CONTROL OF LABORATORY MODEL OF PENDUBOT

MARTIN LAUKO, PAVOL SEMAN, GERGELY TAKÁCS, BORIS ROHAL'-ILKIV
ÚAMAI, SJF STU BA, Námestie slobody 17, 812 31 Bratislava 1, SK,
tel.: +421 2 572 96 111, fax: +421 2 529 25 749,
mail: martin.lauko@stuba.sk

Abstract: This paper presents the concept of swing up and balancing control of a pendubot laboratory model. The pendubot is a two-link under actuated robotic mechanism, presenting the classic inverted pendulum problem; well suited for control theory education as well as for research in the control of nonlinear mechatronic systems with fast dynamics. Brief introduction into the laboratory hardware and its mathematical modeling is followed by an account on the controller design process. The proposed control scheme is verified in experiment and results are evaluated in the presented work. The swing up of the pendulum is based on a method of impulse build up of energy. A fourth order linearized time-invariant state-space system model is identified for the pendubot system. The basic balancing control is then based on a linear-quadratic (LQ) controller, implemented and tested real-time on the experimental setup. Laboratory verification confirms, that the linear quadratic controller actuated system response is excellent, however the inclusion of process constraints in a model predictive control (MPC) based balancing scheme could potentially offer numerous benefits.

Keywords: inverted pendulum, pendubot, LQ control, pendulum swing up

1 INTRODUCTION

Stability control is one of the basic problems of control theory. This problem is studied on different models, which include the pendubot system. The pendubot is a two-link planar robot with an actuator on the first arm and no actuator on the second arm. It is underactuated and has fast nonlinear dynamics. The objective is to stabilize the system in one of its unstable equilibrium positions. That means keep the second arm, the free pendulum, upright and the first arm in a desired position. We began with a already build physical model of pendubot. We will show a brief derivation of the motion equations, using the Lagrange energy balance method, and the resulting state space model.

2 PENDUBOT MODEL

There are different ways how to realize the construction of pendubot. Our approach is based on the work of Mates Mates and Seman [2009].The physical model is a combined model of pendubot and furuta pendulum. We will in this paper discus only pendubot control. This design uses a servo motor to rotate the arm. This has the benefit that, we can directly control the torque applied to the arm and also brake at any given time. The brake is useful to apply braking moment in impulse control.

2.1 Model hardware

The Figure 1 shows connections between hardware components. As can be seen the control model is designed on the Host PC. It is equipped with Matlab / Simulink software. The model is compiled and transfered to the Target PC, where it is run in a simplified environment in real time. Communication with hardware is done through the I/O card in Target PC. As can be seen on Figure 1 only the pendulum angle is read directly by the I/O card. Communication with the servo motor is done through the control unit. The control unit translates the encoder signal to arm angle and controls the motor power to achieve the desired torque.

The physical model is made of these components:

