Swinging Up and Stabilization Control of the Furuta Pendulum using Model Predictive Path Integral Control

Hannes Homburger, Stefan Wirtensohn and Johannes Reuter

Abstract—This paper presents the swinging up and stabilization control of a Furuta pendulum using the recently published nonlinear Model Predictive Path Integral (MPPI) approach. This algorithm is based on a path integral over stochastic trajectories and can be parallelized easily. The controller parameters are tuned offline regarding the nonlinear system dynamics and simulations. Constraints in terms of state and input are taken into account in the cost function. The presented approach sequentially computes an optimal control sequence that minimizes this optimal control problem online. The control strategy has been tested in full-scale experiments using a pendulum prototype. The investigated MPPI controller has demonstrated excellent performance in simulation for the swinging up and stabilizing task. In this paper, the determination of the controller parameters for the MPPI algorithm is described in detail. Further, a discussion treats the advantages of the nonlinear MPPI control. Videos are available under the nonlinear Model Predictive Path Integral (MPPI) approach. This algorithm is based on path integrals is presented in [10]. This algorithm is characterized by its ability to handle stochastic input disturbances, nonsmooth cost functions and to be parallelized without compromising accuracy. The question of how this novel approach can be used to swing up and stabilize the Furuta pendulum and what is achieved by its application is addressed in this paper. In section II, the basics of path integrals and the MPPI approach are described. In section III, the equations of motions according to [11] and a cost function are presented, which are combined to an optimal control problem (OCP). This is followed by determining the controller parameters in section IV. In section V an optional LQR is derived. While section VI presents the results of the MPPI control in simulation, section VII presents the results of the MPPI control in full-scale experiments. Finally, section VIII summarizes the results of this paper and presents ideas for future work.

I. INTRODUCTION

The ability to autonomously perform difficult tasks with nonlinear systems leads to many new possibilities in various fields of engineering and society. Scientific and technical progress in the context of control algorithms and computing power enables complex behavior in demanding control tasks. Recently, the swinging up and stabilization control of the rotating, underactuated and inverted pendulum known as Furuta Pendulum has become a benchmark for nonlinear control. Eponymous, it was first described by Furuta et al. [1]. Different nonlinear control approaches such as sliding mode control [2], trajectory tracking using feedback linearization [3], input-output feedback linearization [4], active disturbance rejection control (ADRC) [5], fuzzy control [6], energy shaping [7], model predictive control (MPC) [8] and nonlinear model predictive control (NMPC) using the MATMPC library [9] can be applied to swing up and stabilize the Furuta pendulum. The classical approaches are easy to implement and computationally cheap but need a lot of energy and are dependent on a precalculated trajectory. The NMPC approach using the MATMPC library mentioned above leads to good behavior in simulation but is very computationally expensive, is sensitive to stochastic disturbances and difficult to parallelize. A new type of NMPC algorithm called MPPI

\[ x_{t+1} = F(x_t, v_t), \]

where \( x_t \in \mathbb{R}^m \) with \( v_t \sim \mathcal{N}(u_t, \Sigma) \) denotes the commanded control vector \( u_t \in \mathbb{R}^m \) with additive white Gaussian noise (AWGN) with covariance \( \Sigma \in \mathbb{R}^{m \times m} \) at the discrete-time instance \( t \). The distribution of the uncontrolled system denoted by \( \mathbb{P} \) with zero input \( u_t = 0 \), \( t = 0, 1, ..., T - 1 \) leads to the probability density function (PDF)

\[ q(V) = \prod_{t=0}^{T-1} \frac{1}{(2\pi)^{m/2} |\Sigma|^{1/2}} \exp \left( -\frac{1}{2} v_t^T \Sigma^{-1} v_t \right), \]

where \( V = \{v_0, v_1, ..., v_{T-1}\} \) denotes the sequence of actual input values. The distribution of the controlled system denoted by \( \mathbb{P} \) with an open-loop sequence of manipulated variables \( u_t = u_t^0, t = 0, 1, ..., T - 1 \) leads to the PDF

