The mechatronics kit

IPICYT

Title report:

The Mechatronics Kit
First survey

DCT-number: 2004-27
Author: Jos A. C. Meesters
Advisors: Dr. David Lizárraga
Prof. Dr. Henk Nijmeijer

30 November 2003
San Luis Potosi, México

Contents

Chapter 1 Introduction

Underactuated systems have become more important these days. The first main reason for research on underactuated systems is that they have less inertia, thus most probably higher positioning accuracy can be maintained. The second is that all flexible robots are essentially underactuated, when one accounts for the flexibility modes of the joints. Besides this, underactuated systems are cheaper because less actuators are needed than in the fully actuated case.

The Applied Mathematics Department of the University of San Luis Potosí (Mexico), IPICyT, acquired a Mechatronics Kit. With this Mechatronics Kit it is possible to construct four different configurations, the *inertia wheel*, the *pendubot*, the *reaction wheel* and the *Furuta pendulum*. The *reaction wheel*, the *pendubot* and *the Furuta pendulum* are underactuated systems because they have 2 degrees of freedom and only one actuator. The *inertia wheel* is the commonly used experimental setup for experiments of students starting in control.

This report presents the results of a traineeship that lasted for three months. Upon its arrival, the Mechatronics Kit was mechanically built up, but further almost nothing was known, so the first thing to do was to get familiar with the hardware and software delivered by the manufacturer *Quanser*. A short review about this topic is given first, then the equations of motions are derived in chapter 2, followed by an attempt to identify the parameters describing these equations of motion in chapter 3. As will be shown, this task is not trivial. Furthermore some controllers are implemented in chapter 4. Conclusions and recommendations can be found in chapter 5. The main goal of the traineeship is to gain as much knowledge as possible about how one should operate the system, how to identify the systems and to investigate what controllers are already implemented. The knowledge has to be reported in a way such that future students can start with their experiments very easily.

Chapter 2 Systems description

With the Mechatronics Kit, four configurations can be built that will be described in this section one by one. In order to be able to send data to the actuator and retrieve data from the optical encoders, a *Texas Instruments* microprocessor board is included which is able of acquiring, processing and generating data. The card is connected through the parallel port of a host PC. On this host PC one can write source code which optionally can be flashed onto the board for demonstration purposes. The working environment is *Code Composer Studio*, which is a program where one can very easily analyse and change C-languague and assembler files that are needed for communicating with the Mechatronics Kit. The source code is written in such a way that it waits for *Visual Basic* commands to start or stop action. The *Visual Basic* application is a very simple and straightforward interface with just the most necessary buttons for controlling the specific configurations. More detailed information about the ways of communicating, necessary files and typical source code commands can be found in appendix 6.

The Mechatronics Kit is supplied with only one *Pitmann* motor. The input of this motor is the current I. When the motor shaft is rotating a voltage is created known to produce a back Electro Magnetic Force, or in this case torque. On the Mechatronics Kit an external controller is mounted that makes sure that the input current I is achieved despite the back EMF. It is assumed that both the dynamics of the motor and its controller are negligible compared to the dynamics of the mechanical parts. Consequently, the following proportional relation between the torque and the current holds

$$\tau = kI$$

Here k is the motor's torque constant in $\left[\frac{Nm}{A}\right]$

Furthermore there is a proportional relation between the current and the control command in the computer, u, namely

$$kI = k_u u$$

It is decided to interpret u as a scaled current. This scaled current is saturated at the numerical value of 10 (or –10). This is the situation in which the external controller cannot compensate anymore for the back EMF due to too high rotational velocity of the motor shaft. In the sequel of this report the input is denoted as τ, but obviously should be interpreted as $k_u u$. The value of k_u is documented as 0.00494 (no unit if u is interpreted as a current).

2.1 Inertia Wheel

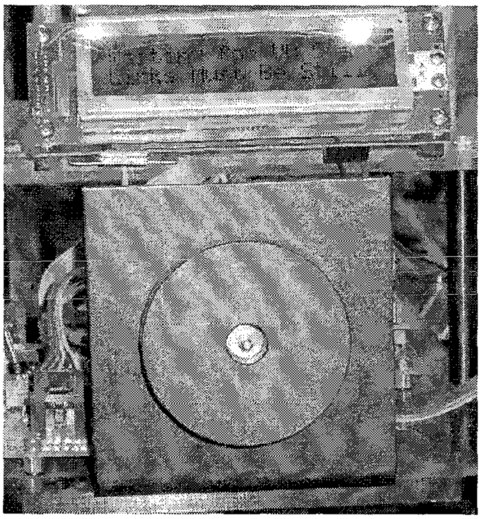


Figure 2.1: Picture of the experimental set-up of the inertia wheel

A picture of the *inertia wheel* configuration can be seen in figure 2.1. Indeed this is one of the simplest configurations possible in control engineering. The system is fully actuated and there are many textbooks that describe the control of this configuration. It is interesting, however, to identify (that is, to numerically estimate) the parameters of this system because they will be needed for the *reaction wheel* configuration as well.

Because the system has only one degree of freedom there is only one 2nd order equation of motion, namely

$$J\ddot{\theta} = \tau - f \tag{2.1}$$

with

J : Inertia of the inertia wheel ($kg \cdot m^2$)
θ : Angle of the inertia wheel with respect to the downward vertical (rad)
τ : Input (Nm)
f : Friction term (Nm)

The friction term only contains a Coulomb offset and a viscous term. Although this friction model is not highly accurate, our attempts to employ the more accurate "LuGre Model", which includes the Stribeck curve, see [6] and [7], were hindered by the limited resolution (4000 pulses per revolution) of the optical encoders provided with the Mechatronics Kit. Indeed, that resolution is not high enough to allow for an accurate estimation of the parameters involved in the LuGre friction model. Therefore a simple friction model is used. The model is further simplified (in the sense that discontinuous functions are approximated by differentiable functions) since differential equations with a discontinuity in the right hand side are very hard to integrate numerically. The price to pay for this simplification is the loss of the stick-phase, which is very characteristic in friction behaviour. Then the final friction model becomes

$$f = b_1 \cdot \dot{\theta} + b_2 \cdot F_c \qquad (2.2)$$

with

$$F_c = 1 - \frac{2}{e^{2 \cdot k \dot{\theta}} + 1} \qquad (2.3)$$

F_c is known as the Sigmoid function. It approximates the Coulomb offset, but has no discontinuity at zero velocity. When k>15, the function satisfactorily approximates the real Coulomb offset, meaning that increasing k further gives no significant difference.

To estimate the numerical values of the friction parameters a very simple experiment was carried out. If the disk is brought to a reference speed then in theory no more input torque is needed to keep the disk at this speed. In practice this is not true because the system is subject to friction forces that needs to be compensated for. By measuring the input torque at constant velocity, a velocity-dependent friction model can be estimated, fulfilling (2.2). A graphical representation of this model can be seen in figure 2.2. For each velocity there is now a measured (averaged) torque. Taking the least squares approximation of these measurements, the following model is estimated

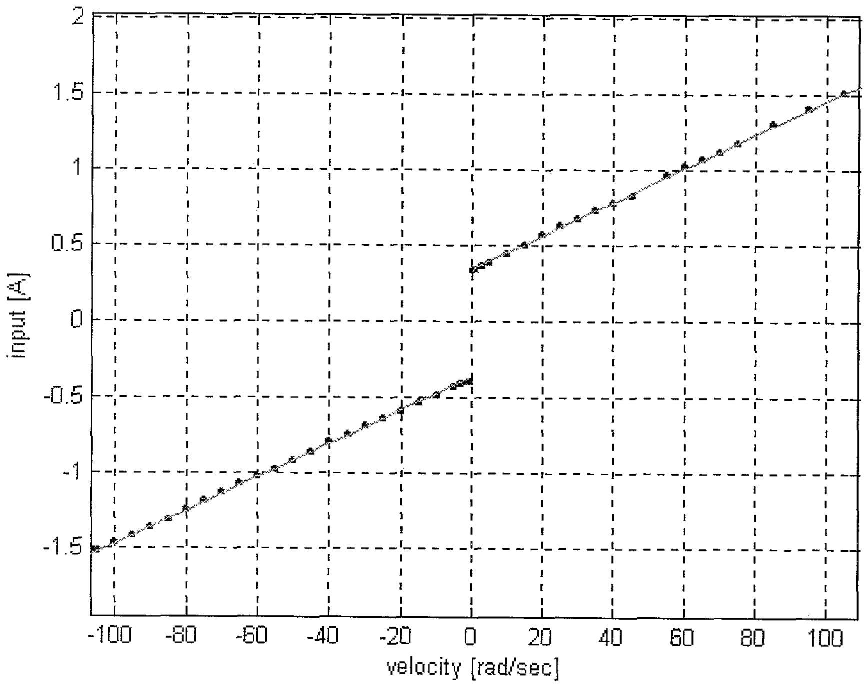


Figure 2.2: Least squares fitting of the friction model

$$f\left(\dot{\theta}\right) = \begin{cases} 0.3399 + 0.0112\dot{\theta}, & \dot{\theta} \geq 0 \\ -0.3575 + 0.111\dot{\theta}, & \dot{\theta} < 0 \end{cases} \qquad (2.4)$$

The parameter J consists of both the inertia of the disk and the inertia of the shaft of the motor. J is estimated by means of eight measurements. During these measurements different reference accelerations were applied and a simple velocity controller was capable of controlling the different ramps of the velocities that caused these constant reference accelerations. Special attention has to be given to the fact that the motion speeds should not be too low because the friction model is not accurate enough at low speeds. If one compensates for the friction by means of (2.4), one can measure the torque that is needed for letting the disk constantly accelerate. In fact this comes down to dividing the input by the acceleration for each reference acceleration. Taking the averaged value of these eight measurements J is found to be equal to

$$J = 0.0045 \ kgm^2$$

2.2 *Pendubot*

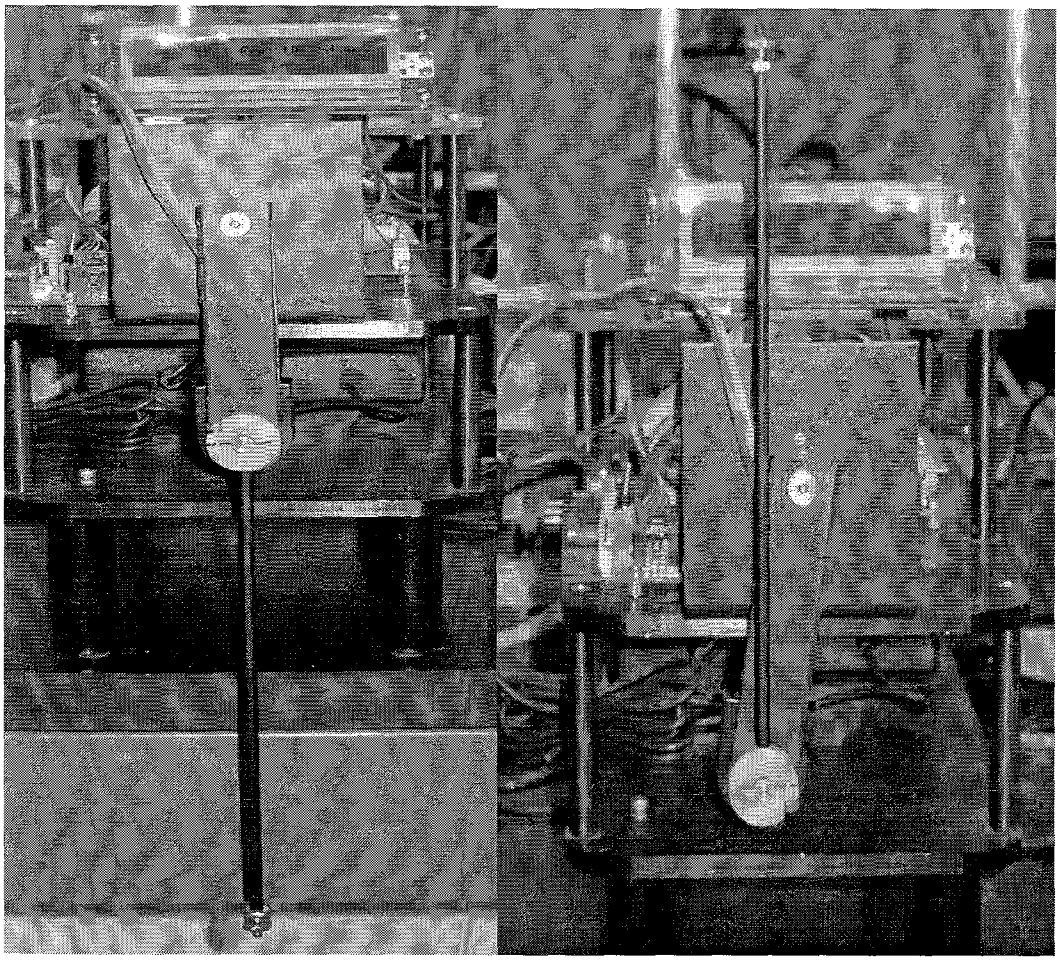


Figure 2.3: a) Pendubot in initial condition *b) Pendubot in semi-inverted position*

A picture of the *pendubot* configuration can be seen in figure 2.3. The only actuator of this system can be found at the first joint, i.e. the shoulder of the first link. The second link can rotate freely with respect to the first. After some first experiments it is assumed to neglect the friction of the movement of the second link relatively to the first link. The friction of the first links motion, due to the motor, cannot be neglected and is therefore modelled by means of a sigmoid function that approximates the coulomb offset and additionally a viscous term.

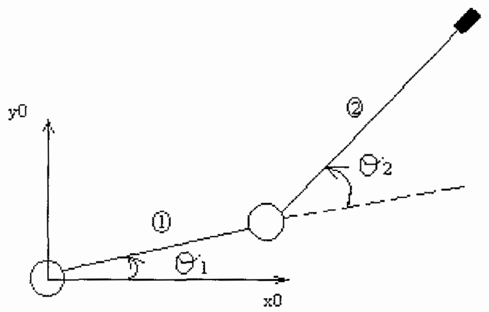


Figure 2.4: Systematic drawing of the pendubot

Considering figure 2.4 and taking

m_1	:	mass of the first link (kg)
l_{c1}	:	centre of mass of the first link (m)
L_1	:	length of the first link (m)
J_1	:	inertia of the first link with respect to its centre of mass (kgm^2)
m_2	:	mass of the second link (kg)
l_{c2}	:	centre of mass of the second link (m)
J_2	:	inertia of the second link with respect to its centre of mass (kgm^2)
$q_1 = \theta_1$	:	first generalised coordinate, representing the angle of the first link with respect to the horizontal (rad)
$q_2 = \theta_2$	:	second generalised coordinate representing the angle of the second link with respect to the first link (rad)
τ	:	input (Nm)
f	:	friction term (Nm)

the derivation of the equations of motion can be found in appendix 1, and the main result is the following

$$\underline{D} \cdot \ddot{q} + \underline{C} \cdot \dot{q} + \underline{G} = \underline{\tau} - \underline{f} \qquad (2.5)$$

with

$$\underline{D} = \begin{bmatrix} p_1 + p_2 + 2 \cdot p_3 \cdot \cos(q_2) & p_2 + p_3 \cdot \cos(q_2) \\ p_2 + p_3 \cdot \cos(q_2) & p_2 \end{bmatrix}$$

$$\underline{C} = \begin{bmatrix} -p_3 \cdot \sin(q_2) \cdot \dot{q}_2 & -p_3 \cdot \sin(q_2) \cdot (\dot{q}_1 + \dot{q}_2) \\ p_3 \cdot \sin(q_2) \cdot \dot{q}_1 & 0 \end{bmatrix}$$

$$\underline{G} = \begin{bmatrix} p_4 \cdot g \cdot \cos(q_1) + p_5 \cdot g \cdot \cos(q_1 + q_2) \\ p_5 \cdot g \cdot \cos(q_1 + q_2) \end{bmatrix}$$

$$\underline{\tau} = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix}$$

$$\underline{f} = \begin{bmatrix} p_6 \cdot \dot{q}_1 + p_7 \cdot \left(1 - \dfrac{2}{\exp(2 \cdot k \cdot \dot{q}_1) + 1}\right) \\ 0 \end{bmatrix}$$

Here, p_1 through p_7 represent the base parameter set. These are the minimal number of parameters needed to completely describe the system's properties. Their values are non-linear combinations of the physical parameters of the system:

$$\begin{aligned} p_1 &= m_1 \cdot l_{c1}{}^2 + m_2 \cdot l_1^2 + I_1 \\ p_2 &= m_2 \cdot l_{c2}^2 + I_2 \\ p_3 &= m_2 \cdot l_1 \cdot l_{c2} \\ p_4 &= m_1 \cdot l_{c1} + m_2 \cdot l_1 \\ p_5 &= m_2 \cdot l_{c2} \end{aligned} \tag{2.6}$$

A digital encoder that is connected to the motor shaft measures the first angle. The second links angle is measured by means of a digital encoder that is connected to the first link. The movement of the first link is therefore limited. Another point of concern is that by putting the cables for the encoders on the *pendubot* there exists more, and more important, uncertain friction. This friction is very hard (if even possible) to identify.

