Nonlinear control of a Furuta rotary inverted pendulum

Nonlinear control of a Furuta Rotary inverted pendulum

C.J.A. van Kats

DCT Report no: 2004.69
June 2004

TU/e Bachelor Final Project Report

Coach:	Dr. Ir. L. Moreau
Supervisor:	Prof. Dr. Ir. H. Nijmeijer

Eindhoven University of Technology
Department of Mechanical Engineering
Division Dynamical Systems Design
Dynamics and Control Group

Eindhoven, June 24, 2004

Abstract

Inverted pendulums have been widely used by scientists to illustrate the ideas of control technology. In the beginning these were ideas in linear control. Because of their nonlinear behaviour, nowadays they are also used to illustrate new ideas in nonlinear control. An example of an inverted pendulum is the Furuta pendulum. This machine has two rotational degrees of freedom and only one actuator and is thus an under-actuated system. Balancing the pendulum in the vertical unstable equilibrium position requires continuous correction by a control mechanism. In this report a nonlinear control design procedure is discussed based on *input-output linearization.* After this, a *forwarding* procedure is presented to achieve asymptotic stability for the rotational velocity of the actuated arm.

Contents

Chapter 1

Introduction

Systems that have more degrees of freedom than actuators are called underactuated systems. A Furuta pendulum, with its two rotations and one motor, is an example of such a system.

Since most dynamic systems are nonlinear, a lot of research is done on nonlinear control. The goal of this project is to design a nonlinear controller for the Furuta pendulum. It has to bring the pendulum rod vertical upright position and balance it there. Once the pendulum rod is positioned, the rotation of the actuated arm has to be stopped. The position of the arm then converges to a random position.

A description of the pendulum set-up will be given in chapter 2. Before the design process can begin, a mathematical model of the system is needed. The used model and assumptions that are made in it will be discussed. (chapter 2). The model was taken from the literature, so a derivation of it is not part of this project. Knowing the systems' equations the controller design process can begin (chapter 3). First a control strategy has to be formulated. This strategy will consist of two parts. Two different controllers will be used for be used for the swing-up and balancing phase. Switching between the two at the right time will position the pendulum and stabilize it. The design of the balancing controller will be based on the theory of input-output linearization. After the pendulum has been stabilized, the balancing control law will be extended in order to stop the rotation of the horizontal arm. The theory of forwarding will be used to design this action. After simulations have produced proper results, the controllers will be implemented on a real Furuta robot.

Chapter 2

The Furuta pendulum

2.1 Pendulum set-up

In figure 2.1 a picture of the used Furuta pendulum robot is shown. Figure 2.2 shows a schematic representation of this set-up. The set-up consists of a rod, which is connected to a horizontal arm. The horizontal rotation of the arm (α) is actuated by an electromotor. The pendulum rod is not actuated. Since the system has two degrees of freedom and only one actuator, it is called an under-actuated system. The arm has length l_0 and has a moment of inertia around its centre of mass J_{z0}. The length between the joint and the centre of mass of is called l_1 and the rod has moment of inertia J_{z1}.

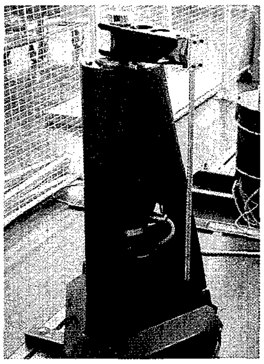

Figure 2.1: The furuta pendulum

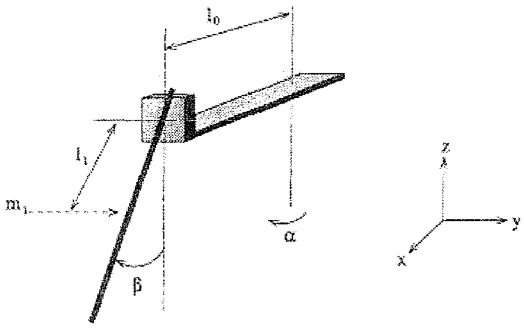

Figure 2.2: The Furuta pendulum schematically

The robot in the lab is equipped with a TUeDACS/1 QAD data-acquisition module, which is operated via Wintarget. This combination makes it possible to build MATLAB SIMULINK files for real-time control. Rotations α and β are measured by optical encoders. Velocities are computed numerically by differentiation.

2.2 The model

The derivation of mathematical equations that were used can be found in [Arno03]. Model parameter values are taken from [Hame04] and can be found in Appendix A. Motor dynamics have been neglected and thus removed from the model in literature. After doing so, the equations of motion can be written as:

$$\begin{bmatrix} (J_{z0}+m_1 l_0^2 \sin^2 \beta)\ddot{\alpha}+m_1 l_1^2 \dot{\alpha}\dot{\beta}\sin\beta\cos\beta+m_1 l_0 l_1 \cos\beta\ddot{\beta}-m_1 l_0 l_1 \sin\beta\dot{\beta}+C_0\dot{\alpha}+F_f \\ (J_{z1}+m_1 l_1^2)\ddot{\beta}+m_1 l_0 l_1 \ddot{\alpha}\cos\beta-m_1 l_1^2 \dot{\alpha}^2 \sin\beta\cos\beta+m_1 g l_1 \sin\beta+C_1\dot{\beta} \end{bmatrix}=\begin{bmatrix} u \\ 0 \end{bmatrix} \tag{2.1}$$

Herein u is the engine torque. The equations take into account viscous friction in the motor axis $C_0\dot{\alpha}$ andin the pendulum joint $C_1\dot{\beta}$. For the rotation of the arm, dry friction is included in the form of a simplified symmetric Coulomb friction model:

$$F_f = K_f \cdot sign(\dot{\alpha}) \tag{2.2}$$

When performing numerical simulations the discontinuous *sign* function will be approximated by a smoothened function:

$$F_f = K_f \cdot \sigma(\dot{\alpha}) = K_f \cdot \left(1-\frac{2}{e^{2k\dot{\alpha}}+1}\right) \tag{2.3}$$

