 4 and 5, namely around  $45^\circ$ . Due to the initial swinging, the trajectories are far from the optimal ones in the beginning but the feedback brings the actual trajectories closer to the optimal trajectories with time. This shows that the introduced model approximations are accurate enough to achieve good performance in the physical pendulum system. Videos of similar experiments, with and without initial swinging of the pendulum, can be found in [14].

## 6 Performance evaluation

In this section the performance of the applied control scheme is compared to optimal performance. The performance evaluation is made to analyze the effects caused by the linearization of the pendulum dynamics as well as effects from other approximations on the best achievable performance. The only state variable that affects the linearized model is the pendulum angle. For this reason, we only investigate performance degradation and stability region for different initial pendulum angles. The comparison is made with swing-up trajectories as feed-forward. The performance of the control scheme is calculated as the running stage cost. The control action is applied to the continuous pendulum system (2)-(3) with initial conditions  $x(0) = [p(0) \dot{p}(0) \theta(0) \dot{\theta}(0)]^T = [0 \ 0 \ \theta_0 \ 0]^T$  for different initial pendulum angles  $\theta_0$ . The applied control signal is piece-wise constant with pieces lasting one sample time,  $h$ . The running stage cost is defined as

$$\sum_{t=0}^K \Delta x(t)^T Q \Delta x(t) + \Delta u(t)^T R \Delta u(t)$$

where  $\Delta x(t) = x(t) - x^0(t)$ ,  $x(t)$  contains the actual continuous time system state trajectories and  $x^0(t)$  are nominal state trajectories corresponding to the applied feed-forward  $u^0(t)$ . This is a natural performance metric to choose since the MPC-controller minimizes a truncated version of this sum. The  $K$  variable is a large number such that the corresponding stage cost is small enough to have negligible impact on the sum. This performance measure quantifies how well the MPC-feedback manages to control the continuous time pendulum system towards the nominal trajectories.

This performance can be compared to the optimal performance with is calculated by solving the following optimization problem

$$\begin{aligned} \min_{\Delta u_c(\cdot)} \quad & \int_{t=0}^{t_f} \Delta x_c(t)^T Q \Delta x_c(t) + \Delta u_c(t)^T R \Delta u_c(t) dt \\ \text{subject to} \quad & \ddot{\theta}_c = -\frac{g}{l} \sin \theta_c + \frac{u_c}{l} \cos \theta_c \\ & \dot{p}_c = u_c \\ & \Delta x_c = x_c - x^0 \\ & u_c = \Delta u_c + u^0(t) - L(t) \Delta x_c \\ & |p^0 + \Delta p_c| \leq 0.5 \quad |u_c| \leq 7 \\ & \theta_c(0) = \theta_0 \quad \dot{\theta}_c(t_f) = 0 \\ & p_c(t_f) = 0 \quad \dot{p}_c(t_f) = 0 \\ & \theta_c(t_f) = \pi \quad \dot{\theta}_c(t_f) = 0 \\ & p_c(t_f) = 0 \quad \dot{p}_c(t_f) = 0 \end{aligned} \quad (10)$$

where the subscript,  $c$ , means that the system is controlled with a continuous feedback and  $t_f = Kh$  to have the same amount of time to control the system to the terminal state as in the MPC case. The problem is solved using the JModelica.org platform, just like the time-optimal feed-forward trajectories in Section 4. The resulting optimal trajectories are sampled with sample time,  $h$ , and summed according to

$$\sum_{t=0}^K \Delta x_c(t)^T Q \Delta x_c(t) + \Delta u_c(t)^T R \Delta u_c(t)$$

to get a value comparable to the MPC-performance.

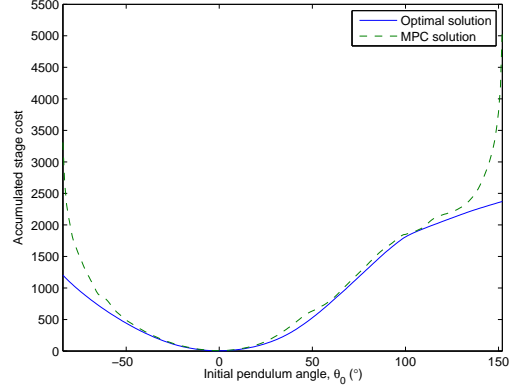


Figure 8. Performance comparison between applied MPC and optimal feedback for all stabilizing initial pendulum angles.

