ISSN 0280-5316
ISRN LUTFD2/TFRT--7624--SE

Gradient-Based Model Predictive Control in a Pendulum System

Pontus Giselsson

Lund University
Department of Automatic Control
October 2012

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
TECHNICAL REPORT

Date of issue

Document Number
ISRN LUTFD2/TFRT--7624--SE

Author(s)
Pontus Giselsson

Supervisor

Sponsoring organization

Title and subtitle
Gradient-Based Model Predictive Control in a Pendulum System

Abstract
Model predictive control (MPC) is applied to a physical pendulum system consisting of a pendulum and a cart. The objective of the MPC controller is to steer the system towards precomputed, time-optimal feedforward trajectories that move the system from one stationary point to another. The sample time of the controller sets hard limitations on the execution time of the optimization algorithm in the MPC controller. The MPC optimization problem is stated as a quadratic program, which is solved using the algorithm presented in [10]. The algorithm in [10] is an accelerated gradient method that is applied to solve a dual formulation of the MPC optimization problem. Experiments show that the optimization algorithm is efficient enough to be implemented in a real-time pendulum application.

Keywords
Model Predictive Control, Pendulum system, Gradient-Based Optimization

Classification system and/ or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-21

Recipient's notes

Security classification

Contents

1. Introduction

Model predictive control (MPC) 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, with the current state of the system as initial condition. 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 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. Over the past decade, faster computers and increasingly efficient algorithms have been developed. This development has enabled for systems with faster dynamics to be controlled using MPC. If the system dimensions are small, explicit MPC can be used, [1, 2] for linear systems and [3] for systems with nonlinear dynamics, to speed up online execution times. In [4] the structure and sparsity inherent in MPC optimization problems are exploited to reduce the execution time when using an interior point algorithm to solve the online optimization problem. For systems with input constraints only, accelerated gradient methods are used to solve the resulting MPC optimization problem in [5]. For more on MPC see [6, 7, 8], and for examples of industrial processes that have successfully been controlled using MPC see [9].

In this paper optimal control and model predictive control of a pendulum system is considered. The pendulum system consists of a cart, which is mounted on a track, and has a pendulum attached to it. The track length sets limitations on the cart movement. Two minimum time optimization problems for the pendulum system are posed: one swing-up problem and one translation problem with constraints on the location of the pendulum endpoint. The solutions to the minimum time optimization problems are precomputed and used as feedforward trajectories. We introduce feedback by designing an MPC controller with the objective to steer the system towards the optimal feedforward trajectories. The model used in the MPC optimization problem is a time dependent linear system that is obtained by linearizing the nonlinear pendulum dynamics around the precomputed feedforward trajectories. We use a quadratic cost and linear constraints in the MPC optimization problem. This gives a quadratic program to be solved in the MPC controller. To solve the quadratic program the algorithm presented in [10] is used. The algorithm is an accelerated gradient method applied to a dual formulation of the optimization problem.

This report is based on the material in [11] and to some extent on the material in [12]. The paper is organized as follows. In Section 2, the problem formulated is stated. Section 3 describes the pendulum system. In Section 4 the minimum-time optimization problems are stated and the optimal trajectories are plotted. The model predictive controller is described in Section 5 and experimental results are presented. Finally, in Section 6 the paper is concluded.

2. Problem Setup

The problem considered in this paper is to achieve time optimal transitions through the nonlinear dynamics of the pendulum system. We use the following minimum time optimization formulation

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

where $f(x)$ and $g(x)$ describes the nonlinear 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 defined by the set $\mathcal{X}$. We consider two different minimum time optimization problems. The first problem 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 must avoid a prespecified fixed obstacle.

The resulting optimal control trajectories are applied to the pendulum system as feedforward control trajectories. The problem considered in this paper is to design an MPC controller that controls the system towards the precomputed optimal feedforward trajectories. The resulting optimization problem is solved using the method presented in [10] in which an accelerated gradient method is applied to a dual formulation to the optimization problem. The dynamics of the system are relatively fast, which sets requirements on the execution time of the optimization algorithm.

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 length of the pendulum is $l = 0.4$ m. 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 to extract position measurements of the cart. There are also two programmable Atmel ATmega 16 microprocessors mounted on the cart for control purposes. The first microprocessor can output motor voltage to the motor drive unit and receive current measurements. The second microprocessor receives the motor encoder signals and the angle measurement signal. The two microprocessors can communicate with each other and the second microprocessor communicates with Matlab/Simulink on a PC via the serial interface.

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 current through the DC-motor. P_1 represents the current dynamics that is modeled as a first order dynamical system with a time constant of 0.17 ms. C_1 represents the current controller,

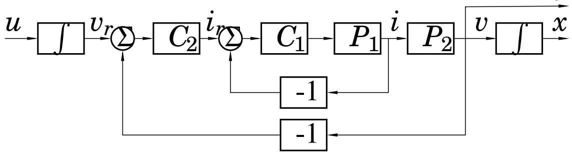


Figure 1 Cascaded control structure for the cart control.