\[ p(V) = \prod_{t=0}^{T-1} \frac{1}{(2\pi)^{m/2} |\Sigma|^{1/2}} \exp \left( -\frac{1}{2} (v_t - u_t^0)^T \Sigma^{-1} (v_t - u_t^0) \right). \]

\[ (3) \]
An initial state $x_0$ and a realized sequence of input values $V$ can be uniquely assigned to a trajectory using the recursive application of the system dynamics (1). To evaluate a commanded control sequence $U = \{u_1, u_2, ..., u_{T-1}\}$, a cost function

$$J(U, x) = \mathbb{E}_p \left[ \phi(x_T) + \sum_{t=0}^{T-1} C(x_t) + u_t^T R u_t \right], \tag{4}$$

is defined, where $\phi(\cdot)$ reflects the terminal cost, $C(\cdot)$ the instantaneous state-dependent cost, and $R = \lambda \Sigma^{-1}$ a coefficient matrix with parameter $\lambda \in \mathbb{R}^+$, that can be interpreted as temperature. Note that this choice of $R$ is necessary to make the stochastic Hamilton-Jacobi-Bellman equation linear, which is a basic condition for its solution by path integrals. For detailed information, the reader is referred to [12], [13] and [14]. Introducing the cumulative state-dependent path costs $S(V) = \phi(x_{n+T}) + \sum_{t=0}^{T-1} C(x_t)$, the value function

$$V(x) = \min_U J(U, x) = \min_U \left[ \mathbb{D}_{KL}(p(V)||q(V)) + \frac{1}{\lambda} \mathbb{E}_p[S(V)] \right],$$

$$\text{(5a)}$$

$$= \min_{p(V)} \left[ \mathbb{D}_{KL}(p(V)||q(V)) + \frac{1}{\lambda} \mathbb{E}_p[S(V)] + \mathcal{L} \left( \int p(V) dV - 1 \right) \right] \tag{5b}$$

can be represented by PDFs using a Lagrange multiplier denoted by $\mathcal{L}$ which considers the constraint $\int p(V) dV = 1$. Note that the quadratic input costs are represented by the Kullback-Leibler divergence denoted by $\mathbb{D}_{KL}$ of the PDFs of the uncontrolled and the controlled system [14]. Solving this minimization leads to the optimal distribution

$$p^*(V) = \frac{1}{\eta} \exp \left( -\frac{1}{\lambda} S(V) \right) q(V), \tag{6}$$

where $\eta$ denotes a normalization factor. An approach using Jensen’s inequality and importance sampling, according to [10] leads to the optimal control sequence

$$u^*_t = \int_{\Omega_V} p(V) \frac{p^*(V) q(V)}{q(V)} v_t dV \tag{7a}$$

$$= \mathbb{E}_p [\omega(V)v_t] \tag{7b}$$

minimizing (4), where $\Omega_V$ denotes an image of the sample space and the importance weighting

$$\omega(V) = \frac{1}{\eta} \exp \left( -\frac{1}{\lambda} S(V) + \sum_{t=0}^{T-1} \frac{1}{2} u_t^T \Sigma^{-1} u_t - v_t^T \Sigma^{-1} u_t \right). \tag{8}$$

can be calculated using the PDFs (2), (3) and (6). Using Monte Carlo simulation, (7a) can be estimated through the iterative update law

$$u^*_{t \text{new}} = u^*_{t \text{old}} + \frac{N}{N} \omega(V_n)(v_t^* - u^*_{t \text{old}}), \tag{9}$$

where $N$ samples were drawn from the system with the commanded control sequence $U^\text{cmd} = \{u_0^\text{cmd}, u_1^\text{cmd}, ..., u_{T-1}^\text{cmd}\}$.