Another less elegant aspect of the *pendubot* configuration is the mechanical play between the first and second link. This play will eventually result in the fact that the most wanted and thus most exciting trajectories are not achievable because the play is affecting the dynamics of these excitations significantly.

The last less wanted fact about the *pendubot* setup is the fact that the ratio between the friction of the motor and the inertia of the links is a little awkward, meaning that the friction is a little on the high side.

There are two possible goals with the *pendubot.* These are stabilization in the semi-inverted position, see figure 2.3b, and stabilization in the fully inverted position, see figure A3 in appendix 3. When defining the state-vector of the *pendubot* as $\underline{x} = \begin{bmatrix} q_1 & \dot{q}_1 & q_2 & \dot{q}_2 \end{bmatrix}^T$, the initial condition will be $\underline{x} = \begin{bmatrix} -\frac{\pi}{2}, 0, 0, 0 \end{bmatrix}^T$. Consequently the semi-inverted position will be $\underline{x} = \begin{bmatrix} -\frac{\pi}{2}, \pi, 0, 0 \end{bmatrix}^T$, and the fully inverted position will be $\underline{x} = \begin{bmatrix} \frac{\pi}{2}, 0, 0, 0 \end{bmatrix}^T$.

Because the dynamics of the *pendubot* are non-linear and the deviations from the desired equilibrium points are not small, a linear state feedback controller will not be sufficient. The system will be swung up to its equilibrium point with the use of a non-linear controller. When the system comes in a neighbourhood (which is to be specified) of the equilibrium point a linear controller will take over the controlling action. First it must be shown that the system's linear approximation at the equilibrium points is fully controllable, so the linear matrices which describe the system at its equilibrium points need to be specified. Thereafter a non-linear controller will be derived.

Expanding the matrix differential equations of (2.5), and assuming that the friction affects can be compensated for, this results in

$$d_{11}\ddot{q}_1 + d_{12}\ddot{q}_2 + c_{11}\dot{q}_1 + c_{12}\dot{q}_2 + G_1 = \tau_1 \qquad (2.7)$$

$$d_{21}\ddot{q}_1 + d_{22}\ddot{q}_2 + c_{21}\dot{q}_1 + G_2 = 0 \qquad (2.8)$$

Solving for $\ddot{q}_2$ in (2.8), and substituting it into (2.7), one obtains

$$\ddot{q}_1 = \left(\frac{d_{12}c_{21}}{d_{22}\bar{d}} - \frac{c_{11}}{\bar{d}}\right)\dot{q}_1 - \frac{c_{12}}{\bar{d}}\dot{q}_2 + \frac{d_{12}G_2}{d_{22}\bar{d}} + \frac{\tau_1 - G_1}{\bar{d}} \qquad (2.9)$$

$$\ddot{q}_2 = \left(\frac{d_{21}c_{11}}{d_{22}\bar{d}} - \frac{c_{21}}{d_{22}} - \frac{d_{12}d_{21}c_{21}}{d_{22}^2\bar{d}}\right)\dot{q}_1 + \frac{d_{21}c_{12}}{d_{22}\bar{d}}\dot{q}_2 + \frac{d_{21}}{d_{22}\bar{d}}G_1 - \left(\frac{1}{d_{22}} + \frac{d_{12}d_{21}}{d_{22}^2\bar{d}}\right)G_2 \qquad (2.10)$$
$$- \frac{d_{21}}{d_{22}\bar{d}}\tau_1$$

with $\bar{d} = d_{11} - \frac{d_{12}d_{21}}{d_{22}}$

With these equations it is now possible to linearize the non-linear equations of motion around their equilibrium points. In order to minimize the likelihood of computational errors the help of *Maple* is used for linearizing the system. With the help of *Maple* matrices need to be found such that

$$\underline{\dot{x}} = \underline{A}x + \underline{B}u \qquad (2.11)$$

equals the linear approximation of the system around the equilibrium points. Here $\underline{x}$ represents small deviations from the equilibrium point. Then consequently:

$$\underline{A} = \frac{\partial f}{\partial x}\left(\bar{x},\bar{u}\right)$$

$$\underline{B} = \frac{\partial f}{\partial u}\left(\bar{x},\bar{u}\right)$$

with $f\left(\bar{x},\bar{u}\right)$ representing (2.9) and (2.10) in the case that the vector $\bar{x}$ (the state vector at the equilibrium points), and $\bar{u}$ (the input at the equilibrium points) are substituted into (2.9) and (2.10).

Because the expressions for the elements of A and B in (2.11) become extremely large, the values of the angles and angular speeds at the equilibrium points are already filled into the matrices. For the full expressions describing these matrices, see appendix 2.

For the semi-inverted position one obtains

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{21} & 0 & a_{23} & 0 \\ 0 & 0 & 0 & 1 \\ a_{41} & 0 & a_{43} & 0 \end{bmatrix} \quad \text{and} \quad \underline{B} = \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b_4 \end{bmatrix} \tag{2.12}$$

For the fully inverted situation one obtains exactly equally structured matrices, except that the nonzero elements have different values. As can be seen in appendix 2, these values depend on the parameter set defined in (2.6). To determine whether the linear system (2.11) is fully controllable at its equilibrium points, the values of the parameter set are needed. In section 4.1 it will be shown that with the measured parameter set the system is fully controllable at its equilibrium points.

It now remains to find a non-linear controller that is capable of swinging up the system to the neighbourhood of its equilibrium point. The strategy is to find a controller that is capable of controlling the first link accurately and at the same time excite the second link enough so that it swings up.
From (2.8) one obtains

$$\ddot{q}_2 = \frac{-d_{21}\ddot{q}_1 - c_{21}\dot{q}_1 - G_2}{d_{22}} \tag{2.13}$$

Substituting (2.13) into (2.7) gives

$$\left(d_{11} - \frac{d_{12}d_{21}}{d_{22}}\right)\ddot{q}_1 + \left(c_{11} - \frac{d_{12}c_{21}}{d_{22}}\right)\dot{q}_1 + c_{12}\dot{q}_2 + G_1 - \frac{d_{12}}{d_{22}}G_2 = \tau_1$$

If τ_1 is now chosen as

$$\tau_1 = \left(d_{11} - \frac{d_{12}d_{21}}{d_{22}}\right)v_1 + \left(c_{11} - \frac{d_{12}c_{21}}{d_{22}}\right)\dot{q}_1 + c_{12}\dot{q}_2 + G_1 - \frac{d_{12}}{d_{22}}G_2 \tag{2.14}$$

This results in

$$\ddot{q}_1 = v_1 \tag{2.15}$$

$$d_{22}\ddot{q}_2 + c_{21}\dot{q}_1 + G_2 = -d_{21}v_1 \tag{2.16}$$

When the new input, v_1, is chosen to be

$$v_1 = \ddot{q}^d + K_d\left(\dot{q}_1^d - \dot{q}_1\right) + K_p\left(q_1^d - q_1\right) \tag{2.17}$$

with $q_1^d(t)$ a – to be determined – reference path. Then asymptotic stability of $q_1^d(t)$ is guaranteed because it can be shown that the error defined as $e_1(t) = q_1(t) - q_1^d(t)$ will go to zero for $t \to \infty$.

It now remains to find a suitable reference path $q_1^d(t)$ that swings the system to the vicinity of its equilibrium points. For the fully inverted equilibrium point a step function is used. Before this step function takes action, the system is first actuated with an open loop input in the wrong direction, while at the same time pumping up the control demand for the non-linear controller. After a small period of time, the open loop input is released and the non-linear controller takes over. Finding the right magnitude and period of the open loop input is a matter of trial and error.

The reference path for the semi-inverted position is somehow more difficult. In [2] it is suggested to take a reference path equal to

$$\begin{aligned} q_1^d(t) &= a\sin(\omega t) - \frac{\pi}{2} \quad , \qquad 0 \le t \le \frac{2\pi}{\omega} \\ q_1^d(t) &= \frac{\pi}{2} \quad , \qquad t > \frac{2\pi}{\omega} \end{aligned} \tag{2.18}$$

Also in this case the amplitude and frequency of the reference path need to be designed by means of trial and error. The reference path is now given for a time up to $t = \frac{2\pi}{\omega}$, so that one full period of oscillation can be executed. After this time, the reference path becomes a point, namely the equilibrium point $q = \frac{\pi}{2}$. Again when the amplitude is not chosen properly the linear controller is not able to catch over the controlling action.

Finally there are some conditions to fulfil for the linear controller to take over. These conditions are the following

Semi-inverted position

- Link one must be within a range of 0.2 radians from $q_1 = -\frac{\pi}{2}$
- Link two must be within a range of 0.3 radians from $q_2 = \pi$
- The input needed for the linear controller to take over must be not too high, for instance less than 80 A (10 A is the upper saturation limit). In fact this imposes a restriction on $\dot{q}_1$ and $\dot{q}_2$, since these are included in the linear control law.

Fully inverted position

- Link one must be within a range of 0.2 radians from $q_1 = \frac{\pi}{2}$

- Link two must be within a range of 0.3 radians from $q_2 = 0$
- The input needed for the linear controller to take over must be not too high, for instance less than 120 A. (The control torque needed for the fully inverted position is larger than for the semi-inverted position). In fact this imposes a restriction on $\dot{q}_1$ and $\dot{q}_2$, since these are included in the linear control law.

2.3 Reaction wheel

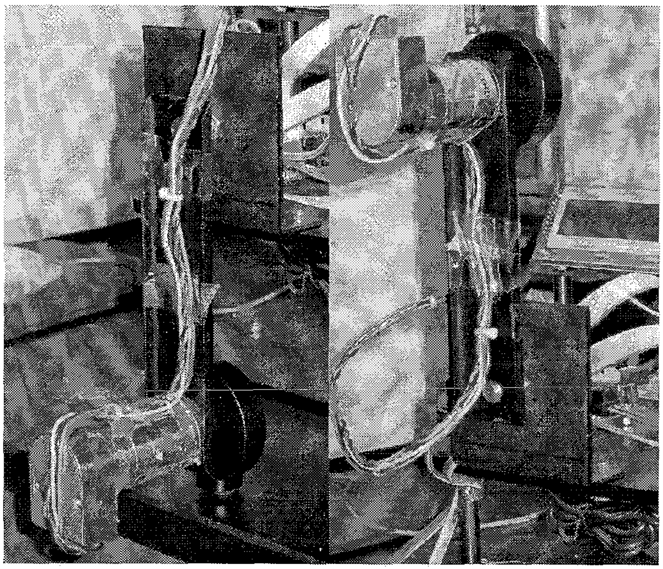


Figure 2.5: a) RW initial condition b)RW inverted position

A picture of the *reaction wheel* configuration can be seen in figure 2.5. In this configuration there exists one actuator but the torque of the motor influences both elements (pendulum and inertia wheel) directly, because Newton's third law ("action is minus reaction") has to be fulfilled.

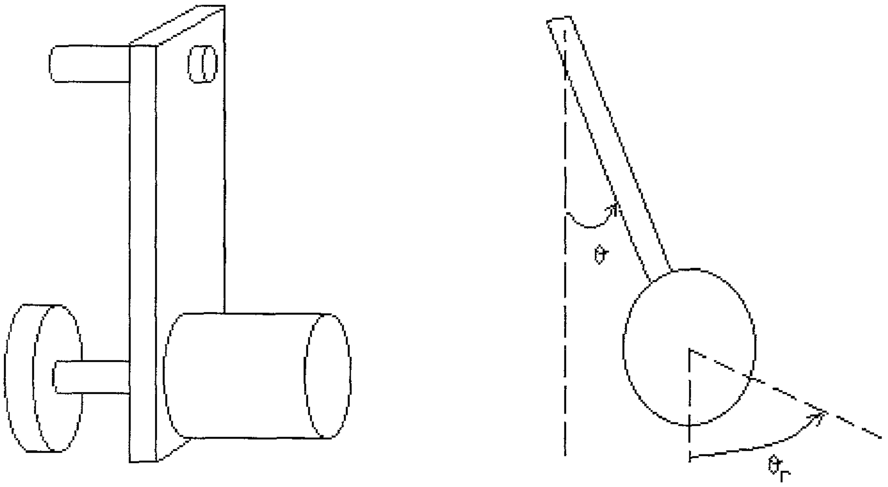


Figure 2.6: Schematic drawing of the reaction wheel

Considering figure 2.6 and taking

m_1 : mass of the pendulum (plus motor) (kg)
l_{c1} : centre of the pendulum (plus motor) (m)
L_1 : length of the pendulum (m)
J_1 : inertia of the pendulum (plus motor) around its centre of mass (kgm^2)
m_2 : mass of the inertiawheel (kg)
l_{c2} : centre of mass of the inertiawheel (equal to L_1) (m)
J_2 : inertia of the inertia wheel (kgm^2)
$q_1 = \theta$: first generalised coordinate representing the angle of the pendulum with respect to the downward vertical (rad)
$q_2 = \theta_r$: second generalised coordinate representing the angle of the inertia wheel with respect to the downward vertical (rad)
τ : Input (Nm)

the derivation of the equations of motion of this configuration can be found in appendix 2. The equations of motion are:

$$p_1\ddot{\theta} + p_3 g \sin(\theta) = -\tau \tag{2.19}$$

$$p_2\ddot{\theta}_r = \tau \tag{2.20}$$

with

$$\begin{aligned} p_1 &= J \\ p_2 &= J_3 \\ p_3 &= ml \end{aligned} \tag{2.21}$$

then, consequently,

$$\begin{aligned} J &= J_1 + m_1 l_{c1}^2 + m_2 L_l^2 \\ ml &= m_1 l_{c1} + m_2 L_1 \end{aligned}$$

Again p_1 through p_3 is the base parameter set. The goal of the *reaction wheel* configuration is stabilization at the inverted position.

Just as in the case of the *pendubot*, a switching strategy is used. First the system is swung up near its equilibrium point by means of a non-linear controller, and when several conditions are fulfilled, a linear controller takes over. First the non-linear controller is derived, its stability is shown, and then the matrices of the linear controller are computed.

For the swing-up phase a passivity-based energy controller is used.
In chapter 5 of [1] it is explained that when a system is passive, infinite gain can be applied in the controller but still the system will not grow unstable. In [8] it is explained that a system is passive when the change of energy inside the system is not larger than the energy that was put into the system through the input. This rule comes down to the following statement

$$\dot{S} \le u^T y \rightarrow S(t) - S(0) \le \int_0^t u^T(s) y(s) ds \tag{2.22}$$

Here, S is a storage function, which is typically taken to be the total energy of the system (kinetic plus potential energy). Further u represents the input, and y represents the output. A simple single mass, for example, is not passive with respect to the position, but it is passive with respect to the speed.
In [1], it is also explained that the non-passive system under consideration, the *reaction wheel*, can be split into two passive parallel parts.

Applying the above properties of passive systems, it is desirable to split up the reaction wheel into two passive systems (because it is not passive yet). Taking (2.20) it is clear that when the output is chosen to be $y_1 = \dot{\theta}_r$, then the system is passive:

$$S_1 = \frac{1}{2} p_2 \dot{\theta}_r^2 \Rightarrow \dot{S}_1 = \dot{\theta}_r \ddot{\theta}_r = u y_1 \le u y_1 \tag{2.23}$$

Taking (2.19) one obtains

$$S = \frac{1}{2} p_1 \dot{\theta}^2 + p_3 g (1 - \cos(\theta)) \Rightarrow \dot{S} = -\dot{\theta} \cdot u \tag{2.24}$$

So when the output is chosen to be equal to $\dot{\theta}$ then also the pendulum system is passive. It is much more interesting however, to take another storage function. The reason for doing this will become clear in the sequel. Take for example

$$S_2 = \frac{1}{2} \left(E - E_{ref} \right)^2 \tag{2.25}$$

E represents the total energy of the pendulum and is equal to $E = \frac{1}{2} p_1 \dot{\theta}^2 + p_3 g (1 - \cos(\theta))$, and E_{ref} is a constant representing a reference energy level, which can be used to determine a reference point for control.
Then it is clear that

$$\dot{S}_2 = -\left(E - E_{ref} \right) \dot{\theta} \cdot u \tag{2.26}$$

Obviously, together with this storage function, the output function is chosen to be

$$y_2 = -\left(E - E_{ref} \right) \dot{\theta} . \tag{2.27}$$

Combining the two systems gives

$$y = k_v y_1 + k_e y_2 \tag{2.28}$$

$$S = k_v S_1 + k_e S_2 = \frac{1}{2} k_v \dot{\theta}_r^2 + \frac{1}{2} k_e \left(E - E_{ref}\right)^2 \tag{2.29}$$

$$\dot{S} = \left(k_v \dot{\theta}_r - k_e \left(E - E_{ref}\right)\dot{\theta}\right) \cdot u \tag{2.30}$$

If the input is now chosen to be

$$u = -k_u y \tag{2.31}$$

then the derivative of the storage function equals

$$\dot{S} = -k_u y^2 \leq 0 \tag{2.32}$$

When the original fourth order system ((2.19) and (2.20)) is approached by a third order system, and thus neglecting the position of the wheel (which is justified due to symmetry), one can consider S being a Lyapunov function. From (2.32) it can be concluded that the time derivative of this Lyapunov function can be zero under certain conditions. It therefore remains to determine what sets are asymptotically stable, and hence LaSalle's principle is applied.