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 microprocessor. 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 modeled as an integrator with a gain. The velocity dynamics are controlled with another PI-controller, C_2. This controller runs on the second microprocessor at a sampling rate of 1 kHz. There are no velocity measurements available. A velocity estimate is obtained by applying a derivative filter to the position measurement in the micro processor at a frequency of 1 kHz. The reference to the velocity control loop, v_r, is received from Matlab/Simulink that runs on a PC. The velocity reference sent form the PC is updated at a frequency of about 50 Hz. We denote the corresponding sampling time by h. The settling time for the velocity controller is faster than the update frequency of the velocity reference. To avoid nonsmooth behavior of the cart, the velocity reference is updated in a first-order-hold manner in the microprocessor. That is, a piece-wise constant acceleration reference u is sent to the microprocessor. The velocity reference is updated internally in the microprocessor at the same rate as the velocity controller. The reference is updated according to $v_r(t) = v_r(t_0) + u(t_0)(t - t_0)$ where $v_r(t_0)$ is the integrated velocity reference and $u(t_0)$ is the acceleration reference at sampling time t_0, and $t \in [t_0, t_0 + h]$.

This cascaded control structure is suitable when fast closed loop dynamics from v_r to v is desired. Since the PC communication is performed at a much slower frequency than the velocity controller updates, $v_r = v$ is a good approximation. Using this approximation, the cart motion can be modeled as a double integrator from acceleration reference to cart position.

System Modeling

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

$$\ddot{p} = u. \tag{2}$$

A 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 these 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 θ be the pendulum angle and the dynamics are described by

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

where $\theta = 0$ is defined to be the pendulum downward position, g is the gravitational acceleration, l is the length of the pendulum, 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_p, and in the vertical direction, y_p, are given by

$$x_p = p - l\sin\theta, \qquad y_p = -l\cos\theta.$$

4. Optimal Feedforward Trajectories

Two different minimum-time optimal control problems are considered in this paper. The first problem is a minimum-time swing-up problem with additional constraints on cart position and control signal magnitude. The second problem is a path-constrained minimum-time problem. The optimization problems are solved using the JModelica.org platform [13] which allow for solving dynamic optimization problems by specifying the dynamical model, the cost function and constraints using a high-level language. The optimal control problems and the solutions obtained by the JModelica.org platform are presented below. For more information on how these optimal control problems were solved, see [12].

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 below.

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

where t_f is the final time. To analyze the plant-model accuracy, the optimal feedforward trajectory was applied to the physical plant with the same initial conditions as in the opti-

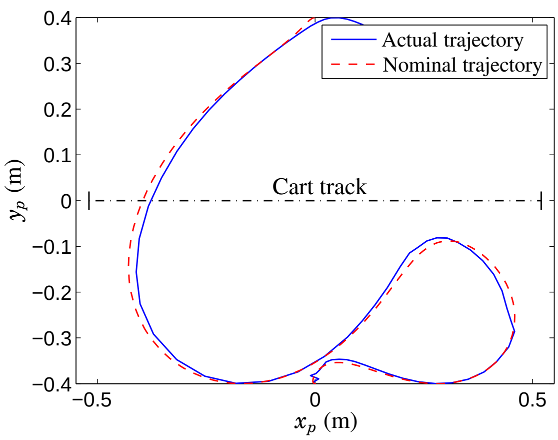


Figure 2 Pendulum end point trajectories for swing-up problem. Both the optimal trajectory and the trajectory obtained when optimal control trajectory is applied to the physical pendulum system in open loop, are plotted.

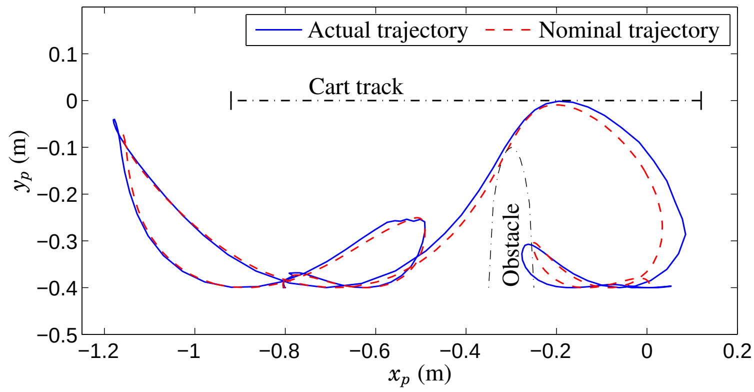


Figure 3 Pendulum end point trajectories for path-constrained problem. Both the optimal trajectory and the trajectory obtained when optimal control trajectory is applied to the physical pendulum system in open loop, are plotted.

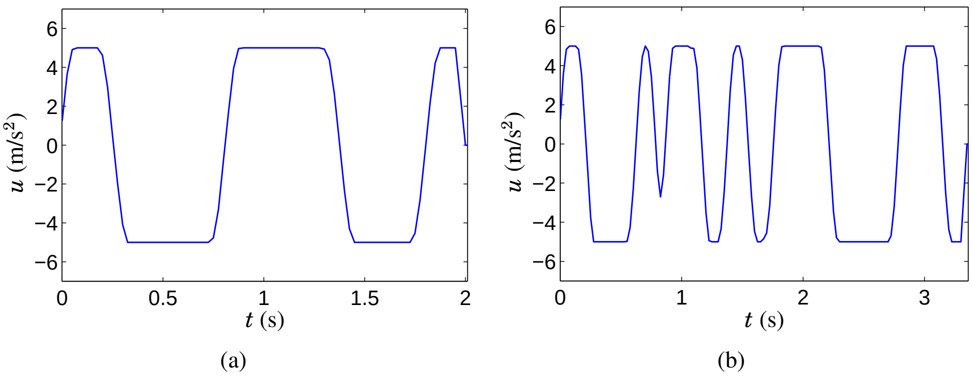


Figure 4 Optimal control trajectories for the swing-up problem (a) and the path-constrained problem (b).

mization. The resulting pendulum end point trajectory, together with the optimal trajectory, is found in Figure 2. The optimal control trajectory for the swing-up example is found in Figure 4(a).