- Servo motor - Mitsubishi HC-KFS43
- Control unit - Mitsubishi MR-J2S-40A

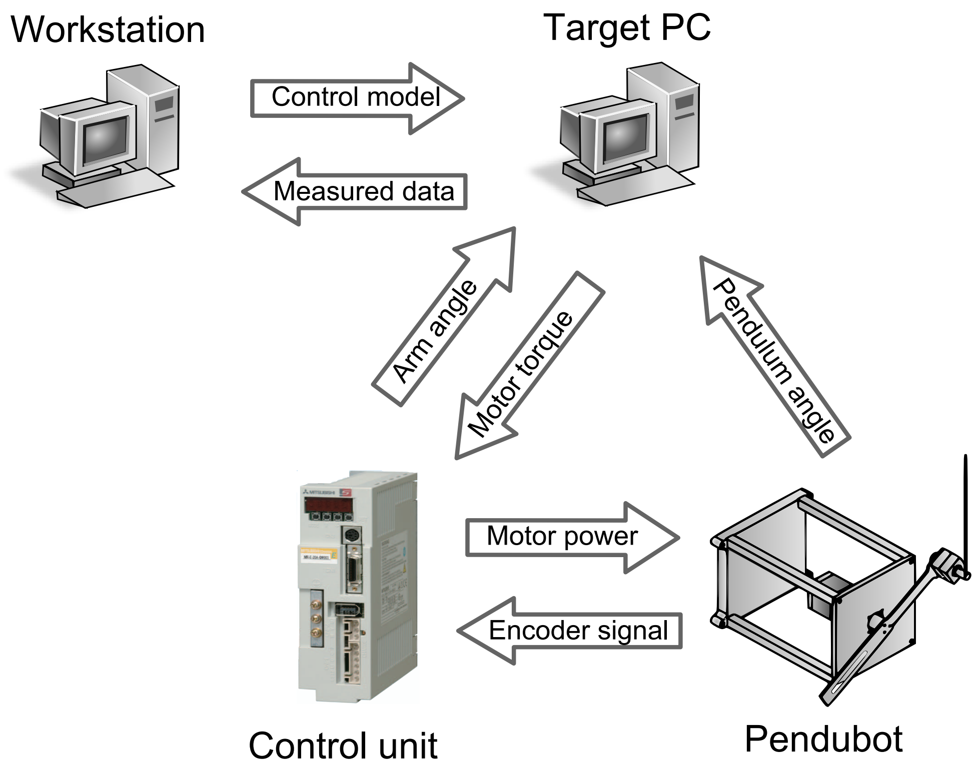


Figure 1 – Hardware connection scheme

- Incremental rotary sensor - OMRON E6B2
- I/O Card - Humusoft MF-624

2.2 Mathematical model

The pendubot is a second order dynamic system. We will use the energy balance method, Lagrange equation, to derive the equations of dynamics.

In this section mathematical equations use the following symbols:

- mr - Weight of arm
- l1 - Length of arm
- lg1 - Distance from center of gravity of the arm to the axis of rotation
- k1 - Friction coefficient in arm joint
- Ir - Mass moment of inertia of the arm
- mk - Weight of pendulum
- lg2 - Distance from center of gravity of the pendulum to the axis of rotation
- Ik - Mass moment of inertia of the pendulum
- k2 - Friction coefficient in pendulum joint

- φ - Pendulum angle
- θ - Arm angle

The basic form Lagrange equations is:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \left(\frac{\partial L}{\partial q_i}\right) = Q_i \tag{1}$$

where L, the Lagrangian, is the difference of kinetic and potentian energy. For the pendubot system these are simple to derive. Details, how to do this, are in different publications Fantoni and Lozano [2001]. The end result from eq.(1) are two equations (2) and (3).

$$\tau = \ddot{\varphi}\left(I_r + m_k l_1^2\right) + \ddot{\theta} m_k l_1 l_{g2} \cos(\varphi - \theta) + m_k l_1 l_{g2} \sin(\varphi - \theta) + \left(m_r l_{g1} + m_k l_1\right) g \cos\varphi \tag{2}$$

$$0 = \ddot{\varphi} m_k l_1 l_{g2} \cos(\varphi - \theta) + \ddot{\theta}(I_k + m_k l_{g2}^2) - m_k l_1 l_{g2} \dot{\varphi}^2 \sin(\varphi - \theta) + m_k g l_{g2} \cos\theta \tag{3}$$

Linearizing these equations around chosen point and solving the result, we get the state space model $\dot{x} = Ax + Bu$. Where x is the state vector $x = [\varphi \quad \theta \quad \dot{\varphi} \quad \dot{\theta}]^T$. Choosing the point $\varphi = 0$ and $\theta = 0$ as the up-up position, the resulting state space matrices A and B for our model are:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 10,714 & -7,162 & -2,424 & 0 \\ -15,755 & 43,320 & 0 & -0,027 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 40,033 \\ -58,873 \end{bmatrix} \tag{4}$$

3 CONTROL DESIGN

To solve the pendubot control problem, it is best to divide it into two parts. The balancing control that will keep the pendulum upright and the swing-up control that will bring the pendulum from complete stop, in the lower position, to the unstable equilibrium position. Switching between these two controls depends on the pendulum and arm position. This is necessary so that the balancing control is active only around the equilibrium point.