The dynamics (2.19), (2.20) are reduced to

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{p_3}{p_1} g \sin(x_1) - \frac{u}{p_1} \qquad , \; u = -k_u y \\ \dot{x}_3 &= \frac{u}{p_2} \end{aligned} \tag{2.33}$$

with $x_1 = \theta$, $x_2 = \dot{\theta}$, $x_3 = \dot{\theta}_r$.

LaSalle states the following. Suppose there exists a bounded (though not necessarily small) space $\Omega_c = x \in \Re^3$, then there exists a set Z, such that $Z = \left\{x \in \Omega_c : \dot{S} = 0\right\}$. If H is the largest invariant set with respect to the dynamics in Z, then it is guaranteed that solutions starting at $x_0 \in \Omega_c$ will approach to the set H if time goes to infinity. The set H is invariant if f_{cl} (vector field on the right hand side of (2.33)) evaluated at H, lies in the tangent plane of Z, evaluated at H.

Because in (2.33) $\theta = x_1$, $\dot{\theta} = x_2$, $\dot{\theta}_r = x_3$, consequently the storage function becomes

$$S = \frac{1}{2} k_v p_2 x_3^2 + \frac{1}{2} k_e \left(E - E_{ref}\right)^2 , \qquad E = \frac{1}{2} p_1 x_2^2 + p_3 g \left(1 - \cos(x_1)\right) \tag{2.34}$$

Setting $\dot{S} = 0$ gives the set Z. Here it is chosen to express x_3 as a function of x_1 and x_2, which gives:

$$x_3 = \frac{k_e}{k_v} x_2 \left\{ \frac{1}{2} p_1 x_2^2 + p_3 g\left(1 - \cos(x_1) - E_{ref}\right) \right\} \tag{2.35}$$

The tangent plane to the set Z is equal to (x is replaced by z)

$$T_Z M = \left\{ \begin{bmatrix} 1 \\ 0 \\ \frac{k_e}{k_v} p_3 g z_2 \sin(z_1) \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ \frac{k_e}{k_v}\left(\frac{3}{2} p_1 z_2^2 + p_3 g\left(1 - \cos(z_1) - E_{ref}\right)\right) \end{bmatrix} \right\} = \{TM(1), TM(2)\}$$

Now, evaluating (2.33) at the set Z gives

$$f_{cl} = \begin{bmatrix} z_2 \\ -\frac{p_3}{p_1} g \sin(z_1) \\ 0 \end{bmatrix} \tag{2.36}$$

When $f_{cl} = \alpha TM(1) + \beta TM(2)$ is evaluated, this gives $\alpha = z_2$, $\beta = -\frac{p_3}{p_1} g \sin(z_1)$, so that

$$f_{cl} = \alpha TM(1) + \beta TM(2) = \begin{bmatrix} z_2 \\ -\frac{p_3}{p_1} g \sin(z_1) \\ -\frac{k_e p_3 g \sin(z_1)}{2k_v}\left[z_2^2 + \frac{2}{p_1}\left(p_3 g - E_{ref}\right) - \frac{2 p_3 g \cos(z_1)}{p_1} \right] \end{bmatrix} \tag{2.37}$$

In order for (2.37) to be equal to (2.36), the third component of the vector field indicated on the right hand side of (2.37) must be zero. The set of points that satisfy this condition are:

$$z_1 = 0 + k\pi \qquad k = 0,1,\ldots,\infty \qquad \text{set1}$$
$$z_2 \in \Re$$

$$z_1 \in \Re \qquad \text{set2}$$
$$z_2 = \sqrt{\frac{2}{p_1}\left(p_3 g \cos(z_1) - p_3 g + E_{ref}\right)}$$

$$z_1 \in \mathfrak{R} \qquad \text{set3}$$
$$z_2 = -\sqrt{\frac{2}{p_1}\left(p_3 g \cos(z_1) - p_3 g + E_{ref}\right)}$$

Because $f_{cl}(set_i)$ needs to be invariant, $f_{cl}(set_i)$ needs to be in the direction of the tangent space to set_i. Therefore the vector fields that span the tangent spaces, $\frac{\partial set_i}{\partial z_i}$ have to be computed.

$$\frac{\partial set_1}{\partial z_2} = \begin{bmatrix} 0 \\ 1 \\ \frac{k_e}{k_v}\left(p_1 z_2 - E_{ref}\right) \end{bmatrix} \tag{2.38}$$

$$\frac{\partial set_2}{dz_1} = \begin{bmatrix} 1 \\ -\dfrac{p_3 g \sin(z_1)}{p_1\sqrt{\frac{2}{p_1}\left(p_3 g \cos(z_1) - p_3 g + E_{ref}\right)}} \\ 0 \end{bmatrix} \tag{2.39}$$

$$\frac{\partial set_3}{dz_1} = \begin{bmatrix} 1 \\ \dfrac{p_3 g \sin(z_1)}{p_1\sqrt{\frac{2}{p_1}\left(p_3 g \cos(z_1) - p_3 g + E_{ref}\right)}} \\ 0 \end{bmatrix} \tag{2.40}$$

Evaluating f_{cl} at set_i gives

$$f_{cl}(set_1) = \begin{bmatrix} z_2 \\ 0 \\ 0 \end{bmatrix} \tag{2.41}$$

$$f_{cl}(set_2) = \begin{bmatrix} \sqrt{\frac{2}{p_1}\left(p_3 g \cos(z_1) - p_3 g + E_{ref}\right)} \\ -\frac{p_3 g}{p_1}\sin(z_1) \\ 0 \end{bmatrix} \tag{2.42}$$

$$f_{cl}(set_3) = \begin{bmatrix} -\sqrt{\frac{2}{p_1}\left(p_3 g\cos(z_1) - p_3 g + E_{ref}\right)} \\ -\frac{p_3 g}{p_1}\sin(z_1) \\ 0 \end{bmatrix} \tag{2.43}$$

Now for each set the following equation should hold, for some α

$$f_{cl}(set_i) = \alpha_i \frac{\partial set_i}{\partial z_i} \tag{2.44}$$

This manipulation gives

$$\alpha_1 = 0 \tag{2.45}$$

$$\alpha_2 = \sqrt{\frac{2}{p_1}\left(p_3 g\cos(z_1) - p_3 g + E_{ref}\right)} \tag{2.46}$$

$$\alpha_3 = -\sqrt{\frac{2}{p_1}\left(p_3 g\cos(z_1) - p_3 g + E_{ref}\right)} \tag{2.47}$$

Furthermore (2.44) is only valid if $z_2 = 0$ for set_1, leaving the origin as the first set. So it can be concluded that the largest invariant set is the union of the origin, set_2 and set_3. Every solution starting in Ω_c will converge to this invariant set.

It is interesting to check whether the origin and the vertical position are saddle points, meaning that there are some solutions approaching these points asymptotically, and some solutions flowing away from the points. If this is the case, the stable two sets can be considered to be homoclinic orbits, meaning that the orbits connect the equilibrium points. In order to check whether the equilibrium points are saddle points, the eigenvalues of the linear approximation at these points need to be computed. If there are no eigenvalues on the imaginary axis then the Grobman Hartmann theorem states that the non-linear dynamics may be approximated by the linear dynamics. If in addition some eigenvalues are negative and some of them are positive, the equilibrium point is a saddle point, see [8].

The linearization of the non-linear equations of motion at the origin $x = [0 \quad 0 \quad 0]^T$ gives

$$\dot{\underline{z}} = \underline{A}\underline{z} \tag{2.48}$$

with

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 \\ a_1 & a_2 & a_3 \\ 0 & a_4 & a_5 \end{bmatrix} \quad \text{with} \quad \begin{aligned} a_1 &= -\frac{p_3 g}{p_1} \\ a_2 &= \frac{2k_u k_e g p_3}{p_1} \\ a_3 &= \frac{k_u k_v}{p_1 p_3} \end{aligned} \qquad \begin{aligned} a_4 &= -\frac{2k_u k_e g p_3}{p_2} \\ a_5 &= -\frac{k_u k_v}{p_2 p_3} \end{aligned}$$

Taking the parameters that will be derived in chapter 3, gives

$$\begin{aligned} \lambda_1 &= 2 \cdot 10^5 \\ \lambda_2 &= 0.12 \\ \lambda_3 &= -0.12 \end{aligned} \tag{2.49}$$

Linearizing around $[\pi \quad 0 \quad 0]^T$ gives

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 \\ a_1 & 0 & a_2 \\ 0 & 0 & a_3 \end{bmatrix} \quad \text{with} \quad \begin{aligned} a_1 &= \frac{p_3 g}{p_1} \\ a_2 &= \frac{k_u k_v}{p_1 p_3} \\ a_3 &= -\frac{k_u k_v}{p_2 p_3} \end{aligned} \tag{2.50}$$

Taking the parameters that will be derived in chapter 3, gives eigenvalues of

$$\begin{aligned} \lambda_1 &= 8.85 \\ \lambda_2 &= -8.85 \\ \lambda_3 &= -44.44 \end{aligned} \tag{2.51}$$

Hence it can be concluded that the vertical equilibrium point is a saddle point, and the stable sets set_2 and set_3 are homoclinic orbits, connecting this saddle point to itself. The combined homoclinic orbit is plotted in figure 2.7. Nothing much can be concluded from (2.49) because two eigenvalues are near zero while one is extremely high. Strictly speaking there exists a saddle point, but because of the rather bad mechanics, possible properties of saddle points can not be used. Figure 2.8 shows a simulation starting near the origin. As can be seen, the solution indeed converges to the orbit. This was already expected because the orbit is contained in the largest invariant set. Because the state only approaches the orbit and only reaches it in infinite time, the third component, x_3, will not be zero, but it will fluctuate around zero.

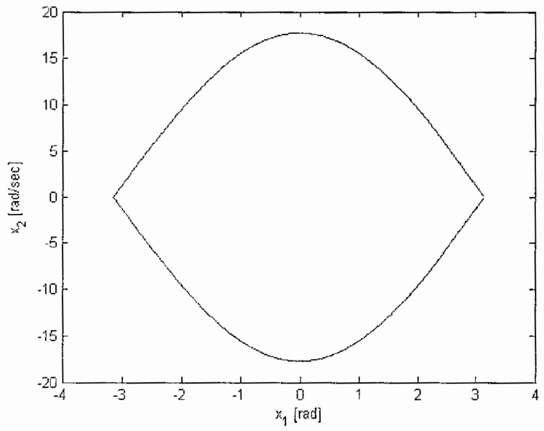


Figure 2.7: Homoclinic orbit by LaSalle analysis

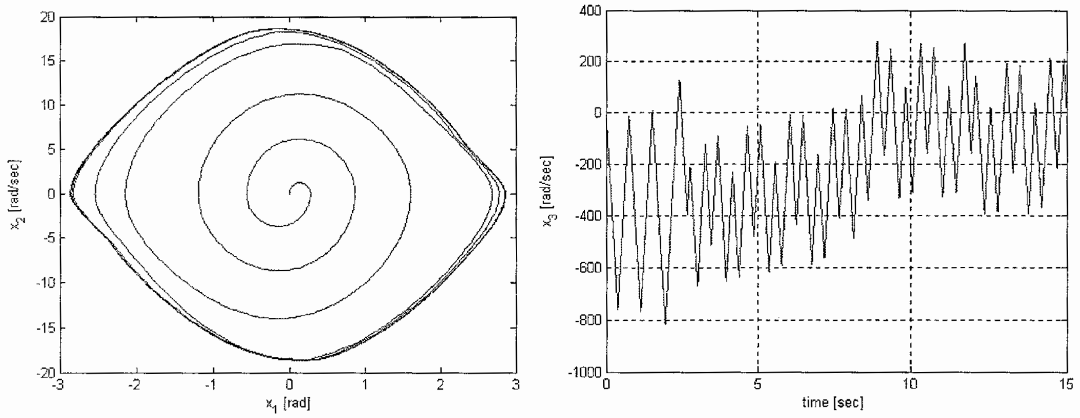


Figure 2.8: Homoclinic orbit by simulation. Left: phaseplot(x1,x2), right: x3

Since with this controller the inverted position, figure 2.5b, is not a stable equilibrium point, the strategy is to swing the system up to the equilibrium point, and let the linear controller take over, under appropriate conditions.

Linearizing the system at the "up" vertical equilibrium point (without controller) gives

$$\underline{\dot{z}} = \underline{A}\underline{z} + \underline{B}u \tag{2.52}$$

with

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{p_3}{p_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \underline{B} = \begin{bmatrix} 0 \\ -\frac{1}{p_1} \\ 0 \\ \frac{1}{p_2} \end{bmatrix}$$

It is desirable to avoid transients between the switching from the non-linear to the linear controller. Therefore several conditions have to be fulfilled between the pole placement of the linear controller and the controller gains of the non-linear controller.

Since the energy level of the equilibrium point of interest is equal to $E = 2p_3g$, which is the situation of zero kinetic energy and maximum potential energy, referring to figure 2.5b, the following condition should hold

$$E = \frac{1}{2}J\dot{\theta}^2 + p_3g(1-\cos(\theta)) = 2p_3g \Rightarrow \dot{\theta} = \omega_p\sqrt{2(1+\cos(\theta))} \tag{2.53}$$

with $\omega_p = \sqrt{\frac{p_3g}{p_1}}$ (2.54)

Taking the linear approximation of (2.53) at $\theta = \pi$ gives

$$\dot{\theta} = -\omega_p(\theta - \pi) \tag{2.55}$$

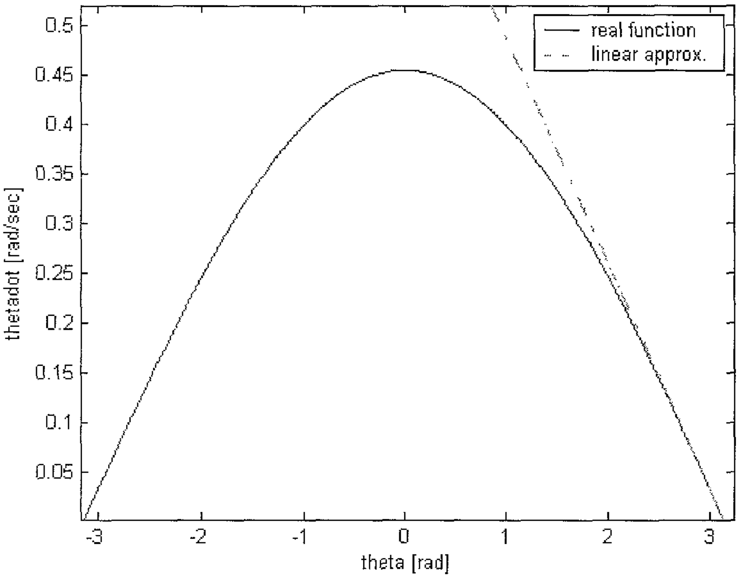


Figure 2.9: Linearization of $\dot{\theta}$ *at* $\theta = \pi$

Because in the non-linear controller the position and velocity of the pendulum and the velocity of the wheel are used in the feedback law, it is convenient to use these states in the linear controller, giving

$$u = -k_{pp}\theta - k_{dp}\dot{\theta} - k_{dr}\dot{\theta}_r \tag{2.56}$$

Using this control law with the linearized system evaluated at the equilibrium point gives

$$\underline{\dot{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \dfrac{p_3 g + k_{pp}}{p_1} & \dfrac{k_{dp}}{p_1} & 0 & \dfrac{k_{dr}}{p_1} \\ 0 & 0 & 0 & 1 \\ -\dfrac{k_{pp}}{p_2} & -\dfrac{k_{dp}}{p_2} & 0 & -\dfrac{k_{dr}}{p_2} \end{bmatrix} \underline{x} = \underline{A}\underline{x} \tag{2.57}$$