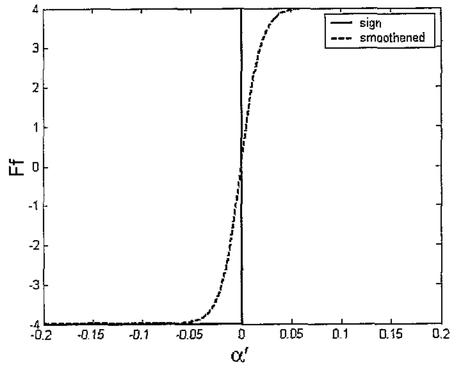

Figure 2.3: Modelled Coulombs friction

In equation (2.3) a large value for k is used (i.e. 50) in order to have a reliable approximation of the *sign* function. In a real-time implementation, the friction model (2.2) will be used for the purpose of friction compensation. A plot of the discontinuous Coulomb friction model and its smoothened version are shown in figure 2.3. The model of the Furuta pendulum may be written as follows:

$$M\begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{bmatrix}+D\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix}+K=U \tag{2.4}$$

with

$$M = \begin{bmatrix} J_{z0} + m_1 l_0^2 + m_1 l_1^2 \sin^2 \beta & m_1 l_0 l_1 \cos \beta \\ m_1 l_0 l_1 \cos \beta & J_{z1} + m_1 l_1^2 \end{bmatrix}$$

$$D = \begin{bmatrix} C_0 + m_1 l_1^2 \dot{\beta} \sin \beta \cos \beta & m_1 l_1^2 \dot{\alpha} \sin \beta \cos \beta - m_1 l_0 l_1 \dot{\beta} \sin \beta \\ - m_1 l_1^2 \dot{\alpha} \sin \beta \cos \beta & C_1 \end{bmatrix}$$

$$K = \begin{bmatrix} K_f \cdot \sigma(\dot{\alpha}) \\ m_1 g l_1 \sin \beta \end{bmatrix}$$

$$U = \begin{bmatrix} u \\ 0 \end{bmatrix}$$

Notice that the model (2.4) is nonlinear since the matrices M, D an K are state dependent.

Chapter 3

Controller design

3.1 Strategy

The goal of the assignment is to design a non-linear controller that is able to stabilize the pendulum in the upright position ($\beta = \pi$). After analysing of the control equations can be concluded that the system is uncontrollable when the pendulum is near horizontal position ($\beta = -\frac{1}{2}\pi, \frac{1}{2}\pi, 1\frac{1}{2}\pi,..$). What causes this is explained in section 3.3.1. Because of this shortcoming the pendulum will have to pass through this position before our stabilizing controller can work. A swing-up controller will therefore be designed first, which brings the pendulum into the upper half circle before the stabilizing controller will stabilize it in the upright position. Switching between the two controllers is performed at an adjustable position β. Figure 3.1 shows a graphical representation of the control strategy.

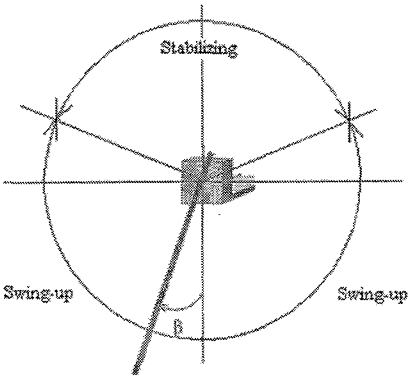

Figure 3.1: Schematic representation of the control strategy

3.2 Swing-up controller

There are different possibilities for bringing the pendulum into the upper half round ($cos(\beta)$< 0). The method that will be used is based on energy control. A complete description of this method can be found in [Åstr00]. The energy of he uncontrolled pendulum rod without the rotating arm can be written as:

$$E = \frac{1}{2}\left(J_{z1} + m_1 l_1^2\right)\dot{\beta}^2 - m_1 g l_1 \left(\cos\beta + 1\right) \tag{3.1}$$

The energy is defined such that it is zero in upright rest position $(\beta = \pi), (\dot{\beta} = 0)$ and $-2m_1gl_1$ in downward rest position $(\beta = 0), (\dot{\beta} = 0)$. The pendulum has to be driven on a trajectory where the energy is zero. This can be achieved by controlling the acceleration of the pendulum joint with control law:

$$s = ng\, sign(E)\, \dot{\beta} \cos\beta \tag{3.2}$$

The number of swings that are needed for the swing-up depends on the value of n For a pendulum on a cart, this control law would be exact, because the theory assumes a two dimensional motion. The dynamics of the Furuta pendulum are somewhat more complicated because the pendulum joint describes a circular trajectory instead of a straight line. The goal of this project is not to build an accurate swing-up controller. If it is able to bring up the pendulum, then this meets our demands. Therefore (3.2) will be applied directly on the Furuta pendulum.

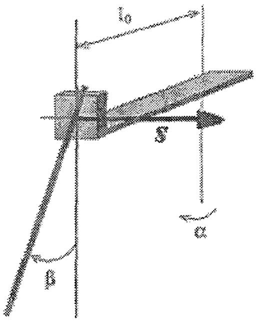

Figure 3.2: Orientation of acceleration s

The desired acceleration of the joint has to be generated by the actuator. As can be seen in figure 3.2, the desired angular acceleration of the arm becomes:

$$\ddot{\alpha} = \frac{s}{l_0} = \frac{1}{l_0} ng\, sign(E)\, \dot{\beta} \cos\beta \tag{3.3}$$

Since it is not possible to give accelerations as input for the actuator, control signal v has to be converted to motor torque. This is done by an input-output linearizing feedback law for u, such that the resulting dynamics of α become:

$$\ddot{\alpha} = v \tag{3.4}$$

Where equation (3.3) will be substituted for v .The theory of input-output linearization can be found in [Khal96]. The MATLAB M-file that computes the implemented input-output linearizing feedback law for the swing-up controller, can be found in Appendix B.