3.1 LQ control

The balancing control is realized using a LQ controller, which was derived using the state-space model from eq.(4). The control law for LQ is $u = -Kx$ where x is the state vector and K is the LQ gain matrix. The gain was calculated by minimizing the cost function

$$J = \int (x'Qx + u'Ru)dt \tag{5}$$

where the weight matrices Q and R have been chosen as

$$Q = \begin{bmatrix} 19 & 0 & 0 & 0 \\ 0 & 26 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R = [3,5] \tag{6}$$

The resulting LQ control gain matrix is

$$K = [-1,54 - 7,11 - 0,69 - 1,22] \tag{7}$$

3.2 Swing-up control

There exist different approaches to inverted pendulum swing-up. Predominantly, they are based on adding energy to the pendulum system until it equals the potential energy of the pendulum in the upper position (Albahkali et al. [2009]). Our approach is based on the same idea. Adding of energy is done by swinging the arm around zero position. The arm swings are limited to help smoothen transition to balancing control. The limits should allow the arm to move as widely as possible, to allow more energy build up per swing. For the calculation of torque the the following equation was used:

$$u = sat_n \left(k(E - E_0) sign(\dot{\theta} \cos \theta) \right) \tag{8}$$

where E is the current energy of the system and E_0 is the desired energy. The gain constant K determines speed at which energy is added to the system (Astrom and Furuta [1996]).

4 TEST RESULTS

The functionality of the designed control scheme was tested on our physical model. The measured results are in Fig. 4. The swing-up algorithm was active the first 5 seconds, after which it switched to balancing control, because the pendulum was close to the upper equilibrium position. In Fig. 2(a) can be seen the swinging motion of the pendulum and its stabilization. Speed of the pendulum is in Fig. 2(b). The position and speed of the arm can be seen in Fig. 2(c) and 2(d). The corresponding torque applied to the arm is seen in Fig. 2(e).

5 MPC CONTROL SIMULATION

This section presents the planned model based predictive control (MPC) of the pendubot system through simulations performed using the state-space model of the laboratory hardware. The preliminary considerations introduced here give a basis for further work on the pendubot system and also point out the weaknesses of saturated linear-quadratic control.

All simulations assume identical models for the pendubot system, including an initial condition equivalent to displacing the pendulum 5 degrees away from its upright position. The pendulum is assumed to be placed around its nominal upright position, while the swing-up portion of the control assignment is ignored here.

The constraints placed on the system are applied only to the torque requirement passed onto the stepper motor. The model input is in fact the force moment in N/m applied to the pendulum arm. The MPC model includes this constraint, while the simple LQ version is saturated to these bounds. System sampling is set to $Ts = 0.01$ seconds in all cases.

All simulation cases utilize the linear, time-invariant state-space model as introduced in (4), penalizations set on the saturated LQ controller and in the MPC controller are according to (6).

5.1 Unconstrained system

The MPC method considered in this paper is a dual-mode constrained controller with guaranteed stability [Mayne et al., 2000; Chen and Allgöver, 1998; Rossiter, 2003], providing the maximal admissible region of attraction and target set through high order polyhedral constraints.

Given a five degree initial condition, the MPC controller requires a prediction horizon $n_c = 10$ steps. This places the initial condition just on the edge of the admissible set for all three presented simulation cases.

In the first simulation case, the constraints slightly exceed the expected moments provided to the stepper motor. Here a $0.06N/m$ constraint practically produces an unconstrained response from the system. The torque profile for this case is shown on Figure 3, where the responses produced by the

MPC and saturated LQ controllers are identical[1]. Subject to the same input signal, the system behaves identically therefore the individual states are not shown here.

5.2 Constrained system

The following two simulation examples involve constraints placed on the torque requirement passed to the actuator. In the first case a stable response is produced, however the benefits of using an MPC controller over LQ are clearly demonstrated. The second simulation involves a more limitive constraint, rendering the saturated LQ controlled system entirely unstable.

5.2.1 Stable

Figure 4 shows the individual pendulum states and the torque requirement, when the system is subject to a $\pm 0.05N/m$ moment constraint. As it is visible from the arm and pendulum position and velocity diagrams on Figure 4(a)-(d), the saturated LQ controller stabilizes the pendulum less efficiently. While the control course remains stable, the constrained MPC governed output settles the pendulum arm position much faster. The constraints and the controller outputs may be observed on Figure 4(e), where it is evident that the LQ controller leaves the torque input on its lower saturation limit for a longer period than its MPC counterpart.