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 must avoid an obstacle defined by

$$\left(\frac{x_p + 0.3}{0.05}\right)^2 + \left(\frac{y_p + 0.4}{0.3}\right)^2 = 1.$$

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

$$
\begin{array}{ll}
\text{minimize} & t_f \\
\text{subject to} & \ddot{\theta} = -\frac{g}{l}\sin\theta + \frac{u}{l}\cos\theta \\
& \ddot{p} = u \\
& x_p = p - l\sin\theta \\
& y_p = -l\cos\theta \\
& \left(\frac{x_p+0.3}{0.05}\right)^2 + \left(\frac{y_p+0.4}{0.3}\right)^2 \geq 1 \\
& -0.9 \leq p \leq 0.1 \\
& |u| \leq 5 \qquad |\dot{u}| \leq 100 \\
& \theta(t_f) = 0 \qquad \dot{\theta}(t_f) = 0 \\
& p(t_f) = 0 \qquad \dot{p}(t_f) = 0 \\
& \theta(0) = 0 \qquad \dot{\theta}(0) = 0 \\
& p(0) = -0.8 \qquad \dot{p}(0) = 0
\end{array}
\tag{5}
$$

where t_f again is the final time. This is a highly nonconvex problem due to the nonlinear dynamics and, more significantly, due to the obstacle. To solve this problem using the JModelica.org platform, an initial guess needs to be constructed and sent to the solver. An initial guess is created by dividing the optimization problem into two parts. The first part has the same initial condition as the original problem and the terminal point constraint at a position above the obstacle. The second part has the initial condition at the position above the obstacle and the same terminal constraint as the original problem. These trajectories are merged and sent to the solver as initial condition. For more details on how these trajectories were created, see [12]. Optimization results, as well as the trajectory obtained when applying the control action to the physical system with the same initial condition as in the optimization, are found in Figure 3. The optimal control trajectory for the path-constrained problem is found in Figure 4(b).

Discretization of the Optimal Trajectories

The results from optimization problems (4) and (5) are continuous time state and control trajectories which we denote by $p^\star(t), \dot{p}^\star(t), \theta^\star(t)$ and $\dot{\theta}^\star(t)$ respectively. The sampling time of the PC communication is denoted by h and we introduce the sampling counter $n \in \mathbb{N}$. This implies that the actual time t at sampling instant n is $t = hn$. We define the discrete time variables $p^0, \dot{p}^0, \theta^0, \dot{\theta}^0$ and u^0 at the sampling instants as

$$
\begin{aligned}
p^0(n) &:= p^\star(nh), & \dot{p}^0(n) &:= \dot{p}^\star(nh), \\
\theta^0(n) &:= \theta^\star(nh), & \dot{\theta}^0(n) &:= \dot{\theta}^\star(nh), \\
u^0(n) &:= u^\star(nh),
\end{aligned}
$$

for every $n \in \mathbb{N}$ such that $nh \leq t_f$. We also define $x^0(n) = [p^0(n)\ \dot{p}^0(n)\ \theta^0(n)\ \dot{\theta}^0(n)]^T$. Using these definitions, discrete time trajectories are created and used as feedforward trajectories to the pendulum system.

5. Model Predictive Control

The feedforward 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

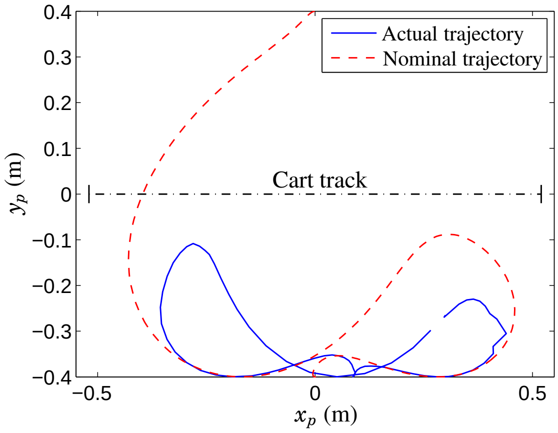


Figure 5 Pendulum end point trajectory for the swing-up problem when the real pendulum is swinging initially and no feedback is used.