3.3 Stabilizing controller

Now it is possible to bring the pendulum towards upright position, a controller that stabilizes it in this position has to be designed. Applying linear control theory on a linearized system is a possible method. This has been done many times before and it has proved to work. By linearization, system dynamics are approximated around a certain working point. The goal of this project is to use nonlinear control theory, which does not make these approximations.

3.3.1 Stabilizing the pendulum rod

The stabilizing controller is based on the theory of feedback input-output linearization, which is described in [Khal96]. For now the goal is to control the rotation of the rod. Therefore the chosen output is $y = \beta$. The system model (2.4) can be rewritten in the state space form

$$\frac{d}{dt}\begin{bmatrix}\alpha \\ \dot{\alpha} \\ \beta \\ \dot{\beta}\end{bmatrix} = \begin{bmatrix}\dot{\alpha} \\ \frac{1}{det(M)}\left(a+\left(J_{z1}+m_1 l_1^2\right)u-b\right) \\ \dot{\beta} \\ \frac{1}{det(M)}\left(c-\left(m_1 l_0 l_1 \cos\beta\right)u+d\right)\end{bmatrix} \quad , \qquad y=\beta \qquad (3.5)$$

where

$$\begin{aligned} a &= -\left(J_{z1}+m_1 l_1^2\right)\left(C_0\dot{\alpha}+2m_1 l_1^2\dot{\alpha}\dot{\beta}\sin\beta\cos\beta - m_1 l_0 l_1\dot{\beta}^2\sin\beta + K_f\sigma(\dot{\alpha})\right) \\ b &= m_1 l_0 l_1\cos\beta\left(m_1 l_1\dot{\alpha}^2\sin\beta\cos\beta - C_1\dot{\beta} - m_1 g l_1\sin\beta\right) \\ c &= m_1 l_0 l_1\cos\beta\left(C_0\dot{\alpha} + 2m_1 l_1^2\dot{\alpha}\dot{\beta}\sin\beta\cos\beta - m_1 l_0 l_1\dot{\beta}^2\sin\beta + K_f\sigma(\dot{\alpha})\right) \\ d &= \left(J_{z0}+m_1 l_0^2+m_1 l_1^2\sin^2\beta\right)\left(m_1 l_1\dot{\alpha}^2\sin\beta\cos\beta - C_1\dot{\beta} - m_1 g l_1\sin\beta\right) \end{aligned}$$

When evaluating the time derivative of the output y, the control input u appears for the first time in its second derivative

$$\ddot{y} = \frac{1}{det(M)}\left(c-\left(m_1 l_0 l_1\cos\beta\right)u+d\right) \qquad (3.6)$$

The system thus has relative degree two, provides $\cos\beta \neq 0$. An input-output linearizing feedback law can now be formulated of the form:

$$u = \frac{det(M)\upsilon - c - d}{-m_1 l_0 l_1\cos(\beta)} \quad , \qquad \beta \neq \ldots -\frac{\pi}{2},\frac{\pi}{2},\frac{3\pi}{2},\ldots \qquad (3.7)$$

Feedback law (3.7) yields linear input-output dynamics

$$\ddot{y} = \upsilon$$

An M-file that computes feedback law (3.7) using the MATLAB symbolic toolbox can be found in Appendix B. The aim is to achieve that y converges to the reference angle $y_d = \pi$. This can be done by choosing a PD controller:

$$\upsilon = PD(\beta, \dot{\beta}) = -a_0(\beta - \pi) - a_1\dot{\beta} \qquad , \qquad a_0, a_1 > 0 \quad (3.8)$$

Because of the cosine term in the denominator of (3.7), the control signal is defined in every position of the rod except for the horizontal. After a swing-up just above the horizontal, it should theoretically be able to stabilize the rod in upright position. Now the subsystem

$$\frac{d}{dt}\begin{bmatrix}\beta\\ \dot{\beta}\end{bmatrix} = \begin{bmatrix}\dot{\beta}\\ \upsilon\end{bmatrix} \qquad (3.9)$$

has been stabilized by means of (3.8), the zero dynamics for the system (3.5) can be determined. By substitution of $\beta = \pi$, $\dot{\beta} = 0$ and $\ddot{\beta} = \upsilon = 0$ into (3.5) and (3.7):

$$\begin{aligned} a &= -\left(J_{z1} + m_1 l_1^2\right)\left(C_0\dot{\alpha} + K_f\sigma(\dot{\alpha})\right)\\ b &= \ 0\\ c &= \ m_1 l_0 l_1\left(C_0\dot{\alpha} \ + K_f\sigma(\dot{\alpha})\right)\\ d &= \ 0 \end{aligned}$$

which yields the system:

$$\frac{d}{dt}\begin{bmatrix}\alpha\\ \dot{\alpha}\\ \beta\\ \dot{\beta}\end{bmatrix} = \begin{bmatrix}\dot{\alpha}\\ \dfrac{1}{det(M)}\left(J_{z1} + m_1 l_1^2\right)\left(-C_0\dot{\alpha} - K_f\sigma(\dot{\alpha}) + u\right)\\ \dot{\beta}\\ \dfrac{1}{det(M)}\left(m_1 l_0 l_1\left(C_0\dot{\alpha} \ + K_f\sigma(\dot{\alpha}) - u\right)\right)\end{bmatrix} \qquad (3.10)$$

Since $\upsilon = \ddot{\beta} = 0$, from (3.10) follows that:

$$u = C_0\dot{\alpha} \ + K_f\sigma(\dot{\alpha}) \qquad (3.11)$$

Substituting (3.11) into (3.10), the zero-dynamics become of the form:

$$\frac{d}{dt}\begin{bmatrix}\alpha\\ \dot{\alpha}\end{bmatrix} = \begin{bmatrix}\dot{\alpha}\\ 0\end{bmatrix} \qquad (3.12)$$

Because subsystem (3.12) has a double eigenvalues at zero, the resulting dynamics are called weakly minimum phase. The PD controller (3.8) stabilizes the Furuta pendulum in upright position, but does not create asymptotic stabilization of $\dot{\alpha}$. Figure 3.3 shows the result of a simulation with this controller where $a_0 = a_1 = 10$.