5.2.2 Unstable

The previous subsection dealt with a constraint control case, where it has been shown that an MPC controlled pendulum arm may bring benefits over simple saturated LQ control. However efficiency is not the only issue here: enforcing more stringent constraints on the stepper motor may render LQ control entirely unstable, while MPC can still function without the hazards of de-stabilizing the control system.

In this case the constraints were decreased slightly, to $\pm 0.0485N/m$, while the rest of the simulation settings remained the same. As it is clearly visible on the position, velocity and torque output of the system on Figure 5; the system became unstable under LQ control. For the case of saturated LQ control, Figures 5(a),(b) show the arm and pendulum positions increasing indefinitely - causing the pendulum to fall from its unstable equilibrium position. Similarly Figures 5(c),(d) shows the LQ controlled velocities to grow uncontrollably.

However the MPC controlled responses still remain within reasonable bounds, preserving the stability of the feedback system.

5.3 Implementing the MPC controller

While the previous simulation cases point out numerous advantages of using an MPC controller with guaranteed stability on the pendubot, there are some issues with the practical implementation of such a system on the laboratory device in real time. The expected sampling period is not extremely short and higher order prediction models have been used before with sampling periods of $0.0002Hz$ [Wills et al., 2008], though without stability guarantees. The MPC control of an unstable equilibrium position requires stability guarantees, which places additional requirements on algorithm efficiency.

5.3.1 On-Line Quadratic Programming

It is possible that a controller with a fourth order prediction model and a $n_c = 10$ steps prediction horizon is implementable without significant issues on a real-time rapid software prototyping system. However during simulation stages it has been noted, that if a maximal admissible reachable and target set is considered, the constraints have to be evaluated much further: requiring constraint checking horizons in the excess of 200 steps. Adding more complexity to the problem leaves and open question, whether this system can be implemented using quadratic programming solvers optimized for MPC usage such as presented in Ferreau [2006] and Ferreau et al. [2008].

[1]Because the two responses are in fact identical, only the MPC controlled torque profile is visible.

The given system with the considered settings also caused various numerical problems; both at the stage of searching for the largest admissible set and during the simulation of on-line quadratic optimization. Linear programming used at the initialization stage exited with a warning; stating that the optimization process iteration tolerance limit has been exceeded. At on-line optimization, the quadratic programming solver[2] issued a warning about a non-symmetric optimization Hessian.

5.3.2 Multi-Parametric Programming

The pendubot device is a mechatronic system with fast dynamics, as such it requires an efficient MPC implementation. A viable alternative to on-line quadratic programming optimization is the use of multi-parametric programming (MP) based MPC. Assuming a piece-wise affine linear MPC problem, the controller pre-computes controller regions and associates them with polyhedral regions in state-space. This way the computational load is transferred to the off-line mode, while the regions and control laws corresponding to the current states are found directly in a very efficient manner [Kvasnica et al., 2006].

With this approach is very promising, the off-line computational load grows very rapidly when increasing prediction horizons. A multi-parametric controller has been evaluated for the given pendubot system, assuming the same settings as introduced for the QP controller. The explicit controller has been calculated using the Multi-Parametric Toolbox [Kvasnica et al., 2004].

By definition, the controller outputs are the same as the direct QP optimization results up to numerical precision, therefore these results are not indicated in the simulations presented in 5.1 and 5.2. However the resulting controller has been computed in more than 37 hours, using a generic personal computer conforming to current standards[3]. The explicit MPC controller is defined over 132927 region sover 4D, resulting an exported C language header over 97 Mbytes in file size. While this is certainly not prohibitive for personal computers, a digital signal processing board (DSP) or similar hardware with limited amount of RAM could not run such a large application. It remains an open question, whether the large number of regions could cause the search times to exceed sampling during the on-line control of the pendubot arm. While this is unlikely, we have to note that if a larger range of expected deviations from the upper unstable position is required also a larger region of attraction is necessary. This can be ensured by increasing the prediction horizon, which certainly will also exponentially increase the number of regions and controller size [Takács and Rohal'-Ilkiv, 2009].

It is also worth noting, that the multi-parametric controller computation procedure was not without numerical errors. During the lengthy computation time, the algorithm issued warnings involving the linear programming solver. Although the final controller passes the invariance check and behaves as expected during simulations, it is not certain whether these errors could produce erroneous controller outputs.