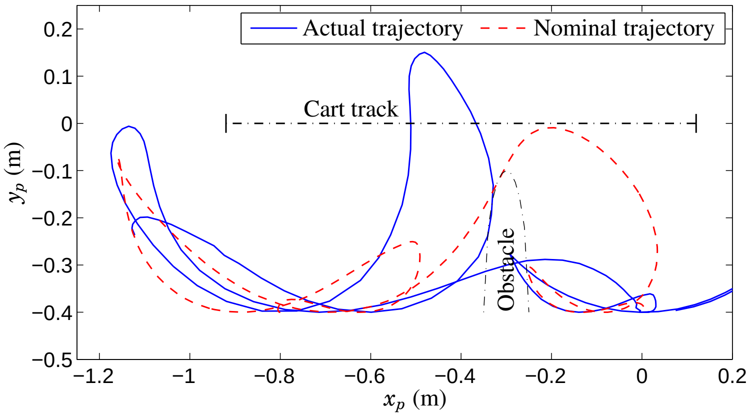


Figure 6 Pendulum end point trajectory for the path-constrained problem when the real pendulum is swinging initially and no feedback is used.

the previous section it is specified that the pendulum should start at rest in the downward position. When disturbances are present in the initial condition of the pendulum, the resulting pendulum end point trajectories are very far from the optimal ones. This is shown in Figure 5 for the swing-up problem and in Figure 6 for the path constrained problem. In the experiments, the pendulum was initially swinging back and forth with a magnitude of approximately 45°. In this section we introduce MPC feedback to cope with such disturbances.

Discrete Time Pendulum Model

The continuous time dynamics of the pendulum system is discretized and linearized to be used for model predictive control. In each sampling instant, n, the system, (2)-(3), is linearized around the nominal states, $x^0(n)$, and the nominal control, $u^0(n)$. We introduce actual states $p(n), \dot{p}(n), \theta(n), \dot{\theta}(n)$ and $x(n) = [p(n)\ \dot{p}(n)\ \theta(n)\ \dot{\theta}(n)]^T$ and the actual control $u(n)$. We also introduce the deviation between the actual states and nominal states

$$\Delta p(n) := p(n) - p^0(n), \qquad \Delta \dot{p}(n) := \dot{p}(n) - \dot{p}^0(n),$$
$$\Delta \theta(n) := \theta(n) - \theta^0(n), \qquad \Delta \dot{\theta}(n) := \dot{\theta}(n) - \dot{\theta}^0(n),$$
$$\Delta x(n) := x(n) - x^0(n), \qquad \Delta u(n) := u(n) - u^0(n).$$

Since the cart dynamics are linear, only the pendulum dynamics need to be linearized. To achieve this, we introduce $\Delta z_\theta(t)$ which is the deviation from the linearization point for the continuous pendulum states and $\Delta v(t)$ which is the continuous control signal for the linearized model. This gives the following linearized pendulum dynamics when linearized around pendulum angle θ^0

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

where $(\omega^0)^2 = \frac{g}{l}\cos\theta^0$. The resulting linear time-varying dynamics depend on the nominal pendulum angle θ^0 only. To obtain a discrete time model for sampling instant n, the linearization is performed around pendulum angle $\theta^0(n)$ and the resulting linear model is discretized using zero-order-hold. The discrete time control signal, which we denote by $\Delta u(n)$ is constant during the sample. The discretized zero-order-hold equations becomes

$$\Delta x_\theta(n+1) = e^{A(\theta^0(n))h}\Delta x_\theta(n) + \int_{s=0}^{h} e^{A(\theta^0(n))(h-s)} B(\theta^0(n)) ds \Delta u(n) \tag{6}$$

where

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

with $(\omega^0(n))^2 = \frac{g}{l}\cos\theta^0(n)$. The last equality comes from the Taylor series expansion of cosine and sine. The integral in (6) becomes

$$\begin{aligned}\int\limits_{s=0}^{h} e^{A(\theta^0(n))(h-s)}B(\theta^0(n))ds &= \int\limits_{s=0}^{h}\begin{pmatrix}\frac{\omega^0(n)}{g}\sin\left(\omega^0(n)(h-s)\right)\\ \frac{\omega^0(n)^2}{g}\cos\left(\omega^0(n)(h-s)\right)\end{pmatrix}ds\\ &= \begin{bmatrix}\frac{1}{g}\cos\left(\omega^0(n)(h-s)\right)\\ -\frac{\omega^0(n)}{g}\sin\left(\omega^0(n)(h-s)\right)\end{bmatrix}_{s=0}^{h}\\ &= \begin{pmatrix}\frac{1}{g}(1-\cos\omega^0(n)h)\\ \frac{\omega^0(n)}{g}\sin\omega^0(n)h\end{pmatrix}.\end{aligned}$$

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

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

where $\Delta x_c = [\Delta p\ \Delta\dot{p}]^T$. This gives the following full linearized model

$$\Delta x(n+1) = \Phi(\theta^0(n))\Delta x(n) + \Gamma(\theta^0(n))\Delta u(n) \tag{7}$$

where

$$\Phi(\theta^0(n)) = \begin{pmatrix}1 & h & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & \cos\omega^0(n)h & \frac{1}{\omega^0(n)}\sin\omega^0(n)h\\ 0 & 0 & -\omega^0(n)\sin\omega^0(n)h & \cos\omega^0(n)h\end{pmatrix}$$

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

and $(\omega^0(n))^2 = \frac{g}{l}\cos\theta^0(n)$.