III. SYSTEM DYNAMICS, COST FUNCTION AND PROBLEM FORMULATION

The MPPI algorithm solving an OCP considering general nonlinear system dynamics (1) and a cost function (4). These parts are presented in this section.

A. Dynamics of the Furuta Pendulum

A detailed derivation of the equations of motion of the Furuta pendulum both, using the Euler-Lagrange approach and using the Newton approach can be found in [9]. Nevertheless, a short description of the modeling is given in the following since it is an elementary component of the choice of controller parameters. Further, the identified parameters are presented. According to [11], the state vector

$$x = (\theta_1 \ \theta_2 \ \omega_1 \ \omega_2)^T \tag{10}$$

is a combination of the two orientation angles $\theta_1, \theta_2$ shown in Fig. 1 and the associated angular velocities $\omega_1, \omega_2$. While $\theta_2$ is not constrained, $\theta_1$ is constrained in the range $\theta_{1,\text{min}} \leq \theta_1 \leq \theta_{1,\text{max}}$. The dynamic model of the Furuta pendulum, according to [11], is given by

$$x = f(x, u) = (\omega_1 \ \omega_2 \ \dot{\theta}_1 \ \dot{\theta}_2)^T, \tag{11}$$

where the angular accelerations $\dot{\theta}_1$ and $\dot{\theta}_2$ depend nonlinearly on the system state $x$ and the input vector $u$. The angular accelerations are given by

$$\ddot{\theta}_1 = -\frac{J_2 b_1}{m_2 L_1^2} \cos(\theta_2) b_2 - \frac{J_2}{m_2 L_1^2} \sin(2\theta_2) \tag{12}$$

$$+ \left( \begin{array}{c} \omega_1 \omega_2 \\ \omega_1^2 \\ \omega_2^2 \\ \omega_1 \omega_1 \omega_2 \end{array} \right)^T$$

$$\dot{\omega}_1 = \left( \begin{array}{c} -J_2 b_1 \\ m_2 L_1^2 \cos(\theta_2) b_2 \\ -\frac{J_2}{m_2 L_1^2} \sin(2\theta_2) \\ -\frac{1}{2} J_2 m_2 L_1^2 \cos(\theta_2) \sin(2\theta_2) \\ J_2 m_2 L_1^2 \sin(\theta_2) \\ -m_2 L_1^2 \cos(\theta_2) \\ \frac{1}{m_2 L_1^2} \sin(2\theta_2) \\ g \end{array} \right)$$

where $J_2$, $b_1$, and $b_2$ are the moments of inertia, the gravitational constant, and the mass of the masses, respectively.
and \( \dot{\theta}_2 = \)

\[
\begin{pmatrix}
  m_2 L_1 L_2 \cos(\theta_2) b_1 \\
  -b_2 (J_0 + J_2 \sin^2(\theta_2)) \\
  m_2 L_1 L_2 \dot{J}_2 \cos(\theta_2) \sin(2\theta_2) \\
  -\frac{1}{2} \sin(2\theta_2) (J_0 \dot{J}_2 + J_2 \sin^2(\theta_2)) \\
  -\frac{1}{2} m_2^2 L_1^2 \dot{\theta}_2 \sin(2\theta_2)
\end{pmatrix}^T \begin{pmatrix}
  \omega_1 \\
  \omega_2 \\
  \omega_3 \\
  \omega_4 \\
  \omega_5
\end{pmatrix} + \begin{pmatrix}
  -m_2 L_1 \cos(\theta_2) \\
  J_0 + J_2 \sin^2(\theta_2) \\
  -m_2 L_2 \sin(\theta_2) (J_0 + J_2 \sin^2(\theta_2))
\end{pmatrix}^T \begin{pmatrix}
  \tau_1 \\
  \tau_2 \\
  g
\end{pmatrix}
\]