6 CONCLUSION

Measurements are in correspondence with our expected results. The designed swing-up needed several swings to bring the pendulum to the upright position. This was caused by the limitations to movement of the arm and maximum allowed torque. The balancing LQ controller has oscillations around the equilibrium position, but they were negligibly small.

The simulations performed utilizing the MPC controller clearly show the drawbacks of saturated linear-quadratic control. In the absence of constraints, the LQ and MPC controllers provide identical outputs to the actuator. However a constrained torque request results in a sub-optimal LQ control course when compared to MPC, furthermore a more stringent torque boundary may result in the loss of stability. The preliminary controller implementation analysis presented in this paper points out several difficult aspects of applying MPC on the laboratory device in real time.

[2] The solver assumed throughout the simulation was "quadprog", default QP solver in the Matlab suite.
[3] AMD Athlon X2 DualCore 4400+ at 2.00GHz, 2.93GBytes of RAM.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support granted by the Slovak Research and Development Agency under the contracts APVV-0280-06 and APVV-0160-07, and reviewers' comments. This research is also supported by the grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is co-financed from the state budget of the Slovak Republic.

References

ALBAHKALI, T.; MUKHERJEE, R.; DAS, T. 2009. Swing-up control of the pendubot: An impulse-momentum approach. *IEEE Transactions on Robotics*, 25, 4, 975–982.

ASTROM, K. J.; FURUTA, K. 1996. Swinging up a pendulum by energy control. In *IFAC 13th World Congress*. San Francisco, California.

CHEN, H.; ALLGÖVER, F. 1998. Ca quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34, 1205–1217.

FANTONI, I.; LOZANO, R. 2001. *Non-linear Control for Underactuated Mechanical Systems*. Springer. ISBN 978-1852334239, 295 pp.

FERREAU, H. 2006. *An Online Active Set Strategy for Fast Solution of Parametric Quadratic Programs with Applications to Predictive Engine Control*. Master's thesis, University of Heidelberg.

FERREAU, H.; BOCK, H.; DIEHL, M. 2008. An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, 18, 8, 816–830.

KVASNICA, M.; GRIEDER, P.; BAOTIĆ, M. 2004. Multi-Parametric Toolbox (MPT). Online. Available: `http://control.ee.ethz.ch/`.

KVASNICA, M.; GRIEDER, P.; BAOTIC, M.; CHRISTOPHERSEN, F. J. 2006. *Multi-Parametric Toolbox (MPT)*. Extended documentation.

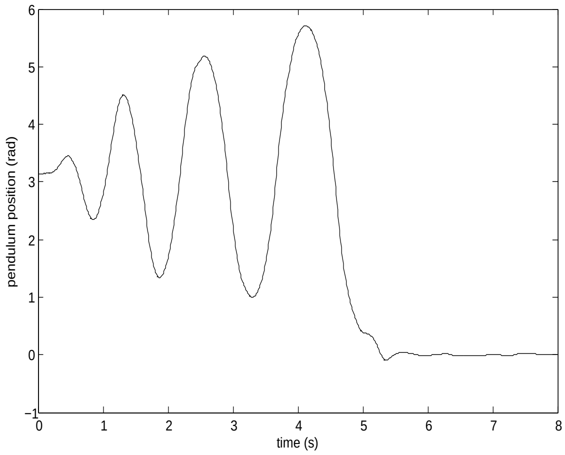
MATES, M.; SEMAN, P. 2009. Combined model of pendubot and rotational inverted pendulum. In *IWCIT '09 : Proceedings of 8th International PhD Student's Workshop on Control and Information Technology*. Brno: Brno University of Technology.

MAYNE, D. Q.; RAWLINGS, J. B.; RAO, C. V.; SCOKAERT, P. O. M. 2000. Constrained model predictive control: Stability and optimality. *Automatica*, 36, 789–814.