MPC Optimization Problem

The model developed in the previous section is unstable for some pendulum angles. Due to this, predicting future states directly with (7) may result in poor predictions. To avoid that, a discrete time LQ-feedback term that depends on the nominal pendulum angle is introduced, $u_{\text{fb}}(n) = -L(\theta^0(n))\Delta x(n)$, where $L(\theta^0(n))$ is the optimal LQ-feedback for (7). The prediction model becomes

$$\begin{aligned}\Delta x(n+1) &= \left(\Phi(\theta^0(n)) - \Gamma(\theta^0(n))L(\theta^0(n))\right)\Delta x(n) + \Gamma(\theta^0(n))\Delta u(n)\\ &= \Phi_L(\theta^0(n))\Delta x(n) + \Gamma(\theta^0(n))\Delta u(n)\end{aligned} \tag{8}$$

where $\Phi_L(\theta^0(n)) := \Phi(\theta^0(n)) - \Gamma(\theta^0(n))L(\theta^0(n))$. This model is stable for every nominal pendulum angle $\theta^0(n)$. The decision variables in the MPC problem are state and control signal deviations from the nominal trajectories. In every sample instant, $u(n) = u^0(n) + \Delta u(n) + u_{\text{fb}}(n)$, is sent as control signal (acceleration reference) to the system.

The maximal allowed acceleration is $\pm 7m/s^2$ which is the maximal acceleration for which the inner control loops do not saturate. The track on which the cart is attached is one meter. The track length and control magnitude constraints are captured in the following sample dependent constraint set

$$\begin{aligned} \mathcal{X}(n) = \big\{\Delta u(n) \in \mathbb{R}, \Delta x(n) \in \mathbb{R}^4 \;\big|\; &|\Delta u(n) + u^0(n) - L(\theta^0(n))\Delta x(n)| \leq 7, \\ &\Delta p(n) + p^0(n) \leq 1 - p_0, \\ &\Delta p(n) + p^0(n) \geq -p_0\big\} \end{aligned} \tag{9}$$

where $p_0 \in [0,1]$ is the initial position of the cart on the track. We use a quadratic cost, hence the MPC problem to be solved in each sampling instant, n, is

$$\begin{aligned} \min_{\Delta x, \Delta u} \quad & \sum_{l=n}^{n+N} \Delta x(l)^T Q \Delta x(l) + \Delta u(l)^T R \Delta u(l) \qquad (10) \\ \text{s.t.} \quad & \Delta x(l+1) = \Phi_L(\theta^0(l))\Delta x(l) + \Gamma(\theta^0(l))\Delta u(l), \quad l = n, \ldots, n+N-1, \\ & (x(l), u(l)) \in \mathcal{X}(l), \quad l = n, \ldots, n+N, \\ & \Delta x(n) = \bar{x} \end{aligned}$$

where $Q \succeq 0$ and $R \succ 0$. The optimal $\Delta u(n+N) \equiv 0$ and can hence be removed from the optimization. Since the objective function is quadratic and the dynamics and constraints are linear, the resulting optimization problem is a quadratic program.

Optimization Algorithm

The optimization problem (10) is solved using the algorithm presented in [10]. The algorithm in [10] is an accelerated gradient algorithm that is applied to the dual of a condensed version of (10). A condensed version it obtained by eliminating the state variables by expressing them in the control variables. We present the condensed version of (10) and the optimization algorithm from [10] below. To this end we introduce the following matrices

$$\mathbf{A}(n) := \begin{pmatrix} I \\ \mathbf{A}_1(n) \\ \vdots \\ \mathbf{A}_N(n) \end{pmatrix}, \qquad \mathbf{B}(n) := \begin{pmatrix} 0 & \cdots & 0 \\ \mathbf{B}_{11}(n) & \cdots & \mathbf{B}_{1N}(n) \\ \vdots & \ddots & \vdots \\ \mathbf{B}_{N1}(n) & \cdots & \mathbf{B}_{NN}(n) \end{pmatrix}$$

where

$$\mathbf{A}_i(n) := \prod_{l=n}^{n+i-1} \Phi_L(\theta^0(l)),$$

$$\mathbf{B}_{ij}(n) := \begin{cases} \left[\prod_{l=n+j}^{n+i-1} \Phi_L(\theta^0(l))\right] \Gamma(\theta^0(n+j-1)) & j < i, \\ \Gamma(\theta^0(n+j-1)) & j = i, \\ 0 & j > i. \end{cases}$$

We denote the predicted future state and control variables by

$$\Delta\mathbf{x}(n) = \begin{bmatrix} \Delta x(n) \\ \vdots \\ \Delta x(n+N) \end{bmatrix}, \qquad \Delta\mathbf{u}(n) = \begin{bmatrix} \Delta u(n) \\ \vdots \\ \Delta u(n+N-1) \end{bmatrix}.$$

The predicted future state variables can be expressed in the current state $\Delta x(n) = \bar{x}$ and in the control variables $\Delta\mathbf{u}(n)$ as

$$\Delta\mathbf{x}(n) = \mathbf{A}(n)\bar{x} + \mathbf{B}(n)\Delta\mathbf{u}(n). \tag{11}$$