(13)

where \( m_2 \) denotes the mass of the front arm, \( b_1 \) denotes the viscous coefficient of friction in the motor bearing, \( b_2 \) denotes the viscous coefficient of friction in pin coupling, \( J_0 \) denotes the effective moment of inertia at the motor and \( J_2 \) denotes the effective moment of inertia at pin coupling. The lengths \( L_1 \) and \( L_2 \) are shown in Fig. 1. The input vector \( u = (\tau_1, \tau_2, g)^T \)

(14)

contains the torques in the directions \( \theta_1 \) and \( \theta_2 \) denoted by \( \tau_1 \) and \( \tau_2 \) and the gravitational acceleration denoted by \( g \). While \( \tau_1 \) is controllable by motor torque, \( \tau_2 \) represents uncontrollable friction and \( g \) is an uncontrollable constant. The torque \( \tau_1 \) is generated by a DC motor in an underlying control loop. It is assumed that its dynamics is fast compared to (11) and can be neglected. However, due to the motor specifications, the torques are bounded in the ranges \( |\tau_1| < \tau_{1,\text{max}} \) and \( |\tau_2| < \tau_{2,\text{max}} \).

B. Cost Function

The state-dependent cost function \( C(x) \) is used to evaluate the quality of the system behavior mathematically. For example, it can be determined by formalizing linguistic quality criteria mathematically or inverse reinforcement learning [15]. To swing up the Furuta Pendulum the equilibrium point \( x_\pi = (\pi, \pi, 0, 0)^T \) has to minimize the cost function. As widely used in the literature, error coordinates are used, which yields \( \bar{x} = x - x_\pi = (\bar{\theta}_1, \bar{\theta}_2, \bar{\omega}_1, \bar{\omega}_2)^T \). Although it can be chosen arbitrarily, a locally convex cost function leads to faster convergence of the algorithm. Moreover, it should be periodic with \( 2\pi \) in \( \theta_2 \) to reflect the modulo property of this state. The state-dependent cost function

\[
l(x) = c_1 (\cos(\bar{\theta}_1) - 1)^2 + c_2 (\cos(\bar{\theta}_2) - 1)^2 + c_3 \bar{\omega}_1^2 + c_4 \bar{\omega}_2^2,
\]

(15)

\[
\begin{array}{|c|c|c|}
\hline
\text{Parameter} & \text{Value} & \text{Parameter} & \text{Value} \\
\hline
\theta_{1,\text{max}} & \pi/\text{rad} & b_2 & 493 \times 10^{-3} \text{Nm/rad} \\
\theta_{2,\text{max}} & 11\pi/\text{rad} & J_2 & 2.7 \times 10^{-3} \text{kgm}^2 \\
m_2 & 0.192 \text{kg} & J_0 & 2.5 \times 10^{-3} \text{kgm}^2 \\
L_1 & 103.5 \text{mm} & g & 9.81 \text{m/s}^2 \\
L_2 & 95.5 \text{mm} & \tau_{1,\text{max}} & 0.45 \text{Nm} \\
b_1 & 2.7 \times 10^{-3} \text{Nms/rad} & \tau_{2,\text{max}} & 0 \text{Nm} \\
\hline
\end{array}
\]

TABLE I \( \left( \begin{array}{cccc}
\theta_{1,\text{max}} & \pi/\text{rad} & b_2 & 493 \times 10^{-3} \text{Nm/rad} \\
\theta_{2,\text{max}} & 11\pi/\text{rad} & J_2 & 2.7 \times 10^{-3} \text{kgm}^2 \\
m_2 & 0.192 \text{kg} & J_0 & 2.5 \times 10^{-3} \text{kgm}^2 \\
L_1 & 103.5 \text{mm} & g & 9.81 \text{m/s}^2 \\
L_2 & 95.5 \text{mm} & \tau_{1,\text{max}} & 0.45 \text{Nm} \\
b_1 & 2.7 \times 10^{-3} \text{Nms/rad} & \tau_{2,\text{max}} & 0 \text{Nm} \\
\end{array} \right) \)

where \( c_1, c_2, c_3, c_4 \in \mathbb{R}^+ \) denote the weighting coefficients, fulfills the two defined criteria.