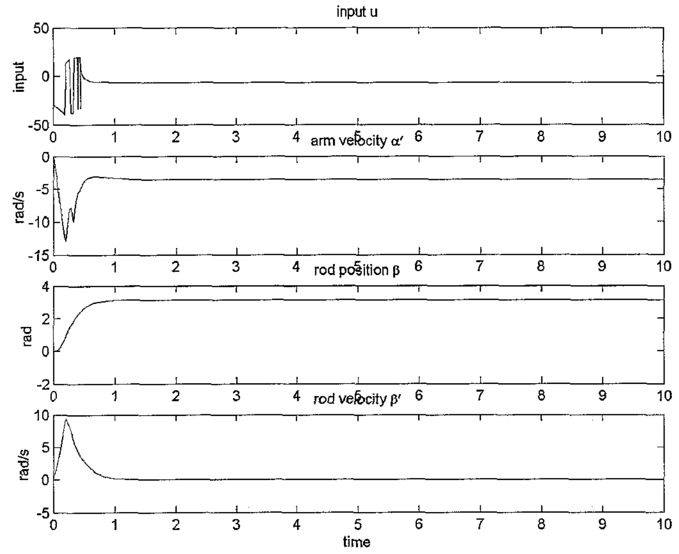

Figure 3.3: Simulation results with PD controller (3.8)

3.3.2 Achieving asymptotic stability

Now the pendulum rod has been stabilized, the new goal is to improve υ such that asymptotic stability can be created for the $(\dot{\alpha}, \beta, \dot{\beta})$ dynamics and $\dot{\alpha}$ converges to zero. To achieve this, an extra part will be added to the PD controller. The design of this new control input is based on the theory of forwarding, which is described in [Sepu96]. The dynamics of the Furuta pendulum after input-output linearization can be formulated in the form:

$$\begin{aligned} \ddot{\alpha} &= -\frac{\theta}{\gamma}\sin\varphi\dot{\alpha}^2 + \frac{\delta}{\gamma}\tan\varphi + \frac{\kappa}{\gamma\cos\varphi}\upsilon \\ \ddot{\varphi} &= \upsilon \end{aligned} \tag{3.13}$$

where:

$$\begin{array}{ll} \varphi = \beta - \pi & \\ \theta = m_1 l_1^2 & \gamma = m_1 l_0 l_1 \\ \kappa = J_{z1} + \theta & \delta = -m_1 g l_1 \end{array}$$

Herein φ is a new notation for the angle of the pendulum rod, such that $\varphi = 0$ corresponds to upwards position ($\beta = \pi$). Since our interest is to stabilize $\dot{\alpha}$, the position α will not be taken into account. Because of the quadratic term $\dot{\alpha}^2$, which will cause unstable behaviour outside a certain attraction region, no global asymptotic stability will be obtained. The attraction region however, is expected to be reasonable large. Introducing the notation $(x_1, x_2, x_3) = (\dot{\alpha}, \varphi, \dot{\varphi})$, equation (3.13) can be written in feedforward form:

$$\begin{aligned} \dot{x}_1 &= -\frac{\theta}{\gamma} sin\, x_2 x_1{}^2 + \frac{\delta}{\gamma} tan\, x_2 + \frac{\kappa}{\gamma\, cos\, x_2} \upsilon \\ \dot{x}_2 &= \dot{x}_3 \\ \dot{x}_3 &= \upsilon \end{aligned} \tag{3.14}$$

Although forwarding is a nonlinear technique, linearization is used to keep calculations simple. A controller derived from a linearized system will work for a nonlinear system, provided saturation on the control signal is used. Since for our system, the controller turned out to work without saturation, no attention has been paid to this subject. Linearization of (3.14) around the desired equilibrium point $(x_1, x_2, x_3) = (0{,}0{,}0)$ leads to:

$$\begin{aligned} \dot{x}_1 &= \mu x_2 + \nu \upsilon \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= \upsilon \end{aligned} \tag{3.15}$$

Where we have introduced the notation $\mu = \frac{\delta}{\gamma}$ and $\nu = \frac{\kappa}{\gamma}$. As was stated in section 3.3.1, the (x_2, x_3) - subsystem (3.15) is asymptotically stabilized with PD controller (3.8).

$$\begin{aligned} \dot{x}_2 &= x_3 \\ \dot{x}_3 &= \upsilon \end{aligned} \tag{3.16}$$

To prove this, a Lyapunov function V can be formulated

$$V = \frac{1}{2} x_2^2 + \frac{1}{2} (x_3 + a x_2)^2 \qquad > 0 \qquad \forall \qquad (x_2, x_3) \neq (0{,}0) \tag{3.17}$$

By construction, the time derivative of V is demanded to be negative away from the origin. This is true if it is has form

$$\dot{V} = -a x_2^2 - b (x_3 + a x_2)^2 \qquad < 0 \qquad \forall \qquad (x_2, x_3) \neq (0{,}0) \tag{3.18}$$

which can be achieved by PD control law (3.8) written as:

$$\upsilon = PD(x_2, x_3) = -\, a_0 x_2 - a_1 x_3 \;=\; -(1 + ab) x_2 - (a + b) x_3 \tag{3.19}$$

Where a and b are positive constants that depend on the choice of a_0 and a_1. Now the second order subsystem (3.16) has been stabilized, we can use the Lyapunov function V to construct a new function for the full third order system (3.15):

$$V_2 \;=\; V + \frac{1}{2} \lim_{t \to \infty} \tilde{x}_1{}^2(t) \tag{3.20}$$

where $\tilde{x}_1(t)$ is the first component of the solution of the linear system (3.15) with initial conditions $\bar{x}_0 = [x_1 \quad x_2 \quad x_3]^T$. A complete derivation of this solution has been worked out in Appendix C. Using the result of this derivation (B.6), V_2 can now be written in the form:

$$V_2 = V + \frac{1}{2}(x_1 + rx_2 + px_3)^2 > 0 \qquad \forall \qquad (x_1, x_2, x_3) \neq (0,0,0) \tag{3.21}$$