By introducing the matrices $I_p := [1,0,0,0]$, $C_\mathrm{p} := \mathrm{blkdiag}(I_p,\ldots,I_p)$ and

$$C_\mathrm{L}(n) := \mathrm{blkdiag}(-L(\theta^0(n)),\ldots,-L(\theta^0(n+N-1)))$$

the constraint set (9) for all $n,\ldots,n+N$ can be written as

$$\underbrace{\begin{pmatrix} C_\mathrm{p} \\ -C_\mathrm{p} \\ C_\mathrm{L}(n) \\ -C_\mathrm{L}(n) \end{pmatrix}}_{C_x(n)} \Delta\mathbf{x}(n) + \underbrace{\begin{pmatrix} 0 \\ 0 \\ I \\ -I \end{pmatrix}}_{C_u} \Delta\mathbf{u}(n) \le \underbrace{\begin{pmatrix} \mathbf{1} - \mathbf{p}_0 - \mathbf{p}^0(n) \\ \mathbf{p}_0 + \mathbf{p}^0(n) \\ \mathbf{7} - \mathbf{u}^0(n) \\ \mathbf{7} + \mathbf{u}^0(n) \end{pmatrix}}_{d(n)} \tag{12}$$

where $\mathbf{1} = [1,\ldots,1]^T$, $\mathbf{p}_0 = [p_0,\ldots,p_0]^T$, $\mathbf{7} = [7,\ldots,7]^T$, $\mathbf{p}^0(n) = [p^0(n),\ldots,p^0(n+N)]^T$ and $\mathbf{u}^0(n) = [u^0(n),\ldots,u^0(n+N-1)]^T$. The constraints in (12) can, using the state predictions in (11), be written as

$$(C_x(n)\mathbf{B}(n) + C_u)\,\Delta\mathbf{u}(n) \le d(n) - C_x(n)\mathbf{A}(n)\bar{x}. \tag{13}$$

By further defining

$$\mathbf{Q} := \mathrm{blkdiag}(Q,\ldots,Q), \qquad \mathbf{R} := \mathrm{blkdiag}(R,\ldots,R),$$
$$C(n) := C_x(n)\mathbf{B}(n) + C_u, \qquad g(n,\bar{x}) := d(n) - C_x(n)\mathbf{A}(n)\bar{x}$$

the optimization problem (10) at sample instant n is equivalently written as

$$\begin{aligned} \min_{\Delta\mathbf{u}(n)} \quad & \tfrac{1}{2}\Delta\mathbf{u}(n)^T H(n)\Delta\mathbf{u}(n) + \bar{x}^T G(n)\Delta\mathbf{u}(n) \\ \text{s.t.} \quad & C(n)\Delta\mathbf{u}(n) \le g(n,\bar{x}) \end{aligned} \tag{14}$$

where $H(n) = \mathbf{B}(n)^T\mathbf{Q}\mathbf{B}(n) + \mathbf{R}$ and $G(n) = \mathbf{A}(n)^T\mathbf{Q}\mathbf{B}(n)$. To solve (14) we introduce dual variables $\boldsymbol{\mu} \in \mathbb{R}^p_{\ge 0}$ for the inequality constraints where p is the total number of constraints. We get the following dual problem

$$\max_{\boldsymbol{\mu}\ge 0}\min_{\Delta\mathbf{u}(n)} \frac{1}{2}\Delta\mathbf{u}(n)^T H(n)\Delta\mathbf{u}(n) + \bar{x}^T G(n)\Delta\mathbf{u}(n) + \boldsymbol{\mu}^T(C(n)\Delta\mathbf{u}(n) - g(n,\bar{x})).$$

As shown in [14] the dual problem can explicitly be written as

$$\max_{\boldsymbol{\mu}\ge 0} -\frac{1}{2}(C(n)^T\boldsymbol{\mu} + G(n)^T\bar{x})^T(H(n))^{-1}(C(n)^T\boldsymbol{\mu} + G(n)^T\bar{x}) - \boldsymbol{\mu}^T g(n,\bar{x}). \tag{15}$$

We define the dual function as the maximand in (15) and denote the dual function by $D_N(\bar{x},\boldsymbol{\mu},n)$. The dual function D_N has Lipschitz continuous gradient with Lipschitz constant $L(n) = \|C(n)(H(n))^{-1}C(n)^T\|$ and the gradient is given by

$$\nabla D_N(\bar{x},\boldsymbol{\mu},k) = -C(n)(H(n))^{-1}(C(n)^T\boldsymbol{\mu} + G(n)^T\bar{x}) - g(n,\bar{x}).$$

As shown in [14, 10] this implies that the dual function can be maximized using an accelerated gradient method. The iterations defining the accelerated gradient algorithm applied to the dual problem (15) are stated below (cf. [10]).

$$\Delta\mathbf{u}^k = -(H(n))^{-1}(C(n)^T\boldsymbol{\mu}^k + G(n)^T\bar{x}) \tag{16}$$

$$\Delta\widetilde{\mathbf{u}}^k = \Delta\mathbf{u}^k + \frac{k-1}{k+2}(\Delta\mathbf{u}^k - \Delta\mathbf{u}^{k-1}) \tag{17}$$

$$\boldsymbol{\mu}^{k+1} = \max\left\{0, \boldsymbol{\mu}^k + \frac{k-1}{k+2}(\boldsymbol{\mu}^k - \boldsymbol{\mu}^{k-1}) + \frac{1}{L(n)}\left(C(n)\Delta\widetilde{\mathbf{u}}^k - g(n,\bar{x})\right)\right\} \tag{18}$$

where k is the iteration number. The algorithm converges as $O(1/k^2)$ in both dual function value and in distance between the primal variables $\Delta\mathbf{u}^k$ and the optimal primal variables (cf. [14, 10]). For more on accelerated gradient methods the reader is referred to [15, 16, 17, 14].