C. Equality and Inequality Constraints

To consider the inequality constraints of \( \theta_1 \), the indicator function

\[
\zeta(\bar{\theta}_1) = \begin{cases} 
1 & \text{if } \bar{\theta}_1 < -\pi + \theta_{1,\text{min}} \vee \bar{\theta}_1 > -\pi + \theta_{1,\text{max}} \\
0 & \text{else}
\end{cases}
\]

(16)

is defined. This indicator function is used to define a scalar equality constraint

\[
g(\bar{\theta}_1) = \zeta(\bar{\theta}_1) = 0.
\]

(17)

Furthermore, the function

\[
h(\tau_1) = (|\tau_1| - |\tau_{1,\text{max}}|) \leq 0
\]

(18)

considers the input’s inequality constraint. In this section, a cost function (15) and equality (17) and inequality (18) constraints are defined to swing up the Furuta pendulum. These are modified in the next part into the form (4) required for the MPPI approach.

D. Resulting Problem Formulation

The task of swinging up and stabilizing the Furuta pendulum can be represented as the general OCP

\[
\min_{u} J(u, x) = \min_{u} \int_{t_0}^{t_0 + T} l(t, x(t), u(t))dt
\]

(19a)

s.t. \( x(t) = f(x(t), u(t)), T > 0, x(t_0) = x_0 \),

(19b)

\[
g(x(t)) = 0, \forall t \in [t_0, t_0 + T], \ (19c)
\]

\[
h(x(t), u(t)) \leq 0, \forall t \in [t_0, t_0 + T]. \ (19d)
\]

where \( (19a) \) minimizes the cost function (15), (19b) represents the pendulum’s dynamics (11), (19c) denotes equality constraints in form of the state equations and (19d) enables to include the state and input inequality constraints. Using the MPPI approach, the time-discrete cost function (4) is minimized by an optimal control sequence \( U^* = \{u_0, u_1, ..., u_{T-1}\} \) subject to the stochastic nonlinear system.
TABLE II

| Variance $\sigma_2^2$ | $T_{LQR}$ | $\max. |\tau|$ | Costs | Comment |
|----------------------|-----------|----------------|-------|---------|
| 0.001Nm$^2$/s       | 2.11s     | 0.21Nm        | $2.48 \times 10^5$ | slow   |
| 0.005Nm$^2$/s       | 1.37s     | 0.27Nm        | $1.09 \times 10^5$ | medium |
| 0.01Nm$^2$/s        | 0.92s     | 0.39Nm        | $7.85 \times 10^4$ | fast   |
| 0.05Nm$^2$/s        | 0.91s     | 0.52Nm        | $1.23 \times 10^5$ | too much torque |

The computing effort of the optimization problem, as well as the influence of the inaccuracies of the model, increase with the rising MPC horizon denoted by $T_{MPC}$. However, a sufficiently large prediction horizon is essential for evaluating the result of the choice of control values. An approach for finding a suitable prediction horizon is using an estimation of the time to swing up. For example, an energy-based approach according to [7] needs 2.25s to swing up the given Furuta pendulum. Consequently, the MPC horizon is chosen as $T_{MPC} = 750ms$ than one-third of this time. Thus, a prediction horizon of $T = 10$ time steps is chosen.