An eigenspace (the space spanned by an eigenvector) is, in a way, the linear version of an invariant trajectory. Therefore the trajectories of this linear system will match the homoclinic orbit if the matrix $\underline{A}$ has an eigenvalue ω_p associated with the eigenvector (see equation 2.55))

$$\underline{v} = \begin{bmatrix} 1 \\ -\omega_p \\ 0 \\ 0 \end{bmatrix} \tag{2.58}$$

Solving the eigenvector problem with respect to the eigenvector (2.58) results in

$$\lambda = -\omega_p \tag{2.59}$$

$$k_{pp} = \omega_p k_{dp} \tag{2.60}$$

Also with this switching controller there exist a condition that needs to be fulfilled for the linear controller to take over. This conditions is

$$|q_1 - \pi| < 0.5rad \tag{2.61}$$

2.4 *Furuta Pendulum*

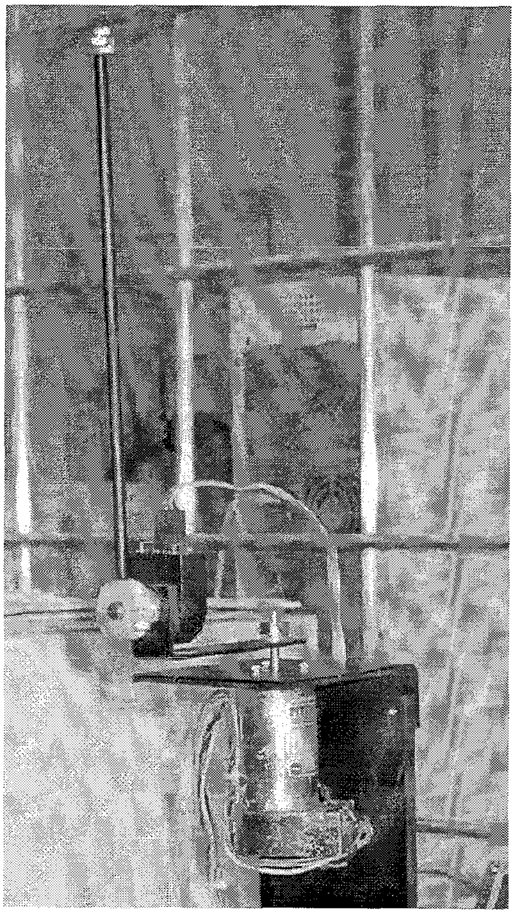


Figure 2.10: Furuta at the "Base frame position"

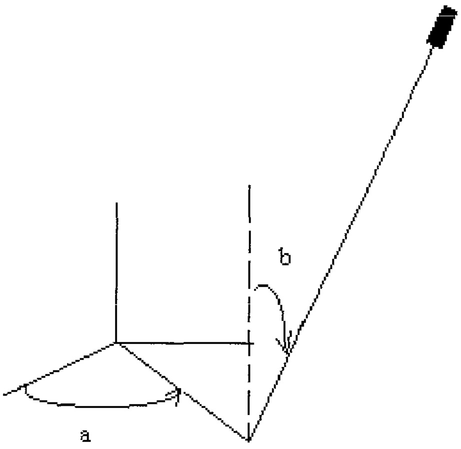


Figure 2.11: Schematic drawwing of the furuta

A picture of the *Furuta Pendulum* configuration can be found in figure 2.10. A schematic drawing can be seen in figure 2.11. Let

m_1	:	mass of the first link (kg)
l_{c1}	:	centre of mass of the first link (m)
L_1	:	length of the first link (m)
J_1	:	inertia of the first link with respect to its centre of mass (kgm^2)
m_2	:	mass of the second link (kg)
l_{c2}	:	centre of mass of the second link (m)
J_2	:	inertia of the second link with respect to its centre of mass (kgm^2)
a	:	first generalised coordinate, representing the angle of the first link with respect to the forward pointing horizontal (rad)
b	:	second generalised coordinate representing the angle of the second link with respect to the upward vertical (rad)
τ	:	input (Nm)
f	:	friction term (Nm)

then the equations of motion are derived in appendix 1, and the main results are the following.

$$\underline{D}\underline{\ddot{q}}+\underline{C}\underline{\dot{q}}+\underline{G}=\underline{\tau} \tag{2.62}$$

with

$$\underline{D}=\begin{bmatrix} p_1+4p_3\sin^2(b) & \frac{1}{2}p_2\cos(b) \\ \frac{1}{2}p_2\cos(b) & 4p_3 \end{bmatrix}$$

$$\underline{C}=\begin{bmatrix} 4p_3\sin(2b)\dot{b} & -p_2\sin(b)\dot{b}+4p_3\sin(2b)\dot{a} \\ -4p_3\sin(2b)\dot{a} & 0 \end{bmatrix}$$

$$\underline{G}=\begin{bmatrix} 0 \\ -p_4 g\sin(b) \end{bmatrix}$$

$$\underline{\tau}=\begin{bmatrix} u-f \\ 0 \end{bmatrix}$$

$$f=p_5\cdot\dot{a}+p_6\cdot\left(1-\frac{2}{\exp(2\cdot k\cdot\dot{a})+1}\right)$$

and

$$\begin{aligned} p_1 &= \frac{1}{2}J_1+\frac{1}{2}m_1 l_{c1}+\frac{1}{2}m_2 L_1^2 \\ p_2 &= m_2 L_1 L_2 \\ p_3 &= J_2 \\ p_4 &= m_2 l_{c2} \end{aligned}$$

Obviously the friction in the second link is neglected. Its value is negligible compared to the value of the motor's friction.
Because of the restriction on time, no effort is made in development or analysis of existing control laws. In [3] a suggestion for an energy-based controller is made.

Chapter 3 Identification

As can be concluded from chapter 2, the control laws all need the values of the parameters in order to function satisfactorily. Therefore much effort is spent in identifying the systems. As will be seen this is not very trivial. The very crucial fact about identifying systems is that the input function needs to be "exciting enough". Thus not every excitation function will work satisfactorily. For a given system, an excitation function may work very well, while for another system the same function might just be useless. In this chapter first the methods are explained, then a very simple rigid pendulum will be used to demonstrate the power of the identification algorithm that is being used for the underactuated systems as well. Then the algorithm is applied for the underactuated systems. The ultimate goal is to identify the parameters by applying an input and measuring positions of the links. When this is not possible, the parameters are identified by means geometric and mass measurements.

3.1 Identification algorithms

In literature there exist several methods for estimating the parameters that are needed for the control laws. There exists an energy-based identification method based on the principle that the input energy should be equal to the change of internal energy. Although this sounds very reasonable, the method is not found to be working well. Swevers et al. [5], from now on to be called Swevers, on the other hand, suggests a much more powerful method. In order to use this method the system's equations of motion need to be linear in the parameter set. As can be seen from chapter 2, this is the case. If this were not the case, the system could be transformed into this form by means of for instance the DH-transformation, because parameters taken with respect to the relative frames of the links and translated to the base frame will always appear linearly in the resulting equations of motion, see [5] and the additional document to this paper. Although the method of Swevers is found to work properly, it is not suited for the systems of the Mechatronics Kit. Swevers suggests that the variance on the measurements of the joint velocities and accelerations is much lower than the variance on the measurements of the control torque, and so they can be neglected. Although this might sound reasonable for most existing systems that use a current measurement device in order to estimate the voltage over the motor, this is not the case for the Mechatronics Kit. When identifying the system, one gives a reference control signal, which is not a function of the actual position or velocity of the links. Therefore the reference control signal will be perfectly smooth. The motor of the Mechatronics Kit is provided with an external controller which makes sure that the reference control signal is achieved. The control signal that is being "measured" is therefore exactly equal to the reference control signal, and does not contain any variance. Hence the assumption of Swevers cannot be justified.
Olsen et al [4], from now on to be called Olsen, on the other hand, suggests taking as much as information as possible out of the measurements in order to get the best parameters. He therefore uses a full covariance matrix.

Finally there exist two other methods, the weighted least squares and the linear least squares method. In a sense Swevers, Olsen, weighted, and linear least squares all work with the same algorithm. The only difference is in one of the matrices they use, as is explained in [4]. The energy based identification works significantly different, but as already mentioned does not give satisfactory results. The working principle of the energy based identification method is explained in appendix 5.

3.1.1 Identification by means of minimization

When we take a system that consists of one degree of freedom, for instance a pendulum, we have to write the equations of motion (in this case only one) in a form so that the right hand side equals zero. This function is called G_1 and becomes

$$G_1 = J\ddot{\theta} + mgl\sin(\theta) - \tau + f = 0 \tag{3.1}$$

with θ the degree of freedom, τ the input, and f the friction term.
It is well known in the field of identification, and also statistics, that the only correct statistical estimate for parameter estimation, is the Maximum Likelihood estimate. For n independent measurements this ML-estimate is equal to

$$\begin{aligned} &\min \frac{1}{2}\sum_{i=1}^{n}(x_i - \psi_i)^T\left[\sigma_i^2\right]^{-1}(x_i - \psi_i) \\ &\quad \text{s.t. } G_k(p,\psi_i) = 0 \qquad \forall i,k \end{aligned} \tag{3.2}$$

So one is looking for the best possible parameter set p that makes $G_k, k = 1,2$ equal to zero and at the same time minimizes the squared difference between the measurement $x_i \in \Re^d, i = 1,\ldots,n$ and the "real" values $\psi_i \in \Re^d, i = 1,\ldots,n$. In here d represents the number of signals, which in the case of the underactuated systems considered in this study is equal to seven (two positions, two velocities, two accelerations and one input). Of course the "real" values are substituted by the measured averages, corresponding to the $d \times d$ covariance matrix σ_{ii}. Hence

$$x_i = \psi_i + r_i \tag{3.3}$$

where $r_i \in \Re^d, i = 1,\ldots,n$ represents [Gaussian] noise. The system is not working when the noise does not have a normal distribution, see [4].
This minimization problem may be solved by means of, for instance, the optimization toolbox of *Matlab.* In order to do so, one has to compute second and even third order derivatives in order to use the Newton-Raphson method that comes along with the optimization routine. Because the identification algorithm is based on large amounts of data, to average out noise, the full computation in this way demands far too much time. Therefore Olsen suggests taking the linear approximation of $G_k(p,\psi_i)$, which is equal to

$$G_k(p,\psi_i) \cong G_k(p,x_i) + \left[\nabla_{x_i} G_k(p,x_i)\right]^T(\psi_i - x_i) \tag{3.4}$$

Fixing p and taking $y_i = \psi_i - x_i$ gives a set of independent optimization problems

$$\begin{aligned} &\min \frac{1}{2} y_i^T \left[\sigma_i^2\right]^{-1} y_i \\ &\left[B^{(i)}(p)\right] y_i = b^{(i)}(p) \end{aligned} \tag{3.5}$$

with the k-th row of the $m \times d$ matrix $B^{(i)}$ containing $\left[\nabla_{x_i} G_k(p, x_i)\right]$ and the k-th element of $b^{(i)}$ is equal to $-G_k(p, x_i)$.

Then this problem has an analytical solution, see [4], that is equal to

$$y_i^T y_i = \sigma_i^2 \left[b^{(i)}(p)\right]^T \left\{\left[B^{(i)}(p)\right] \sigma_i^2 \left[B^{(i)}(p)\right]^T\right\}^{-1} b^{(i)}(p) \tag{3.6}$$

Hence the optimization problem can be reduced to an optimization of only one function in the parameter vector p

$$\min \frac{1}{2} \sum_{i=1}^{n} b^T \left\{B \sigma_i^2 B^T\right\}^{-1} b = \min \frac{1}{2} \sum_{i=1}^{n} b^T \left[S_{ml}\right]^{-2} b = \min \phi_{ml} \tag{3.7}$$

$$\left[S_{ml}\right]^2 = B \sigma_i^2 B^T \tag{3.8}$$

ϕ_{ml} is called the object function

The object function from Swevers is exactly the same, except for the fact that the covariance matrix now has zero elements on the places corresponding to joint velocity and acceleration.
In the weighted least squares estimator the matrix $\left[S_{ml}\right]^2$ now only contains the diagonal elements of the full $\left[S_{ml}\right]^2$ matrix of the Olsen object function. So this estimator neglects the information between the first en second equation of motion.
In the linear least squares estimator, the matrix $\left[S_{ml}\right]^2$ is just the identity matrix.

Finally it is noted that because of the linearization in (3.4), errors can be expected when the magnitude of the variances on the difference between the measurements and the "real" values are large.

(3.7) Is now ready to be implemented in *Matlab* by means of the function *fmincon.*

3.1.2 Optimization of the excitation trajectory

In the deterministic framework there exists an optimal exciting trajectory (in order to produce a minimal estimation error) for any given system. As can be seen in [5], the optimization criterion for the estimation of parameters in the deterministic framework is the condition number of the scaled regressor matrix. Each column of the regressor

matrix is scaled by its norm, and after the estimation procedure the parameters are scaled back, so that the right values of the parameters are obtained. When the condition number of this scaled regressor matrix equals one, there exists an optimal estimation. The optimization of the excitation trajectory optimizes the systems output in order to get the lowest possible condition number. As parameters for optimization there are the amplitudes of base harmonics plus an offset parameter for the position. When, for example, 5 base harmonics are chosen, one obtains 11 parameters to optimize, thus 10 for the harmonics and 1 offset parameter for the position.
In theory one wants the angle, velocity and acceleration of one degree of freedom to be equal to

$$q(t)=\sum_{l=1}^{N}\frac{a_l}{\omega_f l}\sin(\omega_f lt)-\frac{b_l}{\omega_f l}\cos(\omega_f lt)+q_0 \tag{3.9}$$

$$\dot{q}(t)=\sum_{l=1}^{N}a_l\cos(\omega_f lt)+b_l\sin(\omega_f lt) \tag{3.10}$$

$$\ddot{q}(t)=\sum_{l=1}^{N}-a_l\omega_f l\sin(\omega_f lt)+b_l\omega_f l\cos(\omega_f lt) \tag{3.11}$$

N is determined by the user and is often chosen to be three or five. When an initial guess of the parameters is known one can compute the torque needed to give the optimal response of the system, thus resulting in a minimal condition number, because

$$\underline{\tau}=\underline{\phi}\,\underline{p} \tag{3.12}$$

Where $\underline{p}$ is the initial parameter guess and $\underline{\phi}$ is the regressor matrix, corresponding to the trajectory of (3.9), (3.10) and (3.11). Before illustrating the power of this identification method by a simple example, it should be noted that one is actually not dealing with a deterministic system, and thus the condition number of the scaled regressor matrix is the wrong optimization criterion. However, as will be seen for the underactuated systems, optimizing the trajectory this way is not very useful, so not much effort is made in trying to compute the real criterion, which is the Fisher information matrix. For more information on this criterion, see [5].

3.2 Illustration of the method of identification through optimization

The example under consideration is a single pendulum. Clearly the equation of motion for this system is

$$J\ddot{q}+mgl\sin(q)=\tau-f \qquad , f=a\dot{q}+b\left(1-\frac{2}{\exp(30\dot{q})+1}\right) \tag{3.13}$$

Suppose it is desired to estimate J, mgl, a and b, from now on to be called p_1, p_2, p_3 and p_4 respectively, then the regressor matrix becomes

$$\underline{\phi} = \left[\ddot{q} \quad \sin(q) \quad \dot{q} \quad 1 - \frac{2}{\exp(30\dot{q})+1} \right] \tag{3.14}$$

In minimizing the condition number of (3.14), one has to take into account that the algorithm will take no caution in initial conditions of the response. In most cases the optimization will result in an optimal response starting at a position that is not the true initial position, and also at an initial velocity and acceleration not equal to zero. Therefore extra constraints have to be built in.

At time $t = 0$ one wants the optimal position to be equal to $q(0) = -\frac{\pi}{2}$, and also at time zero one wants $\dot{q}(0) = 0$ and $\ddot{q}(0) = 0$.
Considering (3.9), (3.10) and (3.11), and taking only two base harmonic functions for the sake of simplicity, this will result in

$$\begin{aligned} q_0 &= -\frac{\pi}{2} + \frac{3a_1}{8\pi\omega_f} \\ a_2 &= -\frac{a_1}{2} \\ b_2 &= -b_1 \end{aligned} \tag{3.15}$$

So indeed there are only two parameters left for optimization, but it is guaranteed that the optimal trajectory will start with at appropriate initial conditions. Performing this optimization in *Matlab* results in a condition number equal to 4.4452. This number is rather close to one, so no effort is made in including more base harmonics in order to get an even lower condition number. The condition number can also decrease a little when the upper bounds for the harmonic components are increased, but this will naturally lead to larger excitations which are not really wanted nor necessary. The resulting optimal trajectory is plotted in figure 3.1, showing position, velocity and acceleration from top to bottom.