Where r and p are real numbers depending on the particular choice of a and b as well as on the system parameters featuring in (3.15). By construction the time derivative of V_2 satisfies:

$$\dot{V}_2 \Big|_{\upsilon = PD(x_2,x_3)} = \dot{V}\Big|_{\upsilon = PD(x_2,x_3)} \leq 0 \qquad \forall \qquad (x_1, x_2, x_3) \neq (0,0,0) \tag{3.22}$$

This derivative will be rendered negative by adding an extra term to the control signal, such that a feedback becomes $\upsilon = \upsilon(x_1, x_2, x_3) = PD(x_2, x_3) + \upsilon_2$. Using this expression for υ, the feed forward system (3.15) can be written in matrix representation:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \underline{A}\bar{x} + \bar{b}\,\upsilon_2 = \begin{bmatrix} 0 & (\mu - \nu - \nu ab) & -\nu(a+b) \\ 0 & 0 & 1 \\ 0 & -(1+ab) & -(a+b) \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \nu \\ 0 \\ 1 \end{bmatrix}\upsilon_2 \tag{3.23}$$

The time derivative of V_2 with the new feedback $\upsilon(x_1,x_2,x_3)$ can derived by

$$\dot{V}_2 \Big|_{\upsilon = PD(x_2,x_3) + \upsilon_2} = \bar{\nabla} V_2 \underline{A}\bar{x} + \bar{\nabla} V_2\, \bar{b}\,\upsilon_2 = \dot{V}_2\Big|_{\upsilon = PD(x_2,x_3)} + \bar{\nabla} V_2\, \bar{b}\,\upsilon_2 \tag{3.24}$$

and can be rendered negative by choosing $\upsilon_2 = -k \cdot \bar{\nabla} V_2\, \bar{b}$, where k is a positive constant. (3.24) becomes:

$$\dot{V}_2 \Big|_{\upsilon = PD(x_2,x_3) + \upsilon_2} = -ax_2^2 - b(x_3 + ax_2)^2 - k(\bar{\nabla} V_2\, \bar{b})^2 < 0 \quad \forall \quad (x_1, x_2, x_3) \neq (0,0,0) \tag{3.25}$$

The positive definite Lyapunov function V_2 now has a negative definite time derivative. Asymptotic stability of the system (3.15) is thus achieved by the feedback:

$$\upsilon(x_1, x_2, x_3) = -(1+ab)x_2 - (a+b)x_3 - k \cdot \bar{\nabla} V_2\, \bar{b} \tag{3.26}$$

where

$$\begin{aligned} \bar{\nabla} V_2\, \bar{b} &= \begin{bmatrix} \frac{\partial V_2}{\partial x_1} & \frac{\partial V_2}{\partial x_2} & \frac{\partial V_2}{\partial x_3} \end{bmatrix}\begin{bmatrix} \nu \\ 0 \\ 1 \end{bmatrix} = \nu\frac{dV_2}{dx_1} + \frac{dV_2}{dx_3} \\ &= (\nu + p)x_1 + ((\nu + p)r + a)x_2 + (p^2 + \nu p + 1)x_3 \\ &= c_1 x_1 + c_2 x_2 + c_3 x_3 \end{aligned} \tag{3.27}$$

where we have introduced the notation c_1, c_2,, c_3. Substituting (3.27) into (3.25), $\dot{V}_2$ may also be written as:

$$\dot{V}_2\Big|_{\upsilon=PD(x_2,x_3)+\upsilon_2} = -ax_2^2 - b(x_3+ax_2)^2 - k(c_1x_1+c_2x_2+c_3x_3)^2 < 0 \quad \forall \quad (x_1,x_2,x_3) \neq (0,0,0) \tag{3.28}$$

and the corresponding control law (3.26):

$$\upsilon(x_1,x_2,x_3) = -(1+ab)x_2 - (a+b)x_3 - k(c_1x_1+c_2x_2+c_3x_3) \tag{3.29}$$

The values of c_1, c_2 and c_3 depend on the choice of parameters a and b. Any value may be chosen in order to tune the performances of the controller. In all simulations and experimentations however, a and b are 1 and 9 respectively, which leads to the numerical values:

$$c_1 = -4.0041$$
$$c_2 = 161.3267$$
$$c_3 = 23.3412$$

So the implemented final control law is:

$$\upsilon(\dot{\alpha},\beta,\dot{\beta}) = -10\beta - 10\dot{\beta} - k(-4\dot{\alpha} + 161\beta + 23\dot{\beta}) \tag{3.30}$$

Simulation results with this controller are shown in figure 3.4. In this simulation, the gain $k = 3$. A proof of the negative definiteness of $\dot{V}_2$ is given in Appendix D.

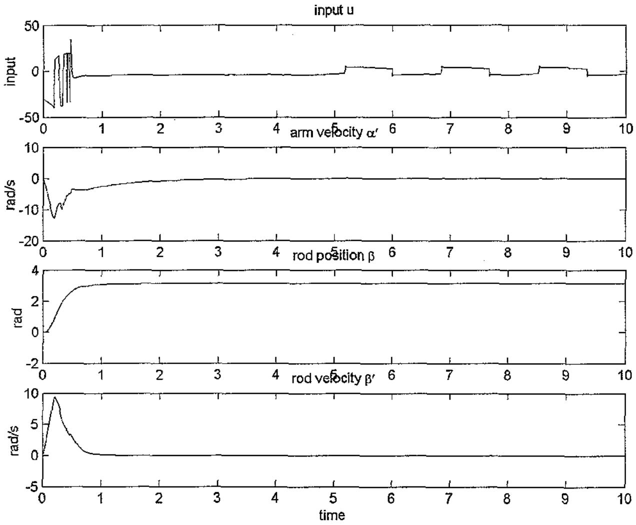

Figure 3.4: Simulation results with the new controller (3.30)

Chapter 4

Experiments

Now the controller turns out to work quite well in simulations, it can be implemented on the real robot. The system is equipped with a TUeDACS data-acquisition module, which is operated via the program Wintarget. After some adaptations, the MATLAB SIMULINK controller that was used for simulations, can directly be used for real-time control.