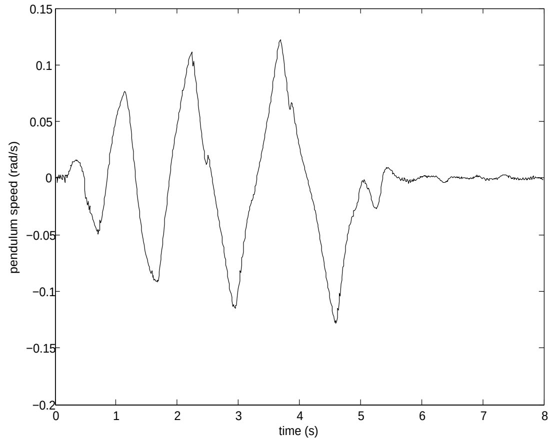
ROSSITER, J. A. 2003. *Model-based Predictive Control: A Practical Approach*. Taylor and Francis (CRC Press). ISBN 08-4931-2914.

TAKÁCS, G.; ROHAL'-ILKIV, B. 2009. MPC with guaranteed stability and constraint feasibility on flexible vibrating active structures: a comparative study. In *Proceedings of the eleventh IASTED interantional conference: Control Applications*, 278–285. Cambridge, United Kingdom.

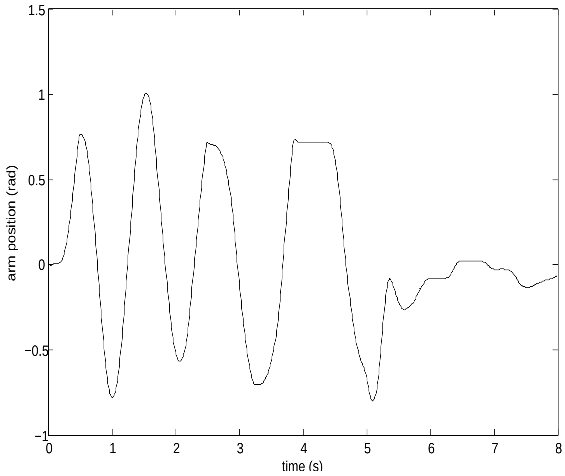
WILLS, A. G.; BATES, D.; FLEMING, A. J.; NINNESS, B.; MOHEIMANI, S. O. R. 2008. Model predictive control applied to constraint handling in active noise and vibration control. *IEEE Transactions on Control Systems Technology*, 16, 1, 3–12.