D. Covariance matrix of AWGN

In path integral control, as described by [12], the input variables are implicitly penalized quadratically by choice of the covariance matrix of the AWGN. However, uncontrollable states are not excited by noise. In the given application only $\tau_1$ is a controllable input. Thus, the covariance matrix is given by a scalar value

$$ \Sigma = \sigma_2^2, \quad (21) $$

which is both, responsible for the exploration of the state-space and implicitly penalizes the input quadratically. The effect of different variances is shown in Table II, where $T_{LQR}$ denotes the time needed to transfer the system’s state in the catching area of the linear quadratic regulator (LQR) developed in the next section. Costs denote the cumulated costs of a simulation run. Due to minimal torque and best stabilization behavior $\sigma_2^2 = 0.001Nm^2/s$ is chosen. In the following, this parameter set is called slow. For fastest swinging up while meeting $\tau_1(t) \leq \tau_{max}$ $\sigma_2^2 = 0.01Nm^2/s$ is chosen. In the following, this parameter set is called fast. A comparison of the explored state-space using the different parameter sets is shown in Fig. 3.

E. Temperature

The temperature $\lambda$ scales the quadratic input costs with

$$ R = \lambda \Sigma^{-1} \quad (22) $$

according to [12]. Therefore, the relation between $R$ and $C(\cdot)$ has a significant impact on the resulting system behavior. However, $C(\cdot)$ has already been selected freely as well as being scaled. Thus, $\lambda = 1$ has been chosen.

F. Number of Predicted Trajectories

Both, the required computing time and the quality of estimating the expected value by the mean increase with the number of predicted trajectories used. A buffer of 5% is targeted to make the real-time capability of the controller robust to changes in the complexity of the system model and/or the cost function. Results from simulation show that using the parameters selected, at least 30 trajectories are required to achieve excellent system behavior. Using the controller board DS1104 by dSPACE, the possibility to sample 40 trajectories every 75ms was experimentally confirmed.
V. LINEAR QUADRATIC REGULATOR

To improve the system’s behavior around the desired equilibrium point \( x = 0 \) a LQR can be designed. As a basis for this, the nonlinear system dynamics is linearized and can be approximated as

\[
\dot{x} = Ax + b\tau,
\]

in accordance with [11]. The quadratic cost function

\[
J_{LQR}(\tau, x_0) = \int_0^\infty x(t)^TQx(t) + R\tau(t)^2dt,
\]

with state-dependent costs \( Q = \text{diag}\{c_1, c_2, c_3, c_4\} \) according to (15) and setting inputs cost coefficient regarding (22) is minimized using the algebraic Ricatti equation

\[
PA + A^TP - PBR^{-1}B^TP + Q = 0.
\]

This leads to the optimal feedback control

\[
\tau(t) = -Kx(t), \quad K = R^{-1}b^TP = (k_1 \ k_2 \ k_3 \ k_4),
\]

which can be applied in the state-space area around the equilibrium point \( x = 0 \). This area can be expressed by

\[
\frac{\bar{\theta}^2}{r_\theta^2} + \frac{\bar{\omega}^2}{r_\omega^2} \leq 1,
\]

under assumption \( \bar{\omega} \approx 0 \), where \( r_\theta \) and \( r_\omega \) denote elliptic scaling factors. In Figure 4 a representative area determined by simulation is shown.

VI. SIMULATION RESULTS

Before a full-scale experiment is presented in section VII, in this section the MPPI controller with the parameters specified in the previous section is tested in simulation. For this purpose, the MPPI controller and the dynamic model are embedded in a simulation environment. The coefficients of the state-dependent cost function (20) are defined as\( c_1 = 50, \ c_2 = 500, \ c_3 = 1, \ c_4 = 1 \) and \( c_5 = 50000 \). Controller performance is examined using swinging up scenarios with parameter set slow and parameter set fast, defined in section IV. The initial state of the simulation is given by \( x_0 = (0 \ -\pi \ 0 \ 0) \), which corresponds to a pendulum oriented vertically downwards. Using both parameter sets the Furuta pendulum is able to swing up. Further, it remains in the immediate vicinity of the equilibrium point \( x_0 = 0 \). A comparison of the maximum of the desired torque \( |\tau_{1,des}| \), the time \( T_{LQR} \) to reach the catching area (27) specified with \( r_\theta = 0.25\text{rad/s} \) and \( r_\omega = 4\text{rad/s} \) and the normalized and cumulative costs \( C_{norm} \) are given in Table III. The optimal sequence of \( \tau_1 \) is shown in Fig. 6. Figure 7 shows the resulting trajectories. Note, for optimal clarity the module property of \( \bar{\theta}_2 \) is exploited in Fig. 7. Compared to the result applying parameter set fast, applying parameter set slow leads to a slower \( \bar{\omega}_2 \) and thus higher \( T_{LQR} \), but also less oscillatory in the ellipse around the equilibrium point \( x_0 = 0 \), defined in (27). To prevent oscillations an LQR can