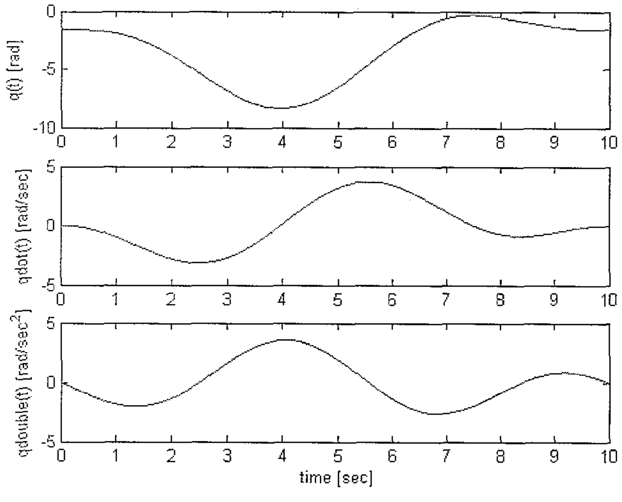


Figure 3.1: Optimal trajectory

The identification method will first be checked in a simulation environment, followed by an experimental test.

Simulation environment

In the simulation environment the "real" parameters can be chosen freely. Then also the initial guess of the parameter vector can be chosen equal to these real parameters. The real parameters are chosen to be in the same order of magnitude as the parameters that represent the experimental setup. Together with (3.12) and the optimal trajectory, this gives the exact input, which is plotted in figure 3.2. Together with this input and the real parameters the simulation data is generated, and afterwards some normally distributed noise is added for two reasons. The first reason is to make things more realistic and the second is that the matrices in the Olsen procedure (as well as Swevers and weighted least squares) cannot handle zero variance on measurements. Matrices will then become singular. The matrices for the Olsen routine that have to be constructed become the following

$$\underline{b} = -p_1\ddot{q} - p_2\sin(q) + \tau - f \tag{3.16}$$

$$\underline{B} = \nabla_q G = \begin{bmatrix} p_2\cos(q) & p_3 + p_4\dfrac{\partial f}{\partial \dot{q}} & p_1 & -1 \end{bmatrix} \tag{3.17}$$

From statistics it is well known that the relation between signals x and y equals

$$\operatorname{cov}(x,y) = \frac{1}{2}\left[\operatorname{var}(x+y) - \operatorname{var}(x) - \operatorname{var}(y)\right] \tag{3.18}$$

Taking this expression one can easily build the covariance matrix

$$\underline{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} \end{bmatrix} \tag{3.19}$$

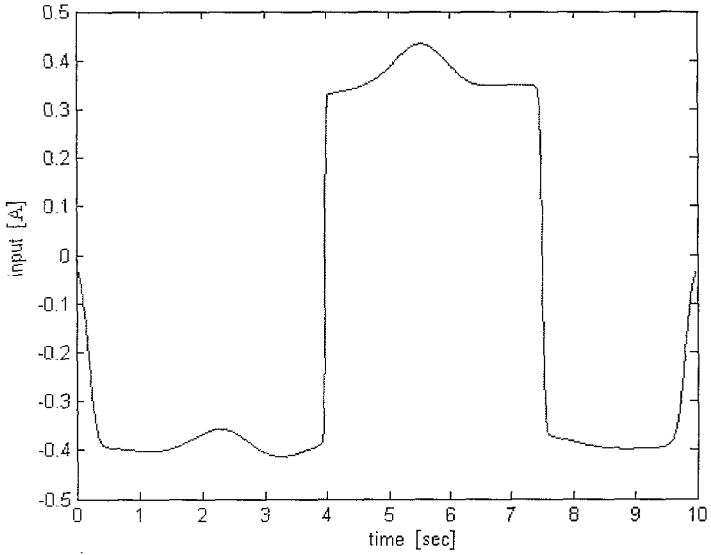


Figure 3.2: Optimal input

If the appropriate equations are implemented in the Olsen optimization routine and the real parameter set p_r is chosen to be

$$p_r = \left[3.06 \cdot 10^{-4} \quad 0.0345 \quad 0.0111 \quad 0.3575\right] \tag{3.20}$$

Then the estimated parameter p_e set is found to be

$$p_e = \left[3.111 \cdot 10^{-4} \quad 0.03447 \quad 0.01108 \quad 0.357507\right] \tag{3.21}$$

and thus the relative error (in %) is equal to

$$e = \left[1.674 \quad 0.0796 \quad 0.1867 \quad 0.0019\right] \tag{3.22}$$

As starting point of the parameters they were all set to one, proving that there is a very wide domain of attraction to the global minimum. Obviously the resulting relative error is small enough to proceed to the experimental setup.

Experimental environment

Recalling (3.12), an initial guess for the parameter vector is needed. This initial guess is obtained from the geometry and mass of the pendulum and its measured inertia. To measure the inertia, the pendulum is taken out of its stable equilibrium point by hand for less than 20 degrees and is then released. The free response is measured, and the resulting frequency should be equal to

$$\omega = \sqrt{\frac{mgl}{J}} \tag{3.23}$$

Performing this experiment several times will average out the measurement error. The resulting frequency is equal to $\omega = 10.62\, rad/\sec$. Next mgl is computed by hand, so that J can be computed as well. Then the initial guess is

$$\begin{aligned} J_0 &= 4 \cdot 10^{-4} \\ mgl_0 &= 0.0345 \end{aligned} \tag{3.24}$$

Putting this initial guess together with the optimal trajectory into (3.12), gives the control signal that has to be applied. Before performing this calculation the dynamic parameters (not the friction parameters) have to be scaled by the constant that translates the control signal to the actuated torque. This constant is found to be equal to $k = 0.004943$. Unfortunately no torque measurement device is available at the laboratory so the correctness of this constant could not be verified. Therefore the parameters that have to be applied in the source-code become

$$\begin{aligned} p_1 &= 0.0624 \\ p_2 &= 7.0408 \end{aligned} \tag{3.25}$$

Taking these parameters gives the input as plotted in figure 3.3.

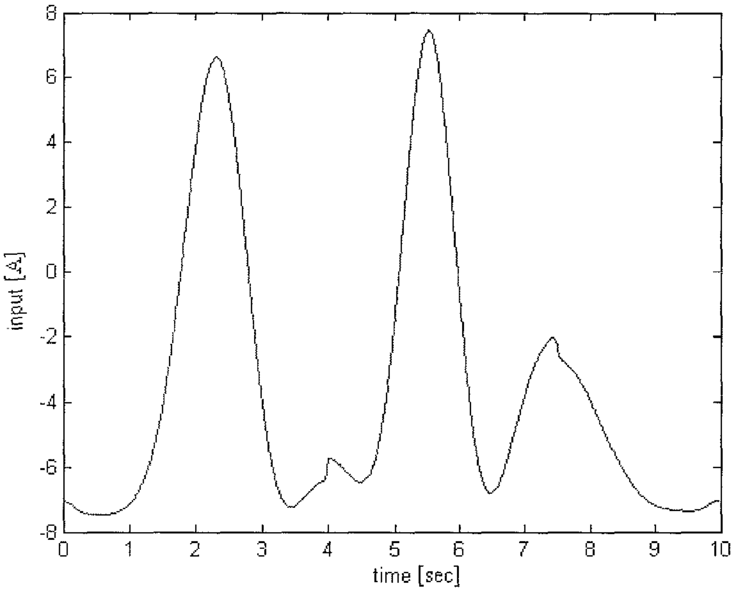


Figure 3.3: Optimal input for the experimental excitation

Unfortunately no time was available to perform the 30 independent measurements with this input. Looking at figure 3.3, one can however conclude that the friction does not play a dominant role anymore, which is the case when you do not scale the inertial parameters by the torque constant. It is even wise to apply a slightly lower torque because with an input of approximately 80 percent of the saturation level the base frame will most probably not be firmly supported on the table anymore.

3.3 Identification of the pendubot

Unfortunately the optimization of the trajectory can not be applied for underactuated systems because the optimization procedure expects that also the second equation of motion, the position, velocity and acceleration, can be realised by means of an independent torque. Since this is not the case, the only thing left to do is finding an appropriate excitation by means of trial and error. There is however still a criterion that says whether a given excitation and the corresponding trajectories are good for estimation or not. Of course this criterion is the same as the one for the fully actuated system, namely the condition number of the scaled regressor matrix. The regressor matrix has dimensions 2 by 7 because the friction parameters have to be estimated as well. The friction of the cables needs to be added to the friction of the motor. Indeed, the cables also add some not negligible stiffness and also some time varying disturbance forces because the cables touch one another and the base plate every now and then. So actually one is trying to capture these two effects by the two friction parameters of the sigmoid function. Larger errors can therefore be expected in the experimental configuration. The regressor matrix for the *pendubot* is then given by

$$\underline{\phi} = \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{14} & F_{15} & F_{16} & F_{17} \\ 0 & F_{22} & F_{23} & 0 & F_{25} & 0 & 0 \end{bmatrix} \tag{3.26}$$

with

$$\begin{aligned}
F_{11} &= \ddot{q}_1 \\
F_{12} &= \ddot{q}_1 + \ddot{q}_2 \\
F_{13} &= \cos(q_2)(\ddot{q}_1 + \ddot{q}_2) - \sin(q_2)(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2) \\
F_{14} &= g\cos(q_1) \\
F_{15} &= g\cos(q_1 + q_2) \\
F_{16} &= \dot{q}_1 \\
F_{17} &= 1 - \frac{2}{\exp(30\dot{q}_1) + 1} \\
F_{22} &= \ddot{q}_1 + \ddot{q}_2 \\
F_{23} &= \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1^2 \\
F_{25} &= g\cos(q_1 + q_2)
\end{aligned}$$

The matrix B is equal to

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} & -1 & B_{15} & B_{16} & B_{17} \\ B_{21} & B_{22} & B_{23} & 0 & B_{25} & 0 & B_{27} \end{bmatrix} \tag{3.27}$$

with

$$\begin{aligned}
B_{11} &= -p_4 g\sin(q_1) - p_5 g\sin(q_1 + q_2) \\
B_{12} &= -2p_3\sin(q_2) + p_6 + p_7 \frac{60(30\exp(\dot{q}_1))}{(\exp(30\dot{q}_1) + 1)^2} \\
B_{13} &= p_1 + p_2 + 2p_3\cos(q_2) \\
B_{15} &= -p_3\sin(q_2)(2\ddot{q}_1 + \ddot{q}_2) - p_3\cos(q_2)(\dot{q}_2^2 + 2\dot{q}_1\dot{q}_2) - p_5 g\sin(q_1 + q_2) \\
B_{16} &= -2p_3\sin(q_2)(\dot{q}_1 + \dot{q}_2) \\
B_{17} &= p_2 + p_3\cos(q_2) \\
B_{21} &= -p_5 g\sin(q_1 + q_2) \\
B_{22} &= 2p_3\sin(q_2)\dot{q}_1 \\
B_{23} &= p_2 + p_3\cos(q_2) \\
B_{25} &= -p_3\sin(q_2)\ddot{q}_1 + p_3\cos(q_2)\dot{q}_1^2 - p_5 g\sin(q_1 + q_2) \\
B_{27} &= p_2
\end{aligned}$$

Obviously the covariance matrix σ will now have dimension 7 by 7.
Just as in the case of the fully actuated example, now also first the simulation environment is explored, and afterwards the experimental environment.

Simulation environment

When simulating one can compute the condition number very fast for different control inputs. Several inputs have been tried out, resulting in condition numbers varying between 39.2380 and 260.2951. It should be mentioned that these condition numbers represent the unscaled regressor matrix, not the scaled one. The actual (deterministic) estimation will therefore be somewhat better than one can expect form a condition number of around 40, but in essence it is clear that a condition number of around 40 is much better than one around 260. Obviously one would chose the input corresponding to the lowest condition number possible. However, in this case the variance on the measurements becomes much too high, which is mainly due to the play of the second link. Therefore one has to choose a less exciting control torque such that the variances are acceptable. The trade-off between these two issues corresponds to a condition number of 69.0251, representing an input that is equal to

$$\tau = 1.5\sin(6t) \tag{3.28}$$

Again in the simulation environment the real parameters can be chosen freely, so they are chosen to be of the same order of magnitude as the initial guess of the experimental setup. Taking input (3.29), together with the following real parameter set

$$p = \begin{bmatrix}0.0541 & 0.0553 & 0.0293 & 0.7823 & 0.3360 & 0.1384 & 0.1981\end{bmatrix} \tag{3.29}$$

The Olsen estimator gives the following estimated parameters

$$p = \begin{bmatrix}0.0541 & 0.0543 & 0.0289 & 0.7808 & 0.3303 & 0.1301 & 0.2080\end{bmatrix} \tag{3.30}$$

and thus the relative error (in %) is equal to

$$e = \begin{bmatrix}0.068 & 1.805 & 1.434 & 0.189 & 1.692 & 5.987 & 5.021\end{bmatrix} \tag{3.31}$$

The errors of p_2, p_3 and p_5 are an order higher than the ones for p_1 and p_4. As can be seen from (2.5), the parameters with higher relative errors are contained in the second link's equation of motion, which is not independently actuated. Also the friction parameters are not perfectly estimated. However it is decided that the present estimation is good enough to proceed to the experimental environment. For completeness, the parameters found by the Olsen, weighted least squares, linear least squares and the energy-based method are given in table 3.1, as well as their relative errors. From this table it can be concluded that in the simulation environment the Olsen estimator is the overall best estimator, although the differences are not important. Therefore, in the experimental environment, only the Olsen estimator will be used.

Estimator	Estimated Parameters
Olsen	[0.0541 0.0543 0.0289 0.7808 0.3303 0.1301 0.2080]
Wls	[0.0542 0.0543 0.0289 0.7810 0.3306 0.1301 0.2080]
Sls	[0.0545 0.0544 0.0289 0.7833 0.3307 0.1308 0.2098]
Eb	[0.0541 0.0528 0.0292 0.7825 0.3214 0.1426 0.1948]
	Relative error (in %)
Olsen	[0.068 1.805 1.434 0.189 1.692 5.987 5.021]
Wls	[0.128 1.731 1.403 0.163 1.618 5.964 5.021]
Sls	[0.800 1.716 1.318 0.128 1.579 5.479 5.890]
Eb	[0.033 4.497 0.370 0.026 4.349 3.058 1.165]

Table 3.1: Estimated parameters for simulation data of the pendubot

Experimental environment

Input (3.28) is applied to the real system, and the positions of the first and second angle are measured. This is done 30 times, in 30 independent experiments. Then the positions are differentiated numerically with respect to time and filtered with the use of a second order butterworth filter whose cut-off frequency lies at 10 Hz. Then these filtered velocities are again differentiated numerically with respect to time and the resulting accelerations are also filtered with the same filter. The measured control signal is exactly equal to (3.28), because of the external controller for the motor. Therefore some random noise is added to the 30 measured control signals because otherwise the Olsen routine will not work properly due to a singular covariance matrix. When this data is fitted appropriately into the Olsen routine, the resulting estimated parameters are

$$p_{ols} = [0.0540 \quad 0.0540 \quad 0.0286 \quad 0.7805 \quad 0.3284 \quad 0.1346 \quad 0.2014] \qquad (3.32)$$

In the experimental environment there is no real parameter set available to compare the estimated parameters with. So the best way of checking whether the estimation is good or not is by plotting the simulation results referring to input (3.28) and the estimated parameters, and comparing it with the experimentally measured responses to this control torque. In figure 3.4 the positions of the first and second link are plotted for input (3.28). As a comparison also the responses for the designer's parameters (delivered together with the Mechatronics Kit) are plotted in figure 3.5, they are marked as "Tutorial" parameters.