In every experiment the swing-up controller is used in order to bring the pendulum. Switching is done when $cos\,\beta = -0.8$, which corresponds to $\beta = 143°$. First experiments were performed with PD-controller (3.8) from section 3.3.1 with $a_0 = a_1 = 10$:

$$\upsilon = PD(\beta, \dot{\beta}) = -10(\beta - \pi) - 10\dot{\beta} \tag{4.1}$$

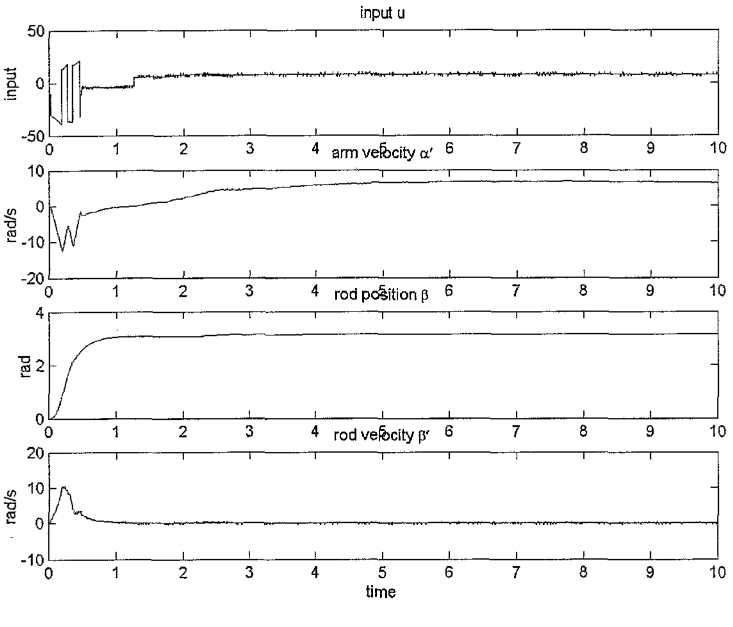

Figure 4.1: Experimental results with the PD controller

As predicted the pendulum is balanced in upright position and the arm ends up rotating with a constant angular velocity.

Next the extended control law (3.30) is tested. Like in simulations choosing gain k = 3 resulted in quite good results, which are shown in figure 4.2. As can be seen, $\dot{\alpha}$ now converges to zero. During most experiments very small movements of the arm remained visible after stabilization. Possible causes for this are the fact that the motor dynamics have been neglected from the model, system noise and the inaccuracy of the simplified coulomb friction model that was used. Adding a position controlling feedback may lead to a better performance.

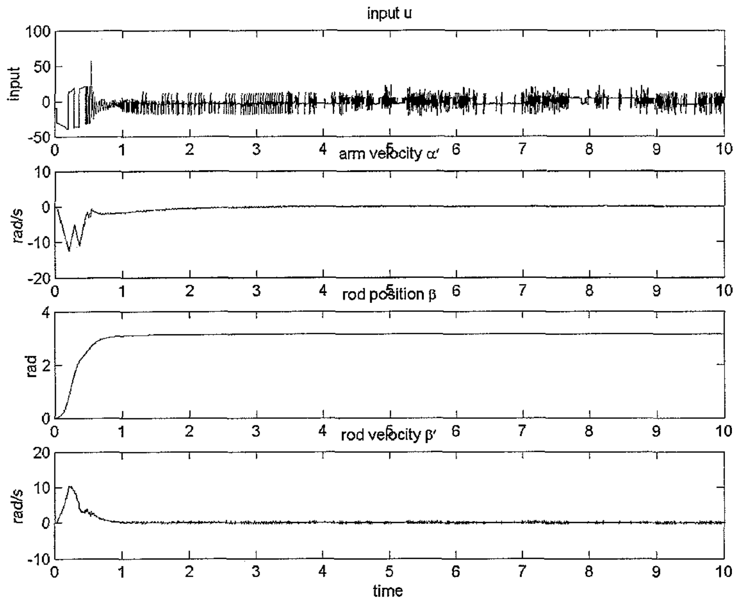

Figure 4.2: Also the arm velocity converges to zero

The switching point $cos\,\beta = -0.8$ that was chosen in foregoing experiments, gave the "smoothest" results with the used swing-up controller. Better performances of the controller may be made possible by fine-tuning the gain values. The theory of forwarding that was used to asymptotically stabilize the arm velocity, does not guarantee global convergence since the quadratic term $\dot{\alpha}^2$ is not allowed by the theory. More information about this can be found in [Sepu96].

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

During the three months of the project, a nonlinear controller has been designed and implemented on a Furuta pendulum. The controller is capable of stabilizing the pendulum rod in the upright position and steering the velocity of the actuated arm to zero. Although motor dynamics have been neglected and a very simplified friction compensation model was used, the performance turned out to be quite good in experiments. The control strategy consists of two parts; a swing-up controller and a stabilizing controller. They take over from each other at an adjustable switching point. The implemented energy based swing-up controller was designed for a pendulum on a cart, so it does not take into account the rotation of the actuated arm. Nevertheless it swings up the Furuta pendulum very well. Making use of a feedback input-output linearizing control law, it was possible to compensate nonlinearities of the pendulum rod dynamics. After a swing-up is performed, a PD controller can stabilize the pendulum rod in vertical position. The resulting zero-dynamics for the actuated arm are weakly minimum phase. Making use of the theory of forwarding, a control feedback is designed that achieves local asymptotic stability steering the actuated arm rotational velocity to zero. The control law is defined in every position of the pendulum, except for when it is horizontally.