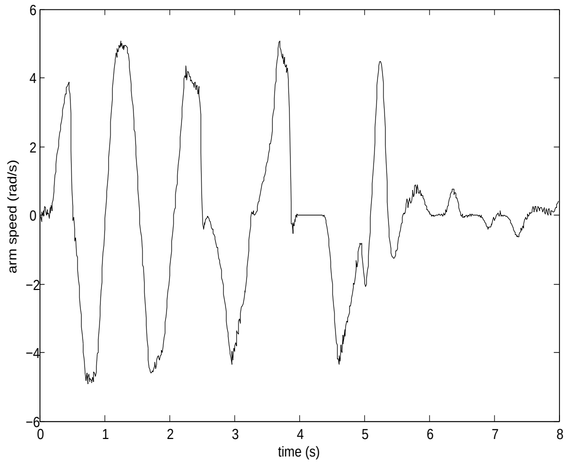
(a) Pendulum position



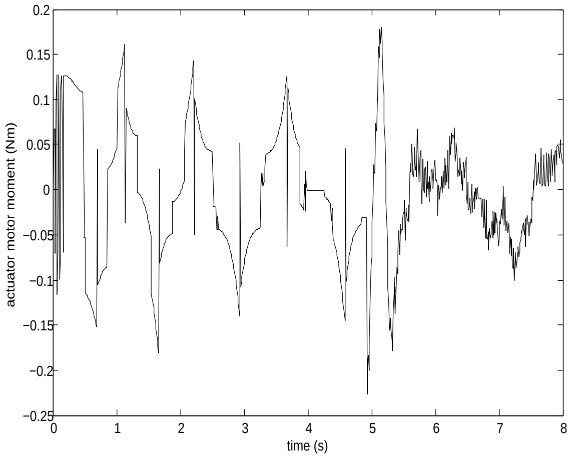
(b) Pendulum speed



(c) Arm position



(d) Arm speed



(e) Motor torque

Figure 2 – Measured results of pendubot swing-up and balancing LQ

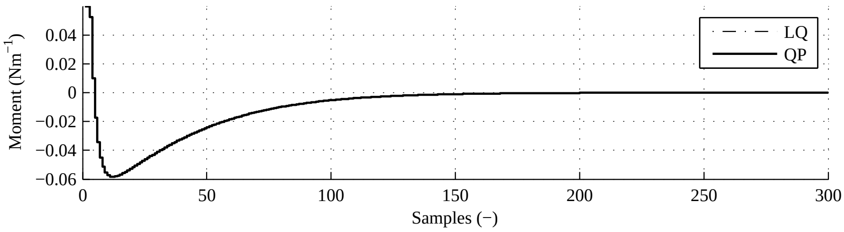
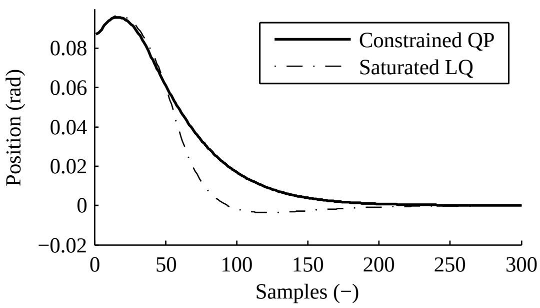
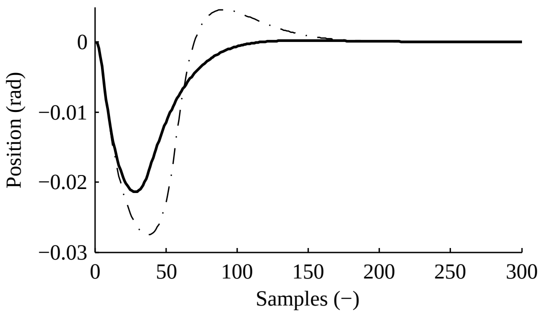