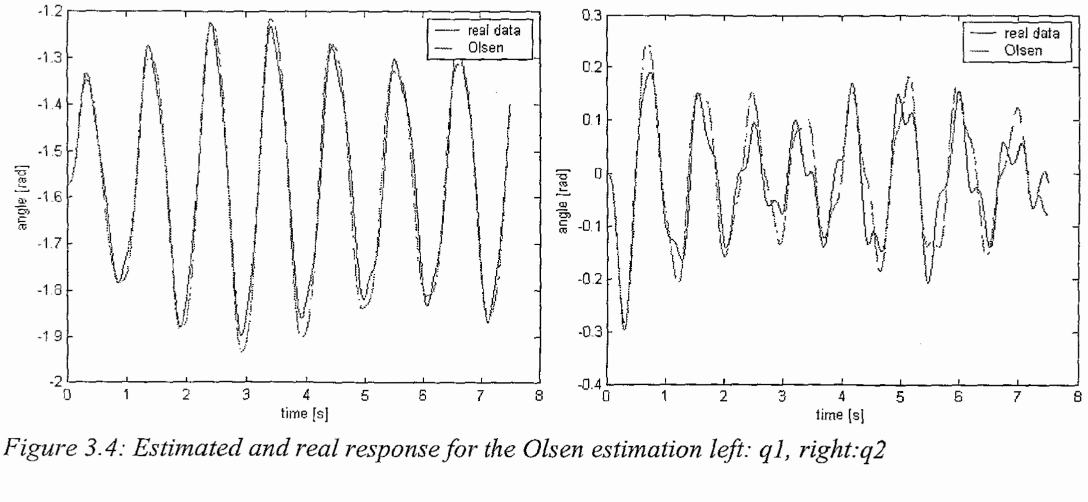


Figure 3.4: Estimated and real response for the Olsen estimation left: q1, right:q2

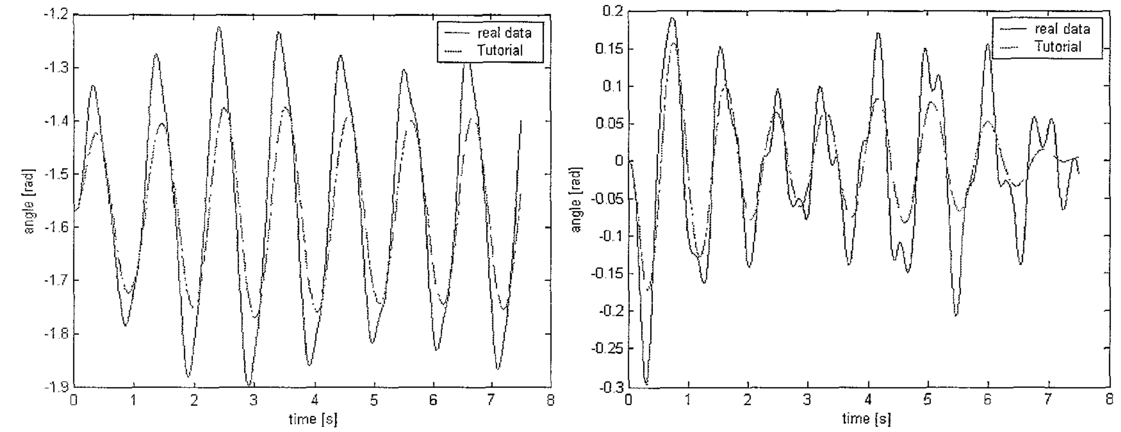


Figure 3.5: Estimated and real response for the designers parameters. Left: q1, right: q2

From figure 3.4 it can be concluded that the motion of the first link is reconstructed more accurately than the motion of the second link by the parameters obtained from Olsen. Also it can be concluded that the parameters from the designers perform worse than the Olsen parameters.

Since figures 3.4 and 3.5 correspond to the estimation data, one could argue that it is unfair to compare the simulations of the Olsen parameters and the tutorial parameters because the Olsen parameters are fitted optimally to this estimation data. However, in figure 3.6 and 3.7 the positions of the first and second link are plotted for the Olsen parameters and the designer's parameters respectively, for the following control torque

$$\tau = 2.5\sin(t) - \cos(4t) \tag{3.33}$$

This is the control torque corresponding to the condition number of 39 (condition number of the unscaled regressor matrix). The reason for not using this torque in the estimation experiments was that the variance on the measurements will then become too large due to the play in the second link. Of course this play is still present for the real data, but not in the simulated data in figures 3.6 and 3.7. Nevertheless the Olsen parameters are still able to capture the dynamics, even when also the friction model is very uncertain. Again, it can be concluded that the motion of the first link is reconstructed better than the motion of the second link, and that the Olsen parameters perform better than the designer's parameters.

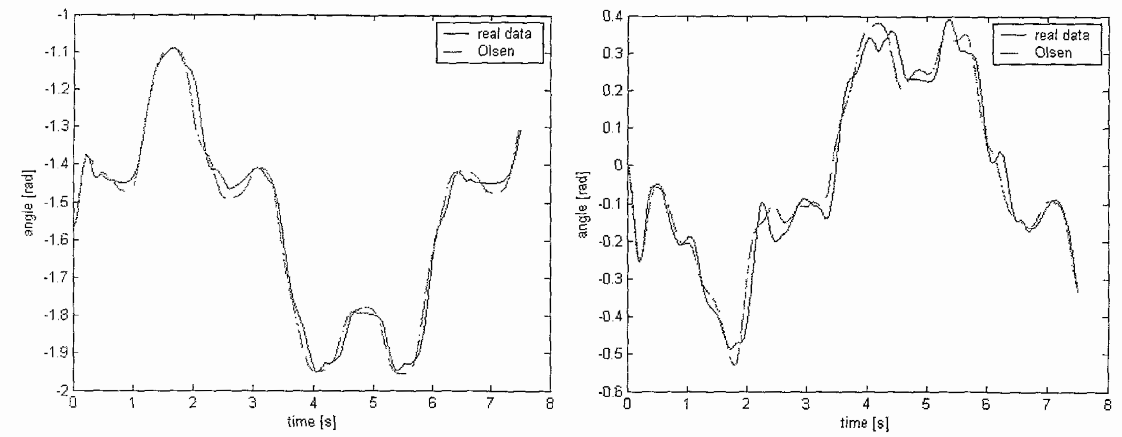


Figure 3.6: Estimated and real response for the Olsen parameters. Left: q1, right: q2

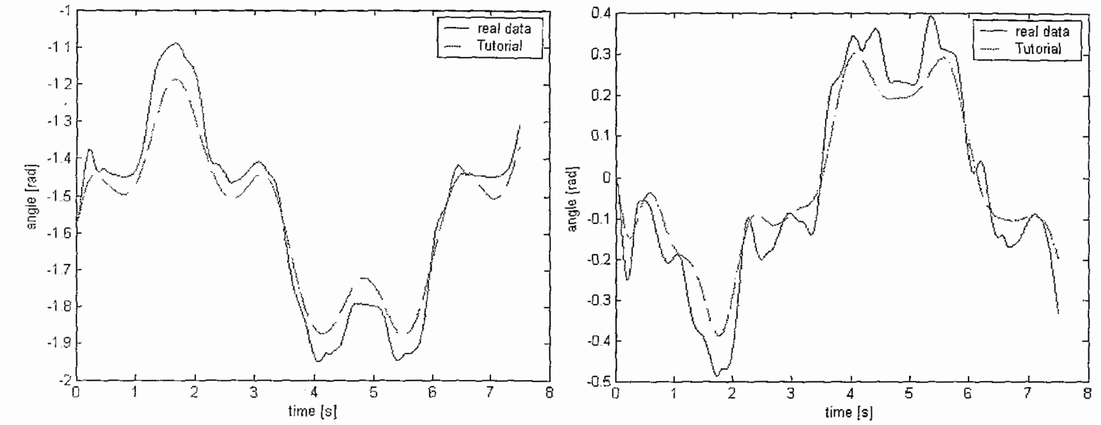


Figure 3.7: Estimated and real response for the designers parameters. Left: q1, right: q2

3.4 *Identification of the reaction wheel*

Considering the rather good results of the Olsen estimation on the *pendubot* it is desirable to estimate the parameters of the reaction wheel configuration in the same way. Again, due to the cables, the friction parameters have to be estimated along with the dynamic parameters. When analysing (2.19) and (2.20), it seems to be a less demanding task to estimate these parameters than the ones of the *pendubot*, because there are only five parameters to estimate instead of seven. However, if experiments are run on the system it becomes clear that the motion of the inertia wheel has a much higher numerical value than the motion of the pendulum. In other words, to make the pendulum move, the inertia wheel has to move vary fast. This high numerical difference causes the matrix b (the squared product of this matrix is needed in the object function) to become singular, because the value of $b(2,1)$ is much higher than the value of $b(1,1)$ as can be seen from (3.34).

$$b = \begin{bmatrix} p_1\ddot{q}_1 + p_3 g\sin(q_1) + \tau - f \\ p_2\ddot{q}_2 - \tau + f \end{bmatrix} \tag{3.34}$$

Therefore neither the Olsen nor the weighted, or standard linear least squares methods can be used. As an alternative, the parameters will be derived using the geometrical values of the links, and weighting the masses. Moreover also the inertia of the whole reaction wheel configuration is measured by means of several independent experiments. During these experiments the pendulum is taken out of the downwards hanging position up to an angle of about 15 degrees, not more, by hand. Then it is released and the response is measured using the encoder. Determining the frequency of oscillation gives a numerical value for

$$\omega = \sqrt{\frac{mgl}{J}} \tag{3.35}$$

Taking 27 independent measurements leads to a frequency of oscillation of

$$\omega = 1.4122Hz \tag{3.36}$$

Because $ml = m_1 l_{c1} + m_2 L_1$, one first has to measure the length of the first link, the mass of the first link (plus motor), the mass of the inertia wheel, and compute l_{c1}. Combining this information with (3.35) and (3.36) gives

$$J = 0.0042 \tag{3.37}$$

Scaling these parameters with the torque constant gives

$$\begin{aligned} p_1 &= 0.8544 \\ p_2 &= 6.864 \end{aligned} \tag{3.38}$$

It should be mentioned that, in these calculations, an error was made not only at determining the centre of mass, but also because the mass of the motor shaft should actually be counted in the second equation of motion instead of the first one. Also the presence of extra friction due to the cables is not accounted for.
The third parameter was already measured in the experiment of section 2.1, and is equal to

$$p_3 = 0.0045 \tag{3.39}$$

Because this parameter was determined by means of reading the encoders and relating it to the torque, this parameter does not have to be scaled anymore.
As a possible subject for future work, one might consider trying the energy-based identification method.

3.5 Identification of the Furuta Pendulum

Using the Olsen method, the b, B and ϕ matrices can be derived. Their contents are reported in appendix 4. Considering the facts that the system has only 6 parameters, and that the *pendubot* has 7 paramenters, and the fact that the motions of both links will be in the same order as in the *pendubot* system, it is expected that the identification of the *furuta pendulum* will give satisfactory results by means of the Olsen method. Due to limitations in time this task remains open for future work.

Chapter 4 Implementation of the controllers

4.1 Pendubot

Taking the parameters of section 3.3, and putting them into the controllers of section 2.2, gives the control laws that are to be implemented in the real experiments. First the controllability of the linearized system is determined. The linear system for the semi-inverted position becomes:

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -153 & 0 & 43.87 & 0 \\ 0 & 0 & 0 & 1 \\ 131.6 & 0 & 38.96 & 0 \end{bmatrix} \qquad \underline{B} = \begin{bmatrix} 0 \\ 25.738 \\ 0 \\ -12.107 \end{bmatrix} \tag{4.1}$$

The linear system for the full inverted position becomes

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 153 & 0 & -43.87 & 0 \\ 0 & 0 & 0 & 1 \\ -174.4 & 0 & 126.71 & 0 \end{bmatrix} \qquad \underline{B} = \begin{bmatrix} 0 \\ 25.738 \\ 0 \\ -39.37 \end{bmatrix} \tag{4.2}$$

Then calculating the rank of the controllability matrix Q

$$\underline{Q} = \begin{bmatrix} B & BA & BA^2 & BA^3 \end{bmatrix} \tag{4.3}$$

gives in both cases

$$rank(\underline{Q}) = 4 \tag{4.4}$$

hence it can be concluded that both linear approximations of the system are controllable, and so the poles can be placed at will. Of course one still has to check whether a pole location is feasible considering the input saturation. This checking has to be done by means of an experiment because we are in fact dealing with a non-linear instead of a linear system. Appropriate full state feedback gains are derivable using the commands *Acker* or *lqrd* from *Matlab*. Take for instance the pole locations of the semi inverted position to be

$$p = \{-10,-10,-10,-10\} \tag{4.5}$$

There is no specific reason for choosing this pole location above another except that we want the poles all to be located in the left half of the complex plane. Then the gain matrix K will be equal to

$$K = \begin{bmatrix} 48.23 & 5.25 & 62.4 & 7.86 \end{bmatrix} \tag{4.6}$$

The same method can be applied for the full inverted position, giving

$$K = [-61.3 \quad -10.47 \quad -62.4 \quad -7.86] \tag{4.7}$$

Both gain matrices give stabilising controllers in the real system.

Now the linear controllers are ready to be implemented. The non-linear controller still needs the parameters for the trajectory, and the parameters for the outer loop gains. As long as the outer loop gains are positive, asymptotic stability of the first links motion is achieved. The following gains give fast enough response to the semi-inverted position.

$$\begin{aligned} K_p &= 150 \\ K_d &= 21 \end{aligned} \tag{4.8}$$

And the following for the full inverted position

$$\begin{aligned} K_p &= 350 \\ K_d &= 24.3 \end{aligned} \tag{4.9}$$

The parameters for the trajectory remain to be found. Because there is no systematic way to derive the values, they have to be found by means of trial and error. The values are found to be:

$$\begin{aligned} \omega &= 5.5 \\ a &= 1.65 \end{aligned} \tag{4.10}$$

for the semi inverted position, and

$$backopenloop = -3 \tag{4.11}$$

for the full inverted position.
The designers suggested a gain matrix for the semi-inverted position equal to

$$K_{mid} = [122.61 \quad 13.11 \quad 152.29 \quad 19.88] \tag{4.12}$$

Resulting in a pole location of

$$p = \{-80.6, -4.5 + 6.2i, -4.5 - 6.2i, -7.06\} \tag{4.13}$$

Obviously the designers chose to add some damping to the system. However, as can be expected, the poles are all situated in the left complex halfplane and so the system is stable.

Combining these controllers and reference paths in an experiment for the semi-inverted position gives the response plotted in figure 4.1.

As can be seen from this figure, the controller is stable, but the poles can be moved further to the left on the complex plane, because the input is far from saturating. However, the goal of the traineeship is not to design the best possible controller.

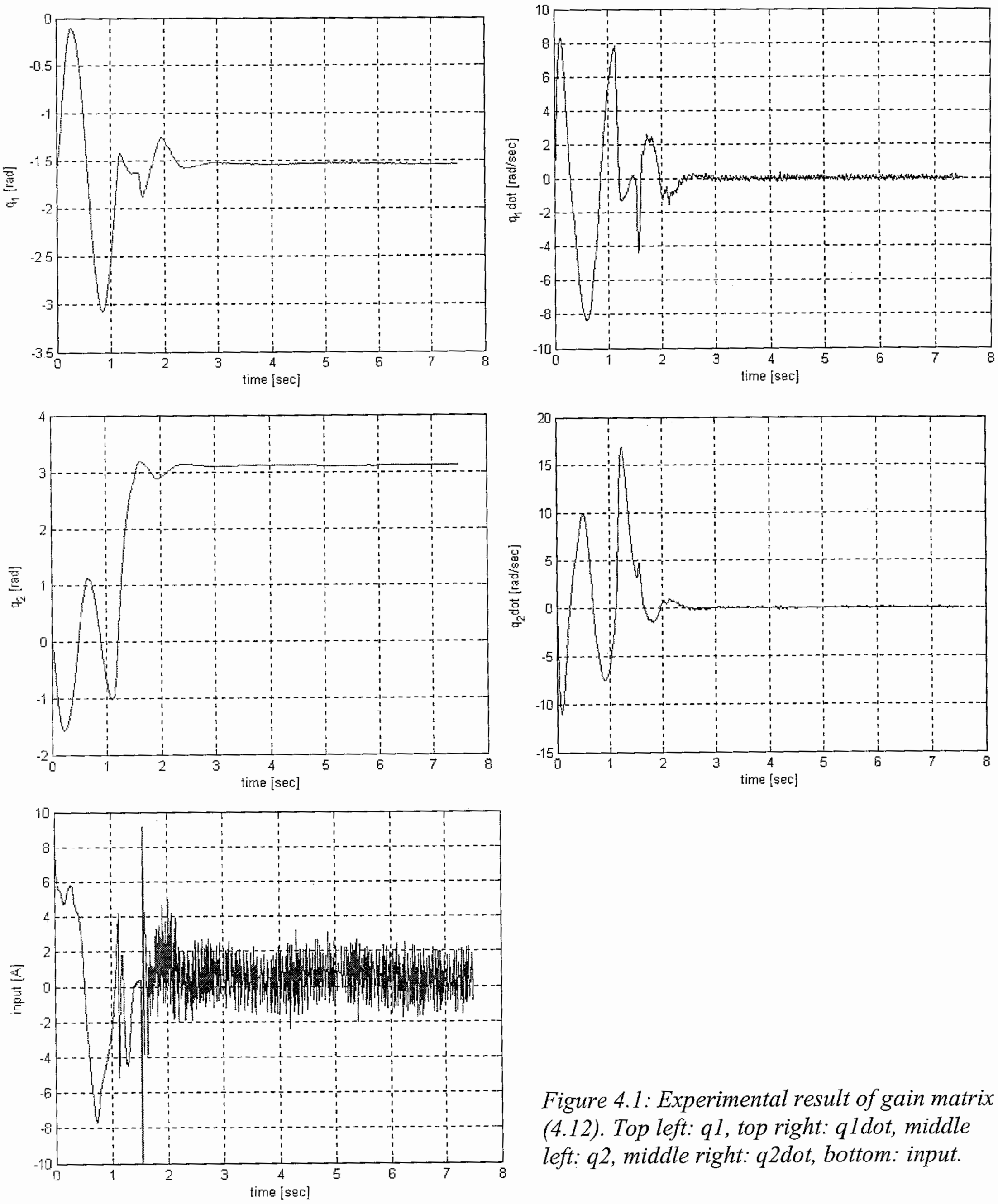


Figure 4.1: Experimental result of gain matrix (4.12). Top left: q1, top right: q1dot, middle left: q2, middle right: q2dot, bottom: input.