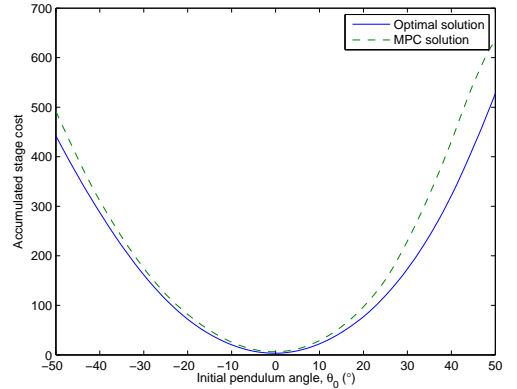


Figure 9. Performance comparison between applied MPC and optimal feedback for initial pendulum angles,  $\theta_0 \in [-50^\circ \ 50^\circ]$ .

The optimization problem, (10), is less restrictive than the discrete-time linear MPC feedback since we do not require a piece-wise constant control action in this formulation. Further the feedback-term,  $L(t) \Delta x$ , is calculated as the continuous optimal LQ-feedback corresponding to the discrete-time optimal LQ-feedback in the MPC-setting. The optimal value of this optimization problem is a lower

bound to what can be achieved for the MPC feedback. The optimal cost and the feedback MPC cost are shown in Figures 8 and 9 for different initial pendulum angles. Figure 8 shows the cost for all initial pendulum angles that the MPC-system manages to control to the terminal state, i.e. the inverted pendulum position. Close to the stability boundary the cost grows dramatically. It is interesting to note that the stability region is not symmetric around the downward position of the pendulum, this is due to the fact the feed-forward swing-up has a predefined trajectory which appears easier to follow if started on one side than the other. Further it is quite remarkable that, despite linearization of the system model, the system is stabilized for such a large region of initial pendulum angles. Figure 9 is a zoom-in of Figure 8 and shows the cost for initial pendulum angles that are in the interval between  $-50^\circ$  and  $50^\circ$ . The MPC cost is relatively close to the optimal one in this region and the feedback system performs well for initial pendulum swings within this region.

## 7 Conclusion

We have developed an Model Predictive Controller that controls the actual system trajectories towards pre-calculated feed-forward trajectories in a pendulum system. The feed-forward trajectories takes the system from one operating point to another. We have studied one swing-up example and one path-constrained example which have been applied to the physical pendulum system. Since the MPC-controller is based on a linearization of the system dynamics, the stabilizing region is limited if the actual dynamics differ too much from the modeled dynamics. The stability region, with respect to initial pendulum positions, turned out to be very large. Further, the performance of the MPC-controller was close to optimal performance inside this stability region.

## References

- [1] J. M. Maciejowski. *Predictive control with constraints*. Prentice Hall, Essex, England, 2002.
- [2] Manfred Morari and Jay H. Lee. Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23:667–682, 1999.
- [3] Eduardo F. Camacho and Carlos A. Bordons. *Model Predictive Control in the Process Industry*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.
- [4] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The Explicit Linear Quadratic Regulator for Constrained Systems. *Automatica*, 38(1):3–20, January 2002.
- [5] A. Bemporad and C. Filippi. Suboptimal explicit MPC via approximate multiparametric quadratic programming. In *IEEE Conference on Decision and Control*, Orlando, Florida, 2001.
- [6] Tor A. Johansen. Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica*, 40(2):293 – 300, 2004.
- [7] Yang Wang and Stephen Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, March 2010.
- [8] Pontus Giselsson, Johan Åkesson, and Anders Robertsson. Optimization of a pendulum system using optimica and modelica. In *Proceedings of the 7th International Modelica Conference 2009*, Como, Italy, September 2009.
- [9] Magnus Wiklund, Anders Kristenson, and Karl Johan Åström. A new strategy for swinging up an inverted pendulum. In *Preprints IFAC 12th World Congress*, Sydney, Australia, July 1993.
- [10] Karl Johan Åström and Katsuhisa Furuta. Swinging up a pendulum by energy control. *Automatica*, 36:278–285, February 2000.
- [11] Adam Mills, Adrian Wills, and Brett Ninness. Non-linear model predictive control of an inverted pendulum. In *ACC09*, jun 2009.
- [12] Johan Åkesson, Magnus Gäfvert, and Hubertus Tummescheit. Jmodelica—an open source platform for optimization of modelica models. In *Proceedings of MATHMOD 2009 - 6th Vienna International Conference on Mathematical Modelling*, Vienna, Austria, February 2009.
- [13] Adrian Wills. QP-solver in C, 2007. <http://sigpromu.org/quadprog/>.
- [14] Pontus Giselsson. Pendulum videos, 2009. <http://www.control.lth.se/user/pontus.giselsson/>.