This project shows that it is possible to design a properly working nonlinear controller for an under actuated system like the Furuta pendulum.

5.2 Recommendations

After stabilization, very small movements of the rotating arm remain visible. A probable cause of this is the rather inaccurate symmetric Coulomb friction model that has been used for friction compensation. Especially in the region around zero velocity, a more accurate knowledge of friction effects could inprove performance. Also, including the motor dynamics in the system equations may give better results.

The swing-up controller that was implemented neglects the effect of the rotational motion of the pendulum joint. Although it turned out to work quite well, a quicker swing-up may be possible when the dynamics of these rotations are taken into account.

A quadratic term is present in the mathematical equations of the arm velocity dynamics. Since the theory of forwarding does not support this. Therefore the forwarding procedure has achieved local asymptotic stability, but does not guarantee global convergence. Outside a certain attraction region, instable behaviour may occur. Further research on this subject may increase the attraction region.

The forwarding procedure has achieved asymptotic stability of the arm velocity. However, the position of the actuated arm is still random after stabilization. Another application of the forwarding procedure may be used to stabilize the actuated arm at a desired position.

Bibliography

[Arno03] M.B. Arnolds, Technical Report Traineeship University of Eindhoven, 2003, DCT report 2003.100

[Åstr00] K.J. Åström and K.Furuta, "Swinging up a pendulum by energy control", *Automatica,* volume 36 pp 287-295, 2000

[Hame04] M. Hamers, personal contact during his work on the Furuta pendulum

[Khal96] Hassan K. Khalil, "Nonlinear systems", Third edition, 1996, New Jersey, Prentice Hall Inc.

[Nijm02] Nijmeijer, H., slides for course no. 4J100, "Control of nonlinear mechanical systems, 2002, Eindhoven.

[Sepu96] Rodolphe Sepulchre, Mrdjan Janković and Petar Kokotović, "Constructive Nonlinear Control", Santa Barbara, California, August 1996

Appendix A

Model parameter values

Table A.1 shows the numerical values of the used model parameters. Parameter values are taken from [Hame04] and have been estimated recently before the start of this project.

Table A.1: model parameters

Symbol	Value	Description
m_1 $[kg]$	0.208	Mass of the pendulum rod
l_0 $[m]$	0.245	Length of the actuated arm
l_1 $[m]$	0.272	Length to the centre of inertia of the pendulum rod
J_{z0} $[kgm^2]$	0.356	Moment of inertia of the actuated arm
J_{z1} $[kgm^2]$	0.0064	Moment of inertia of the pendulum rod
C_0 $[Ns/m]$	0.685	Viscous friction coefficient of the motor axis
C_1 $[Ns/m]$	0.010	Viscous friction coefficient of the pendulum joint
K_f $[-]$	4.0	Coulomb friction coefficient
k $[-]$	50	Steepness of smoothened Coulomb friction model
g $[m/s^2]$	9.81	Gravity acceleration

Appendix B

Numerical computation of I/O linearizing feedback laws

The input-output linearizing feedback laws for the swing-up controller in section 3.2 and for the stabilizing controller in section 3.3.1 have been computed numerically in undergoing MATLAB script.

```
%%% This file computes of input - output linearizing feedback laws U1 for the
%%% swing-up and U2 for the stabilizing controller.

clear all

syms x u Jz0 Jz1 m1 lcm1 l0 l1 C0 C1 Kf g k alpha alphadot beta betadot sig-
maclmb;

%% system model

M  = [Jz0 + m1*l0^2 + m1*l1^2*sin(beta)^2     m1*l0*l1*cos(beta)
                m1*l0*l1*cos(beta)                  Jz1 + m1*l1^2];

D1 = [C0 + m1*l1^2*betadot*sin(beta)*cos(beta)
           -m1*l1*alphadot*sin(beta)*cos(beta)];

D2 = [m1*l1^2*alphadot*sin(beta)*cos(beta) - m1*l0*l1*betadot*sin(beta)
                                                                      C1];

D  = [D1 D2];


K  = [Kf * sigmaclmb
      m1*g*l1*sin(beta)];           %sigmaclmb = 1 - 2/(exp(2*k*alphadot)+1);

U  = [u ; 0];

abdotdot = inv(M) * (-D * [alphadot ; betadot] - K + U);

alphadotdot = abdotdot(1);
betadotdot   = abdotdot(2);
```

```
%% Solving alphadotdot = mu gives the wished feedback law for alpha

syms mu;

b  = alphadotdot - mu;

Ub = solve(b,u)
Us = simple(Ub)           % input-output linearizing feedback

%% Solving betadotdot = nu gives the wished feedback law for beta

syms nu;

p  = betadotdot - nu;

Up = solve(p,u)
U = simple(Up)            % input-output linearizing feedback
```

Appendix C

Solving system of ODE's

For construction of Lyapunov function V_2 (3.20) in section 3.3.2, $\tilde{x}_1(t)$ has to be determined, which is the first component of the solution of the linear system of ODE's (3.15) with initial conditions

$$\bar{x}_0 = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

With PD controller (3.8),

$$\upsilon = -(1+ab)x_2 - (a+b)x_3 \tag{B.1}$$

the feedforward system (3.15) can be written as:

$$\begin{aligned} \dot{x}_1 &= \mu x_2 - \nu(1+ab)x_2 - \nu(a+b)x_3 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= -(1+ab)x_2 - (a+b)x_3 \end{aligned} \tag{B.2}$$

Which can be written in matrix form:

$$\dot{\bar{x}} = \underline{A}\,\bar{x}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & \mu - \nu(1+ab) & -\nu(a+b) \\ 0 & 0 & 1 \\ 0 & -(1+ab) & -(a+b) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{B.3}$$