For the full inverted position the designer chose the gain matrix to be equal to

$$K_{top} = \begin{bmatrix} -160.78 & -26.92 & -158.72 & -20.81 \end{bmatrix} \tag{4.14}$$

This choice of gain matrix leads to pole locations of

$$p = \{-110.5, -2.95, -6.4 + 2.1i, -6.4 - 2.1i\} \tag{4.15}$$

The resulting switching controller gives the responses plotted in figure 4.2. As can be seen, the static value of the second angle is not zero but has an offset from zero. This is because in the Visual Basic application the offset was chosen to be non-zero. The reason for the freedom to chose an offset parameter is because of the uncertainty of the friction. Apparently the estimation of the friction and dynamical parameters is good enough to skip the offset, or at least decrease it a little. Again the poles can be located a little further to the left in the complex plane because the control torque is far from saturating.

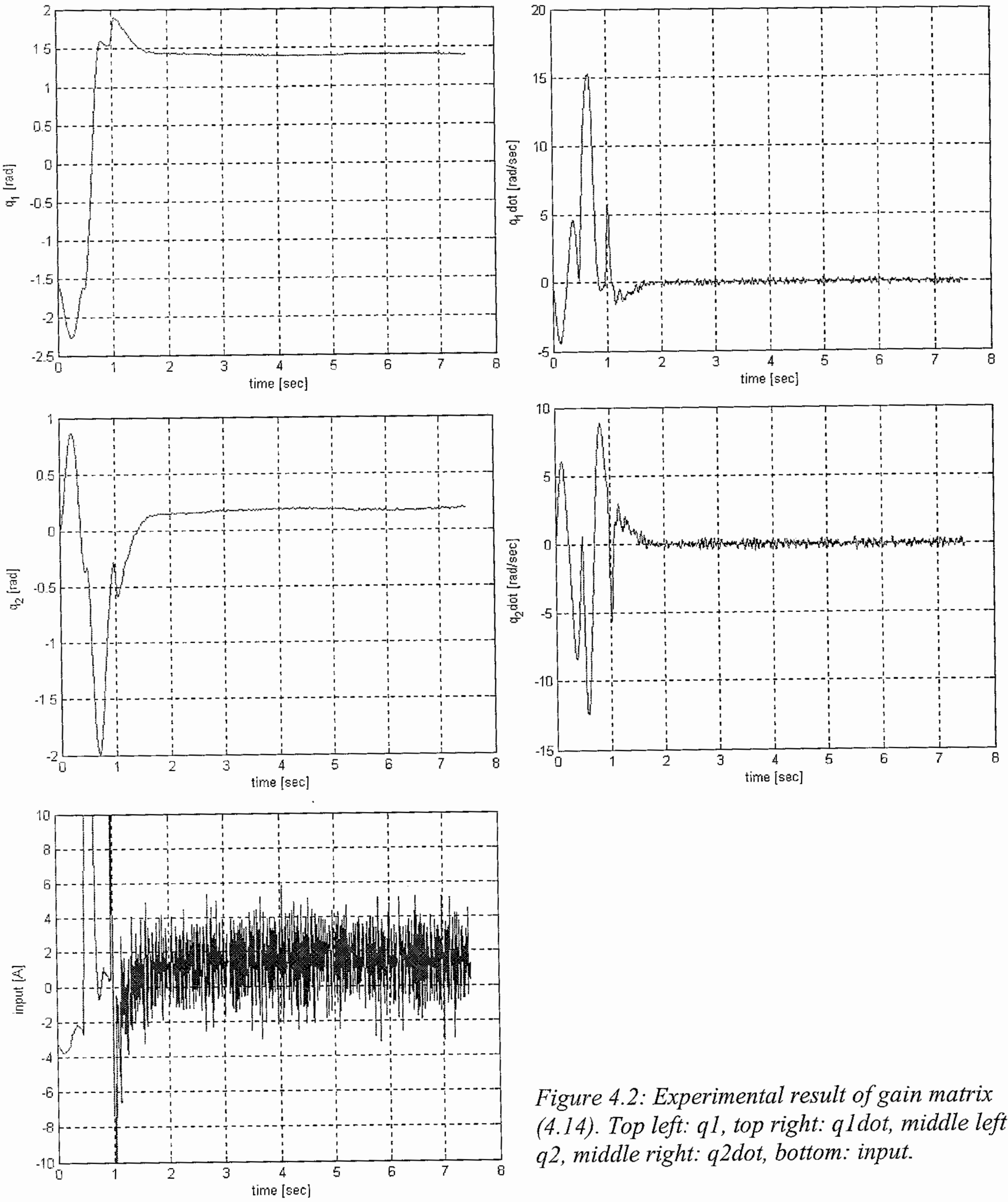


Figure 4.2: Experimental result of gain matrix (4.14). Top left: q1, top right: q1dot, middle left: q2, middle right: q2dot, bottom: input.

In figure 4.2, the bang-bang strategy can be seen. For a period of about half a second after the open-loop control torque, the input is saturated.

4.2 The reaction wheel

Considering the qualitative analysis of chapter 2 on the passive energy-based controller, some parameters are free to be chosen. As long as the sign of the parameters is positive, the controller will behave as required, thus approaching the homoclinic orbit. Because the parameters of the non-linear and linear controller are related, one is not completely free to choose the parameters, because although the non-linear controller is passivity based, and in the case of the *reaction wheel* not sensitive to saturation, the linear controller is not passive, and consequently sensitive to saturation. The designer chose the following parameters

$$\begin{aligned} k_e &= 4000 \\ k_v &= 4 \\ k_u &= 0.4 \\ K &= \begin{bmatrix} -400 & -35 & -0.07 \end{bmatrix} \end{aligned} \tag{4.16}$$

With this gain matrix the following pole locations are achieved

$$p = \{-9.97 + 13.33i, -9.97 - 13.33i, -3.9214\} \tag{4.17}$$

Looking at figure 4.3, the response to this controller looks satisfactory. If one analyses the control law, one will discover that the conditions from (2.62) and (2.63), are not met. The third element in the gain matrix does not equal the numerical value

$$K_3 = \sqrt{\frac{p_3 g}{p_1}} \tag{4.18}$$

Moreover, the ratio between the first two elements of the gain matrix is not equal to

$$K_1 = \omega_p K_2 \tag{4.19}$$

These two conditions are thus not achieved. The poles with the use of the linear controller are in the left half plane, and the non-linear controller is able to approach to the homoclinic orbit, as can be seen from figure 4.3. The switching, however, is not perfectly free from transient behaviour, while actually this was one of the goals to achieve with the switching action. This illustrates one should always be careful in taking the results of another person for granted.

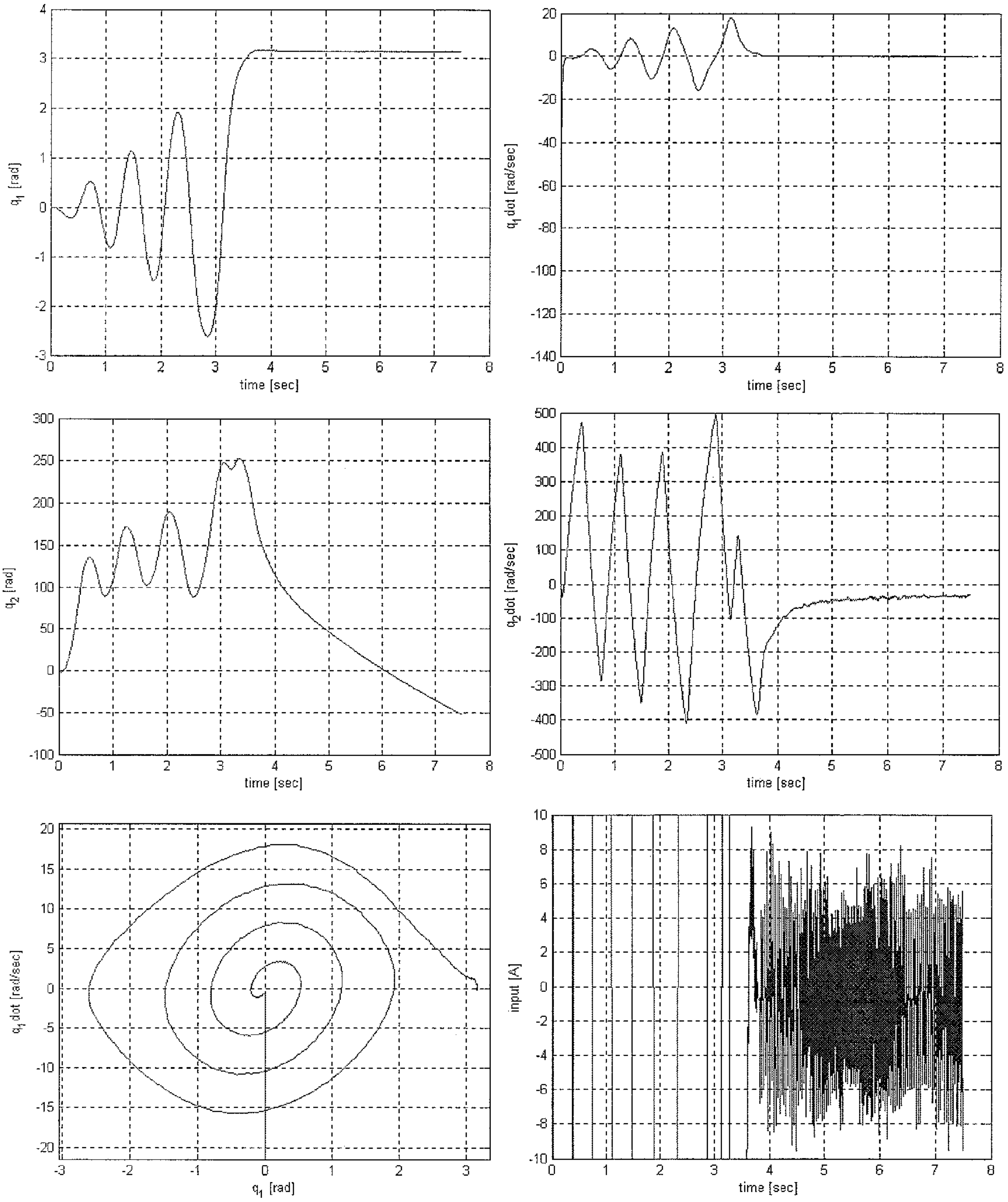


Figure 4.3: Experimental response of reaction wheel with switching controller. Top left: q1, top right: q1dot, middle left: q2, middle right: q2dot, bottom left: phase-plot (q1,q1dot), bottom right: input.

From figure 4.3 it can be seen that the switching, although not very abrupt (because the ratio between controller parameters is in the vicinity of the desired ratio), is not totally free of transient response. Moreover it can be seen from the phase-plot that the initial condition for the velocity is not equal to zero. This is done on purpose because starting at the origin will result in no motion at all. The reason for this is obvious: the origin belongs to the largest invariant set H, derived in section 2.3.

Intuitively one could think that it is a good idea to increase the linear controller gains to put the locations of the poles further in the left complex half plane, because the control input is not saturated yet. One should be careful by doing so. Considering condition (2.62), the third pole location is fixed being a function of the dynamical parameters. Therefore the only freedom is in the level of the first two controller gains.

Note however that the ratio between these two gains is also fixed, which does not leave much freedom for design.

Chapter 5 Conclusions and Recommendations

Applying the Olsen identification method, which is actually suited for fully actuated systems, shows satisfactory results for the *pendubot* configuration, for as long as attention is paid to practical limitations of the system. The Olsen method is found to be inappropriate for the *reaction wheel* configuration. Considering the similarities between the *pendubot* and the *furuta pendulum*, it is expected that the Olsen method will work for the *furuta pendulum* as well. Tuning the controller parameters of the *pendubot* shows satisfactory stability on both the semi-inverted and the fully-inverted position, although the controller parameters are not designed to be optimal. After a careful analysis on the switching controller of the reaction wheel it is discovered that the possible and very elegant goal of the non-transient switching strategy can be achieved, but this is not achieved in the demo versions of the controllers because of the violation of several constraints.

After working with the Mechatronics Kit for three months, it can be concluded that

- The electronic part is advanced, except for the encoders (resolution and attachment) and the maximum torque of the motor.
- The mechanical part is not of very high standard, so really high positioning accuracy will most probably be not possible. This is however not the goal of the Mechatronics Kit. The usage will be for educational and research purposes, and for these goals the Mechatronics Kit is perfectly suited.
- Everything is set up to implement a controller very quickly, even if a person is not familiar with the working environment. The operators (students) should therefore first read appendix 6.

Besides the enormous possible implementations and studies that can be done on the Mechatronics Kit, it is recommended that the following points be considered:

- Use of different cables to transport the pulses of the encoders. This can have a very positive effect on the performance of the controllers since in the present situation the friction and stiffness cannot be neglected. Indeed, their influence is rather significant.
- Implementation of the Olsen method for the simple pendulum configuration using the optimal trajectory. Unfortunately there was no time left to perform this task during the traineeship.
- Implementation of a completely different non-linear controller for the *pendubot* configuration. Although it satisfactorily approaches the equilibrium point, there is no feedback of the second link's motion. This strategy is considered to be a function of "faith and trial and error", which is not the most elegant way in designing controllers.
- Identification of the *reaction wheel* configuration by means of the energy-based method.
- Identification of the *furuta pendulum* configuration by means of the Olsen method.
- Implementation of the right linear controller gains for the reaction wheel configuration, so that there is no transient behaviour during the switching.