Implementational Aspects and Stopping Conditions

The MPC controller is implemented in Matlab/Simulink and communicates with the second microprocessor on the cart via the serial interface. Not all state variables can be measured directly, only cart and pendulum positions are measured. The cart velocity is estimated in the second microprocessor and is sent to the PC when asked for by the MPC controller. The pendulum angular velocity is estimated by a derivative filter in Simulink that is updated ones in every MPC sample. This gives accurate enough pendulum angular velocity estimates since the pendulum dynamics are quite slow. The control horizon is chosen to $N = 40$. The sampling time, which is chosen to $h = 0.025$s, sets hard limitations on the allowed execution time of the MPC controller. In each sampling instant, the matrices $A(n)$, $B(n)$, $C(n)$ and $g(n,\bar{x})$ are built. These matrices are sampling dependent, but can be precomputed and stored for faster online execution. The optimization algorithm (16)-(18) is warm-started in every sample with the solution to the optimization problem in the previous sample, shifted one step, as initial guess. A constraint tightening approach is used to guarantee a feasible solution with finite number of iterations. We use a relative constraint tightening of 0.005, i.e., if the actual constraint is $x \leq 0.5$ the corresponding constraint is set to $x \leq (1 - 0.005)0.5 = 0.4975$ in the optimization problem. The stopping condition of the algorithm is to have primal feasibility, i.e., $x \leq 0.5$ in the example above and a relative duality gap less than 0.005. By construction of the optimization problem, the equality constraints originating from the dynamic equations always hold.

Experimental Results

In Figures 7 and 8 pendulum end point trajectories when feedforward and MPC feedback is used, are plotted. The experiments are initialized with the pendulum swinging back and forth with a magnitude of approximately 45° as in the examples without feedback in Figures 5 and 6. The weight matrices are chosen to be $Q = \text{diag}(50, 0.1, 50, 0.1)$, and $R = 0.3$ in the path-constrained problem and $R = 1$ in the swing-up problem. The feedback gain vector L is the LQ-gain computed using unit weights on both states and control.

Due to the initial swinging of the pendulum, the trajectories are far from the optimal ones at start 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. Figures 9(a) and 9(b) show the control trajectories that are applied to the system for the swing-up problem and the path-constrained problem respectively. Figures 10(a) and 10(b) show the execution time of the MPC controller for the swing-up problem and path-constrained problem respectively. Both total execution time, which include setup of the matrices used by the optimization algorithm and solving the problem, and execution time used by the optimization algorithm are plotted.

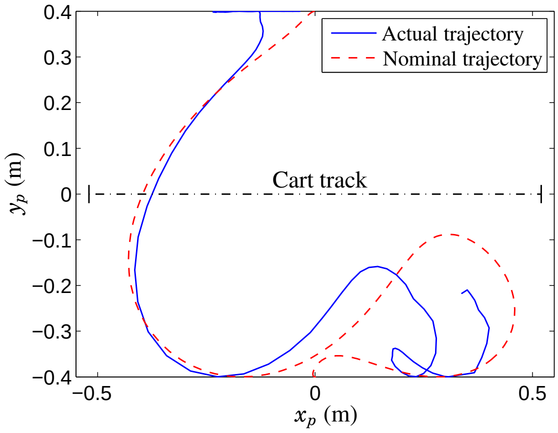


Figure 7 Pendulum end point trajectory for the swing-up problem when the real pendulum is swinging initially and feedback is used.

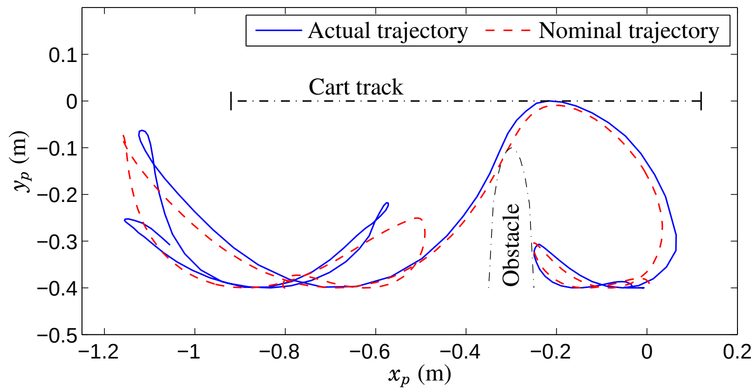
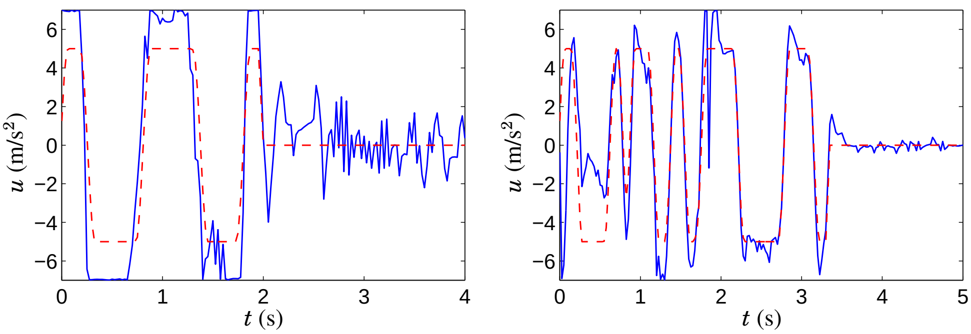


Figure 8 Pendulum end point trajectory for the path-constrained problem when the real pendulum is swinging initially and feedback is used.