![Fig. 5. Architecture of the control components in full-scale experiment](image)

![Fig. 6. Optimal torque sequences of simulations and full-scale experiments and \( T_{LQR} \) (dashed lines)](image)

| TABLE III |
| Variance | \( \max.\ |\tau_1| \) | \( J_{LQR} \) | \( C_{norm} \) |
|----------|-----------------|-------------|-------------|
| Parameter set fast simulation | 0.39Nm | 0.92s | 100% |
| Parameter set fast full-scale | 0.38Nm | 1.18s | 105% |
| Parameter set slow simulation | 0.21Nm | 2.11s | 136% |
| Parameter set slow full-scale | 0.26Nm | 2.76s | 181% |
be applied. The excellent control performance in simulation is validated by full-scale experiments in the next section.

VII. EXPERIMENTAL RESULTS

To test the performance of the MPPI controller in a full-scale experiment, it is applied to swing up a real Furuta pendulum. For this purpose, the MPPI controller is implemented on the rapid control prototyping board DS1104 by dSPACE. Two encoders provide the orientations $\theta_1$ and $\theta_2$. The angular speeds $\dot{\theta}_1$ and $\dot{\theta}_2$ are estimated by numerically differentiate $\theta_1$ and $\theta_2$. Based on this provided data, the MPPI controller approximates the path integral over the weighted predicted stochastic trajectories and thus calculates the desired torque $\tau_{1,\text{des}}$. The actual torque $\tau_{1,\text{act}}$ is controlled implicitly by controlling the actual current $I$. Therefore, a subordinated closed-loop system using a PI controller with the manipulated variable voltage $V$ is implemented. A sampling interval of 1ms for this subordinated control loop is chosen. An overview of the used architecture is shown in Fig. 5. The initial state of the system is given by $x_0 = (0 \ 0 \ 0)$ same as in simulation. A representative swinging up performance is selected from various tests for both parameter set slow and parameter set fast. Despite the influence of real disturbance torques due to unmodelled effects, swinging up the Furuta pendulum is possible. However, since only 40 predicted trajectories are sampled to explore the state-space and due to the influence of unmodelled effects, applying an LQR is required for good performance. While Fig. 6 shows the optimal torque sequence, the resulting trajectories are shown in Fig. 7. The resulting criteria evaluating the performance in full-scale experiments are listed in Table III.

VIII. SUMMARY AND FUTURE WORK

In this paper an OCP representing swing up a Furuta pendulum including nonlinear dynamic equations and equality and inequality constraints is developed. This OCP is solved using the MPPI algorithm, according to [10]. The selection of parameters of the MPPI algorithm is described in detail. The algorithm is tested in a simulation environment, where the controlled system shows excellent performance both, for swinging up and stabilization of the Furuta pendulum. A full-scale experiment shows that the MPPI controller can be used for swinging up the Furuta pendulum. Compared to other approaches previously mentioned, no precalculation of the desired trajectory is required. Concerning other NMPC approaches, the MPPI control algorithm is characterized by a very good parallelizability. In future work, it would be interesting to improve the model accuracy using a neural network approach. Thus, the restoring force of the encoder cable and other state-dependent effects can be included in the prediction.

ACKNOWLEDGMENT

Special thanks to Gabriel Kaddik for determining the parameters of the modeled system dynamics as part of his bachelor thesis, which are listed in Table I.

REFERENCES