The analytical solution of this system of linear differential equations, with initial conditions

$$\bar{x}(0) = \bar{x}_0$$

can be computed with:

$$\bar{x}(t) = e^{At}\bar{x}_0 = Qe^{\Lambda t}Q^{-1}\,\bar{x}_0 = Q\begin{bmatrix} e^{\lambda_1 t} & 0 & 0 \\ 0 & e^{\lambda_2 t} & \\ 0 & 0 & e^{\lambda_3 t} \end{bmatrix}Q^{-1}\,\bar{x}_0 \tag{B.4}$$

Where Λ is a diagonal matrix of the eigenvalues of A and the columns of Q are the corresponding eigenvectors. In (3.20) the first component of the solution for t approaching infinity is needed. Since $\lambda_1 = 1$ and $Re[\lambda_2, \lambda_3] < 0$, using (B.4) this becomes:

$$\lim_{t\to\infty} \tilde{x}_1(t) = \bar{x}(\infty) = Q\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}Q^{-1}\,\bar{x}_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}Q^{-1}(1,:)\,\bar{x}_0 \tag{B.5}$$

.

Where $Q^{-1}(1,:)$ is the first row of Q^{-1}. The numerical solution of (B.5) has been computed with MATLAB. With $a = 1$ and $b = 9$ and the system parameters from appendix A, this results in:

$$\lim_{t\to\infty} \tilde{x}_1(t) = [1 \quad r \quad p]\bar{x}_0 = x_1 + rx_2 + px_3 \tag{B.6}$$

$$r = r(a,b) = -40.04$$
$$p = p(a.b) = -5.58$$

Solution (B.6) is used in section 3.3.2.

Appendix D

Verification of asymptotic stability

In this appendix proofs that control law (3.30) achieves asymptotic stability for system (3.30). The system (3.15) is proved to be asymptotically stabilized by the feedback law:

$$\upsilon(x_1,x_2,x_3)=PD(x_2,x_3)+\upsilon_2=-(1+ab)x_2-(a+b)x_3-k\cdot\overline{\nabla V_2}\,\overline{b} \qquad \text{(D.1)}$$

if can be verified that Lyapunov function

$$V_2=\frac{1}{2}x_2^2+\frac{1}{2}(x_3+ax_2)^2+\frac{1}{2}(x_1+rx_2+px_3)^2>0 \quad \forall \quad (x_1,x_2,x_3)\neq(0,0,0) \qquad \text{(D.2)}$$

has a negative definite time. When control law (D.1) is applied, the time derivative becomes

$$\dot{V}_2\Big|_{\upsilon=PD(x_2,x_3)+\upsilon_2}=-ax_2^2-b(x_3+ax_2)^2-k\left(\overline{\nabla V_2}\,\overline{b}\right)^2 \qquad \text{(D.3)}$$

where

$$\begin{aligned}\overline{\nabla V_2}\,\overline{b}&=\begin{bmatrix}\frac{dV_2}{dx_1} & \frac{dV_2}{dx_2} & \frac{dV_2}{dx_3}\end{bmatrix}\begin{bmatrix}v\\0\\1\end{bmatrix}=v\frac{dV_2}{dx_1}+\frac{dV_2}{dx_3}\\&=(v+p)x_1+((v+p)r+a)x_2+(p^2+vp+1)x_3\\&=c_1x_1+c_2x_2+c_3x_3\end{aligned} \qquad \text{(D.4)}$$

The values of c_1, c_2 and c_3 depend on the choice of parameters a and b. Substitution of (D.4) into (D.1) and choosing $k=1$, the control law becomes:

$$\upsilon(x_1,x_2,x_3)=-(1+ab)x_2-(a+b)x_3-(c_1x_1+c_2x_2+c_3x_3) \qquad \text{(D.5)}$$

And the time derivative of our Lyapunov function:

$$\begin{aligned}\dot{V}_2\Big|_{\upsilon = PD(x_2,x_3)+\upsilon_2} &= -ax_2^2 - b(x_3 + ax_2)^2 - (c_1x_1 + c_2x_2 + c_3x_3)^2 \\ &= -c_1{}^2x_1^2 - \left(a + a^2b + c_2{}^2\right)x_2^2 - (b + c_3)x_3^2 \\ &\quad - 2c_1c_2x_1x_2 - 2c_1c_3x_1x_3 - (2ab + 2c_2c_3)x_2x_3\end{aligned} \tag{D.6}$$

Because (D.6) is a quadratic function of the state variables, it can be written in matrix form as

$$\dot{V}_2\Big|_{\upsilon=PD(x_2,x_3)+\upsilon_2} = \bar{x}\,\underline{P}\,\bar{x}^T = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{12} & p_{22} & p_{23} \\ p_{13} & p_{23} & p_{33} \end{bmatrix} \tag{D.7}$$

$$\begin{aligned} p_{11} &= -c_1^2 & \qquad p_{12} &= -c_1c_2 \\ p_{22} &= -\left(a + a^2b + c_2^2\right) & p_{13} &= -c_1c_3 \\ p_{33} &= -\left(b + c_3^2\right) & p_{23} &= -(ab + c_2c_3) \end{aligned}$$

If $\dot{V}_2$ is a negative definite time derivative of Lyapunov function V_2 away from the origin, then this proves that system (3.15) is asymptotically stabilized by the feedback control law (D.1). This is true if and only if $\underline{P}$ is a negative definite matrix. With $a = 1$ and $b = 9$, the following numerical values have been computed.

$$\begin{aligned} c_1 &= -4.0041 \\ c_2 &= 161.3267 \\ c_3 &= 23.3412 \end{aligned}$$

Substitution in (D.7) yields matrix $\underline{P}$:

$$\underline{P} = 10^4 \cdot \begin{bmatrix} -0.0016 & 0.0646 & 0.0093 \\ 0.0646 & -2.6036 & -0.3775 \\ 0.0093 & -0.3775 & -0.0554 \end{bmatrix} < 0$$

$\underline{P}$ is a negative definite matrix, which proves asymptotic stability for third order system (3.15).