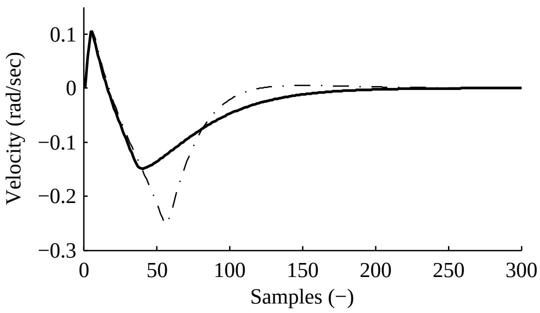
Figure 3 – Torque request to actuator



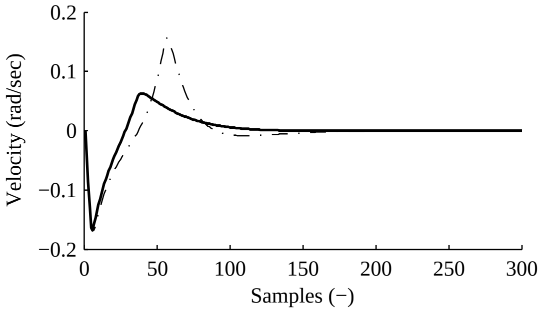
(a) Arm position



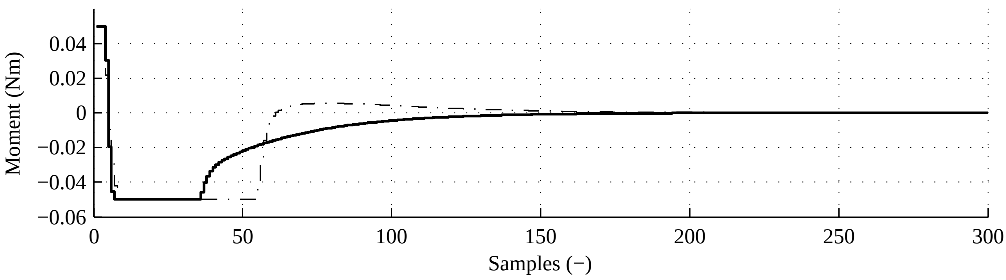
(b) Pendulum position



(c) Arm speed

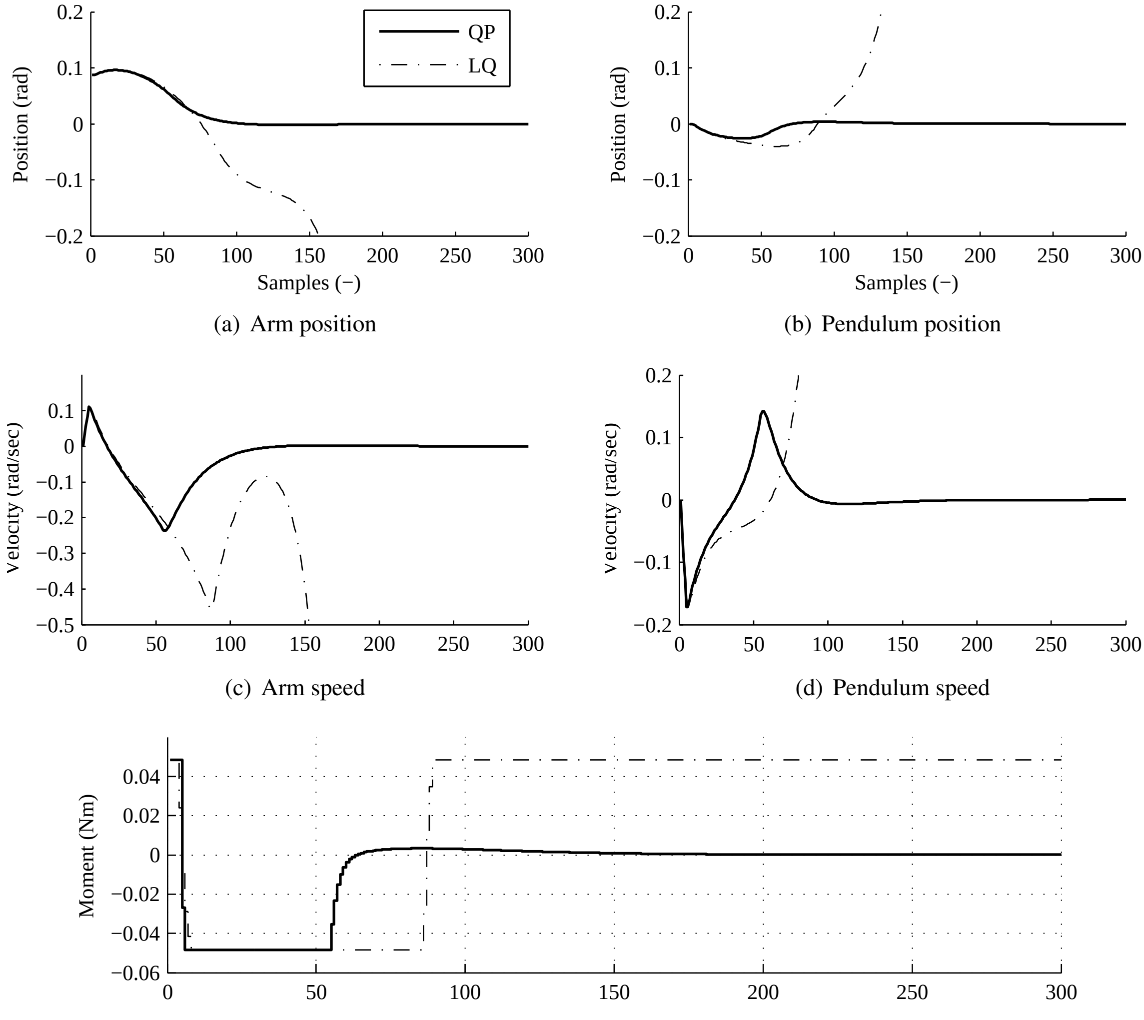


(d) Pendulum speed



(e) Torque request to actuator

Figure 4 – Stable simulation, comparing LQ and MPC based actuator inputs. Arm and pendulum position in radians is visible on (a) and (b), while (c), (d) shows the corresponding arm and pendulum velocities. Figure (e) denotes the torque requirement passed onto the actuator, when its output is limited to $\pm 0.05 N/m$



(a) Arm position

(b) Pendulum position

(c) Arm speed

(d) Pendulum speed

(e) Torque request to actuator

Figure 5 – Unstable controller behavior, comparing LQ and MPC based actuator inputs. Arm and pendulum position in radians is visible on (a) and (b), while (c), (d) shows the corresponding arm and pendulum velocities. Figure (e) denotes the torque requirement passed onto the actuator, when its output is limited to $\pm 0.0485 N/m$