(a) (b)

Figure 9 Control trajectory applied to the real system (solid) and feedforward trajectory (dashed) when initial swinging and feedback is used for the swing-up problem (a) and the path-constrained problem (b).

Figures 10(a) and 10(b) show that the optimization algorithm is efficient enough to find a close to optimal solution well within the sampling time of $h = 25$ ms. Videos of similar experiments, with and without initial swinging of the pendulum, can be found in [18].

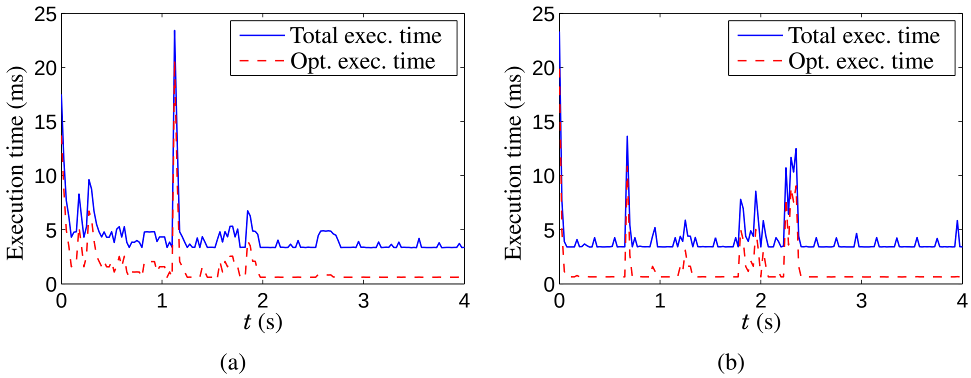


Figure 10 Total (solid) and optimization algorithm (dashed) execution times for the MPC algorithm when initial swinging and feedback is used for the swing-up problem (a) and the path-constrained problem (b).

6. Conclusions

We have developed an MPC controller that controls the actual system trajectories towards precomputed feedforward trajectories in a pendulum system. The feedforward trajectories take the system from one operating point to another. One swing-up problem and one path-constrained problem are considered and both have been applied to a physical pendulum system. The MPC optimization problem is solved using an accelerated gradient method technique presented in [10]. The experiments show that the algorithm is efficient enough for real-time implementation in a pendulum system.

7. References

[1] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, pp. 3–20, Jan. 2002.

[2] A. Bemporad and C. Filippi, "Suboptimal explicit MPC via approximate multiparametric quadratic programming," in *Proceedings of 40th IEEE Conference on Decision and Control*, (Orlando, Florida), pp. 4851–4856, 2001.

[3] T. A. Johansen, "Approximate explicit receding horizon control of constrained nonlinear systems," *Automatica*, vol. 40, no. 2, pp. 293–300, 2004.

[4] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, pp. 267–278, March 2010.

[5] S. Richter, C. Jones, and M. Morari, "Real-time input-constrained MPC using fast gradient methods," in *Proceedings of the 48th Conference on Decision and Control*, (Shanghai, China), pp. 7387–7393, Dec. 2009.

[6] J. M. Maciejowski, *Predictive control with constraints*. Essex, England: Prentice Hall, 2002.

[7] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers and Chemical Engineering*, vol. 23, pp. 667–682, 1999.

[8] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.

[9] E. F. Camacho and C. A. Bordons, *Model Predictive Control in the Process Industry*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.

[10] P. Giselsson, "Execution time certification for gradient-based optimization in model predictive control," in *Proceedings of the 51st IEEE Conference on Decision and Control*, (Maui, HI), 2012. To appear.

[11] P. Giselsson, "Model predictive control in a pendulum system," in *Proceedings of the 31st IASTED Conference on Modelling, Identification and Control*, (Innsbruck, Austria), Feb. 2011.

[12] P. Giselsson, J. Åkesson, and A. Robertsson, "Optimization of a pendulum system using Optimica and Modelica," in *Proceedings of the 7th International Modelica Conference 2009*, (Como, Italy), Sept. 2009.

[13] J. Åkesson, M. Gäfvert, and H. 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), Feb. 2009.

[14] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," *Automatica*, 2012. To appear. Available: `http://www.control.lth.se/Staff/PontusGiselsson.html`.

[15] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course (Applied Optimization)*. Springer Netherlands, 1st ed., 2003.

[16] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, pp. 183–202, Oct. 2009.

[17] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization." Submitted to SIAM J. Optim. Available: `http://www.math.washington.edu/~tseng/papers/apgm.pdf`, May 2008.

[18] P. Giselsson, "Pendulum videos." `http://www.control.lth.se/Staff/PontusGiselsson.html`.