References

[1] Karl J. Åström, Daniel J. Block, Mark W. Spong, "The reaction wheel pendulum", *2001, University of Illinois, Urbana.*

[2] Daniel J. Block, "Mechanical design and control of the pendubot", *Master Thesis 1996, university of Illinois, Urbana.*

[3] Masafumi Iwashiro, Katsuhisa Furuta, Karl, J. Åström, "Energy based control of pendulum", *IEEE International Conference on Control Applications, pp. 715-720, Sept. 1996.*

[4] M.M. Olsen, H.G. Petersen, "A new method for estimating parameters of a dynamic robot model", *IEEE Transactions on robotics and automation, Vol. 17, No. 1, pp. 95-100, Feb. 2001*

[5] Jan Swevers, Chris Ganseman, Dilek Bilgin Tükel, Joris de Schutter, Hendrik van Brussel, "Optimal robot excitation and identification", *IEEE Transactions on robotics and automation, Vol. 13, No. 5, pp. 730-740, Oct. 1997.*

[6] C. de Wit, H. Olsson, K.J. Åström, P. Lischinsky, "A new model for control of systems with friction", *IEEE Transactions on automatic control, Vol. 40, No. 3, pp. 419-425, March 1995.*

[7] Ron H.A. Hensen, René J.G. van de Molengraft, Maarten Steinbuch, "Frequency domain identification of dynamic friction model parameters", *IEEE Transactions on control systems technology, Vol. 10, No. 2, pp. 191-196, March 2002.*

[8] Shankar Sastry, *Nonlinear systems, analysis, stability and control, Springer-Verlag, Berlin, 1999.*

[9] Gene F. Franklin, J. David Powell, Abbas Emami-Naeini, *Feedback control of dynamical systems, 3th edition, Adison Wesley, 1994.*

Appendix 1

Derivation of the equations of motion

A1.1 The pendubot

From figure 2.4 it is evident that the coordinates of the two centres of masses with respect to the base space of the two pendulum links are equal to

$$\underline{r_1} = \begin{bmatrix} l_{c1}\cos(\theta_1) \\ l_{c1}\sin(\theta_1) \end{bmatrix} \text{ and } \underline{r_2} = \begin{bmatrix} l_1\cos(\theta_1) + l_{c2}\cos(\theta_1+\theta_2) \\ l_1\sin(\theta_1) + l_{c2}\sin(\theta_1+\theta_2) \end{bmatrix}$$

so that

$$\underline{\dot{r}_1} = \begin{bmatrix} -l_{c1}\dot{\theta}_1\sin(\theta_1) \\ l_{c1}\dot{\theta}_1\cos(\theta_1) \end{bmatrix} \text{ and } \underline{\dot{r}}_2 = \begin{bmatrix} -l_1\dot{\theta}_1\sin(\theta_1) - l_{c2}\sin(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2) \\ l_1\dot{\theta}_1\cos(\theta_1) + l_{c2}\cos(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2) \end{bmatrix}$$

The kinetic energy is equal to the following:

$$T = \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}m_1\underline{\dot{r}}_1^T\underline{\dot{r}}_1 + \frac{1}{2}J_2(\dot{\theta}_1+\dot{\theta}_2)^2 + \frac{1}{2}m_2\underline{\dot{r}}_2^T\underline{\dot{r}}_2$$

Which after some goniometry leads to

$$T = \frac{1}{2}(m_1l_{c1}^2 + m_2l_1^2 + J_1)\dot{\theta}_1 + \frac{1}{2}(m_2l_{c2}^2 + J_2)(\dot{\theta}_1^2 + 2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \\ + m_2l_1l_{c2}(\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2)\cos(\theta_2)$$

The potential energy is equal to

$$V = (m_1l_{c1} + m_2l_1)g\sin(\theta_1) + m_2l_{c2}g\sin(\theta_1+\theta_2)$$

Herein the potential energy is taken as zero in the base frame, see figure 2.4.
If one now chooses the following parameters

$$p_1 = m_1l_{c1}{}^2 + m_2l_1^2 + J_1$$
$$p_2 = m_2l_{c2}^2 + J_2$$
$$p_3 = m_2l_1l_{c2}$$
$$p_4 = m_1l_{c1} + m_2l_1$$
$$p_5 = m_2l_{c2}$$

Then this finally gives

$$L = T - V = \frac{1}{2} p_1 \dot{\theta}_1 + \frac{1}{2} p_2 \left(\dot{\theta}_1^2 + 2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2 \right) + p_3 \cos(\theta_2) \left(\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2 \right) - p_4 g \sin(\theta_1)$$
$$- p_5 g \sin(\theta_1 + \theta_2)$$

Now apply Lagrange's method

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \underline{\tau}$$

In which

$$\underline{\tau} = \begin{bmatrix} \tau_1 - f_1 \\ 0 \end{bmatrix}$$

and

$$f_1 = p_6 \dot{\theta}_1 + p_7 \left(1 - \frac{2}{\exp(2k\dot{\theta}_1) + 1} \right), \qquad k = 15$$

Because

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}_1} \right) = (p_1 + p_2 + 2p_3 \cos(\theta_2))\ddot{\theta}_1 + (p_2 + p_3 \cos(\theta_2))\ddot{\theta}_2 - p_3 \sin(\theta_2)\dot{\theta}_1^{\,2}$$
$$- 2p_3 \sin(\theta_2)\dot{\theta}_1 \dot{\theta}_2$$

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}_2} \right) = (p_2 + p_3 \cos(\theta_2))\ddot{\theta}_1 - p_3 \sin(\theta_2)\dot{\theta}_1 \dot{\theta}_2 + p_2 \ddot{\theta}_2$$

$$\frac{\partial L}{\partial q_1} = -p_4 g \cos(\theta_1) - p_5 g \cos(q_1 + q_2)$$

$$\frac{\partial L}{\partial q_2} = -p_3 \sin(\theta_2)\left(\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2 \right) - p_5 g \cos(q_1 + q_2)$$

This finally gives

$$\underline{D}\ddot{\underline{q}} + \underline{C}\dot{\underline{q}} + \underline{\phi} = \underline{\tau}$$

With

$$\underline{D} = \begin{bmatrix} p_1 + p_2 + 2p_3 \cos(q_2) & p_2 + p_3 \cos(q_2) \\ p_2 + p_3 \cos(q_2) & p_2 \end{bmatrix}$$

$$\underline{C} = \begin{bmatrix} -p_3 \cdot \sin(q_2) \cdot \dot{q}_2 & -p_3 \cdot \sin(q_2) \cdot (\dot{q}_1 + \dot{q}_2) \\ p_3 \cdot \sin(q_2) \cdot \dot{q}_1 & 0 \end{bmatrix}$$

$$\underline{\phi} = \begin{bmatrix} p_4 \cdot g \cdot \cos(q_1) + p_5 \cdot g \cdot \cos(q_1 + q_2) \\ p_5 \cdot g \cdot \cos(q_1 + q_2) \end{bmatrix}$$

$$f = p_6 \cdot \dot{q}_1 + p_7 \cdot \left(1 - \frac{2}{\exp(2 \cdot k \cdot \dot{q}_1) + 1} \right)$$

A1.2 The Reaction Wheel

From figure 2.6 it can be concluded that

$$\underline{r}_{c1} = \begin{bmatrix} l_{c1} \sin(\theta) \\ l_{c1} \cos(\theta) \end{bmatrix} \qquad \text{and} \qquad \underline{r}_{c2} = \begin{bmatrix} L_1 \sin(\theta) \\ L_1 \cos(\theta) \end{bmatrix}$$

so that

$$\dot{\underline{r}}_{c1}^T \dot{\underline{r}}_{c1} = l_{c1}^2 \dot{\theta}^2 \qquad \text{and} \qquad \dot{\underline{r}}_{c2}^T \dot{\underline{r}}_{c2} = L_1^2 \dot{\theta}^2$$

The kinetic energy is equal to

$$T = \frac{1}{2} m_1 \dot{\underline{r}}_{c1}^T \dot{\underline{r}}_{c1} + \frac{1}{2} J_1 \dot{\theta}^2 + \frac{1}{2} m_2 \dot{\underline{r}}_{c2}^T \dot{\underline{r}}_{c2} + \frac{1}{2} J_2 \dot{\theta}_r^2$$

The Potential energy with respect to the base frame is equal to

$$V = -(m_1 l_{c1} + m_2 L_1) g \cos(\theta)$$

So, consequently

$$L = \frac{1}{2}(m_1 l_{c1}^2 + m_2 L_1^2 + J_1)\dot{\theta}^2 + \frac{1}{2} J_2 \dot{\theta}_r^2 + (m_1 l_{c1} + m_2 L_1) g \cos(\theta)$$

Obviously the minimum parameter set is equal to

$p_1 = m_1 l_{c1}^2 + m_2 L_1^2 + J_1$

$p_2 = J_2$

$p_3 = m_1 l_{c1} + m_2 L_1$

Filling in these parameters into L, one obtains

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = p_1\ddot{\theta}$$

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\theta}_r}\right) = p_2\ddot{\theta}_r$$

$$\frac{\partial L}{\partial \theta} = -p_3 g \sin(\theta)$$

$$\frac{\partial L}{\partial \theta_r} = 0$$

So the final equations of motion become

$$p_1\ddot{\theta} + p_3 g \sin(\theta) = -(\tau - f)$$
$$p_2\ddot{\theta}_r = \tau - f$$

With f equal to

$$f = p_4 \cdot \dot{q}_1 + p_5 \cdot \left(1 - \frac{2}{\exp(2 \cdot k \cdot \dot{\theta}_r) + 1}\right)$$

The parameters of the friction model have the same numerical value as in the other configurations.

A1.3 The Furuta Pendulum

From figure 2.12 it can be seen that

$$\underline{r}_{c1} = \begin{bmatrix} l_{c1}\cos(a) \\ l_{c1}\sin(a) \\ 0 \end{bmatrix}$$

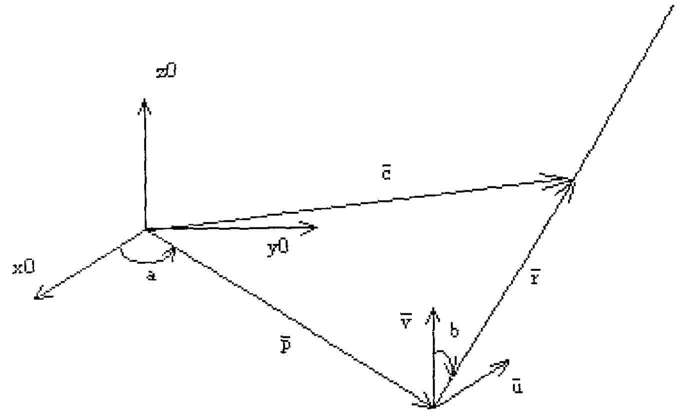


Figure A1: Vector drawing with respect to the Fyryta pendulum

The expression for $\underline{r}_{c2}$ is less trivial but can be retrieved from a graphical interpretation and some vector rules.
From figure A1 it is clear that

$$\bar{c} = \bar{p} + \bar{r}$$

In this figure $\bar{p}$ represents the position vector of the coupling joint between the arm and the pendulum, expressed in the base frame $\{x0, y0, z0\}$. Because the plane in which the pendulum rotates will always be orthogonal to the vector $\bar{p}$, one can span this plane with two vectors $\{\bar{u}, \bar{v}\}$, and express the vector $\bar{r}$, which represents the position of the centre of mass of the pendulum as a function of these two vectors.
The expression is equal to

$$\bar{r} = l_{c2}\sin(b)\bar{u} + l_{c2}\cos(b)\bar{v}$$

The vector $\bar{p}$ can be expressed as

$$\bar{p} = L_1\cos(a)\overline{x0} + L_1\sin(a)\overline{y0}$$

In order to be able to write the expression for $\bar{r}$ in terms of the base frame, one has to find a relation between $\{\bar{u}, \bar{v}\}$ and $\{\overline{x0}, \overline{y0}, \overline{z0}\}$. This relation can be found through the orthogonality constraints. For two vectors that are perpendicular to one another their improduct has to be equal to zero. From the figure we can see that $\bar{u}$, $\bar{v}$ and $\bar{p}$ always are perpendicular to each other, therefore the constraints are:

$$\langle \bar{u}, \bar{v} \rangle = 0$$
$$\langle \bar{u}, \bar{p} \rangle = 0$$
$$\langle \bar{v}, \bar{p} \rangle = 0$$

The vectors

$$\bar{u} = -\sin(a)\overline{x0} + \cos(a)\underline{y0} \qquad \text{and} \qquad \bar{v} = \underline{z0}$$

fulfil these conditions.
Filling them in and adding $\bar{p}$ and $\bar{r}$ gives

$$\bar{c} = \begin{bmatrix} L_1\cos(a) - l_{c2}\sin(a)\sin(b) \\ L_1\sin(a) + l_{c2}\cos(a)\sin(b) \\ l_{c2}\cos(b) \end{bmatrix} \begin{bmatrix} \overline{x0} & \overline{y0} & \overline{z0} \end{bmatrix}$$

This relation could also have been found by means of the more general euler angles method thus using rotation matrices which combines with translations gives transformation matrices.
Obviously, the time derivatives equal

$$\underline{\dot{r}}_{c1} = \begin{bmatrix} -l_{c1}\dot{a}\sin(a) \\ l_{c1}\dot{a}\cos(a) \\ 0 \end{bmatrix} \quad \text{and} \quad \underline{\dot{r}}_{c2} = \begin{bmatrix} -L_1\dot{a}\sin(a) - l_{c2}\left(\dot{a}\cos(a)\sin(a) + \dot{b}\sin(a)\cos(b)\right) \\ L_1\dot{a}\cos(a) + l_{c2}\left(\dot{b}\cos(a)\cos(b) - \dot{a}\sin(a)\sin(b)\right) \\ -\dot{b}l_{c2}\sin(b) \end{bmatrix}$$

Therefore

$$\underline{\dot{r}}_{c1}^T\underline{\dot{r}}_{c1} = l_{c1}^2\dot{a}^2$$

hence the kinetic energy of the first link equals

$$T_1 = \frac{1}{2}J_1\dot{a}^2 + \frac{1}{2}m_1 l_{c1}^2\dot{a}^2$$

The kinetic energy for the second link is derived in a somehow different manner. From the basic principles of energy, the kinetic energy of the second link has to be equal to

$$T_2 = \frac{1}{2}\int \rho\left|\dot{r}(x)\right|^2 dV$$

Here $r(x)$ stands for the position along the link, written in the base frame. Obviously $r(x)$ equals the position of the centre of mass, only now evaluated at x, accept now l_{c2} should be replaced by the dummy variable x which moves along the link. The density can be put before the integral, giving

$$T_2 = \frac{1}{2}\frac{m_2}{L_2}\int_0^{L_2}\left|\dot{r}(x)\right|^2 dx$$

Computing this integral completes the kinetic energy of the system. The potential energy is only contained in the second link, and is obviously equal to

$$V = m_2 g l_{c2} \cos(b)$$

Now the total energy is given by

$$L = T_1 + T_2 - V$$

Choosing the minimal parameter set to be

$$\begin{aligned} p_1 &= \frac{1}{2} J_1 + \frac{1}{2} m_1 l_{c1} + \frac{1}{2} m_2 L_1^2 \\ p_2 &= m_2 L_1 L_2 \\ p_3 &= J_2 \\ p_4 &= m_2 l_{c2} \end{aligned}$$

this becomes

$$L = (p_1 - p_2 \cos(2b) + p_3)\dot{a}^2 + 2 p_2 \dot{b}^2 + \frac{1}{2} p_3 \cos(b) \dot{a}\dot{b} - p_4 g \cos(b)$$

Because again $\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q} = \tau$, this gives

$$\underline{D}\begin{bmatrix} \ddot{a} \\ \ddot{b} \end{bmatrix} + \underline{C}\begin{bmatrix} \dot{a} \\ \dot{b} \end{bmatrix} + \underline{G} = \underline{\tau}$$

with

$$\underline{D} = \begin{bmatrix} p_1 + 4 p_3 \sin^2(b) & \frac{1}{2} p_2 \cos(b) \\ \frac{1}{2} p_2 \cos(b) & 4 p_3 \end{bmatrix}$$

$$\underline{C} = \begin{bmatrix} 4 p_3 \sin(2b)\dot{b} & - p_2 \sin(b)\dot{b} + 4 p_3 \sin(2b)\dot{a} \\ - 4 p_3 \sin(2b)\dot{a} & 0 \end{bmatrix}$$

$$\underline{G} = \begin{bmatrix} 0 \\ - p_4 g \sin(b) \end{bmatrix}$$

$$\underline{\tau} = \begin{bmatrix} u - f \\ 0 \end{bmatrix} \qquad f = p_5 \cdot \dot{a} + p_6 \cdot \left(1 - \frac{2}{\exp(2 \cdot k \cdot \dot{a}) + 1}\right)$$

Appendix 2

Parameters for the linear representation in the equilibrium points.

$$a_{21} = \frac{g(p_3 p_5 - p_2 p_4)}{p_1 p_2 - p_3^2}$$

$$a_{23} = \frac{g p_3 p_5}{p_1 p_2 - p_3^2}$$

$$a_{41} = \frac{g(-p_3 p_4 - p_3 p_5 + p_2 p_4 + p_1 p_5)}{p_1 p_2 - p_3^2}$$

$$a_{43} = \frac{g p_5 (p_1 - p_3)}{p_1 p_2 - p_3^2}$$

$$b_2 = \frac{p_2}{p_1 p_2 - p_3^2}$$

$$b_4 = \frac{-p_2 + p_3}{p_1 p_2 - p_3^2}$$

For the fully inverted *pendubot* one obtains

$$a_{21} = \frac{g(p_2 p_4 - p_3 p_5)}{p_1 p_2 - p_3^2}$$

$$a_{23} = -\frac{g p_3 p_5}{\mathrm{p}_1 p_2 - p_3^2}$$

$$a_{41} = \frac{g(-p_2 p_4 - p_3 p_4 + p_3 p_5 + p_1 p_5)}{\mathrm{p}_1 p_2 - p_3^2}$$

$$a_{43} = \frac{g p_5 (p_1 + p_3)}{\mathrm{p}_1 p_2 - p_3^2}$$

$$b_2 = \frac{p_2}{p_1 p_2 - p_3^2}$$

$$b_4 = -\frac{p_2 + p_3}{p_1 p_2 - p_3^2}$$

Appendix 3

Picture of the pendubot fully-inverted position

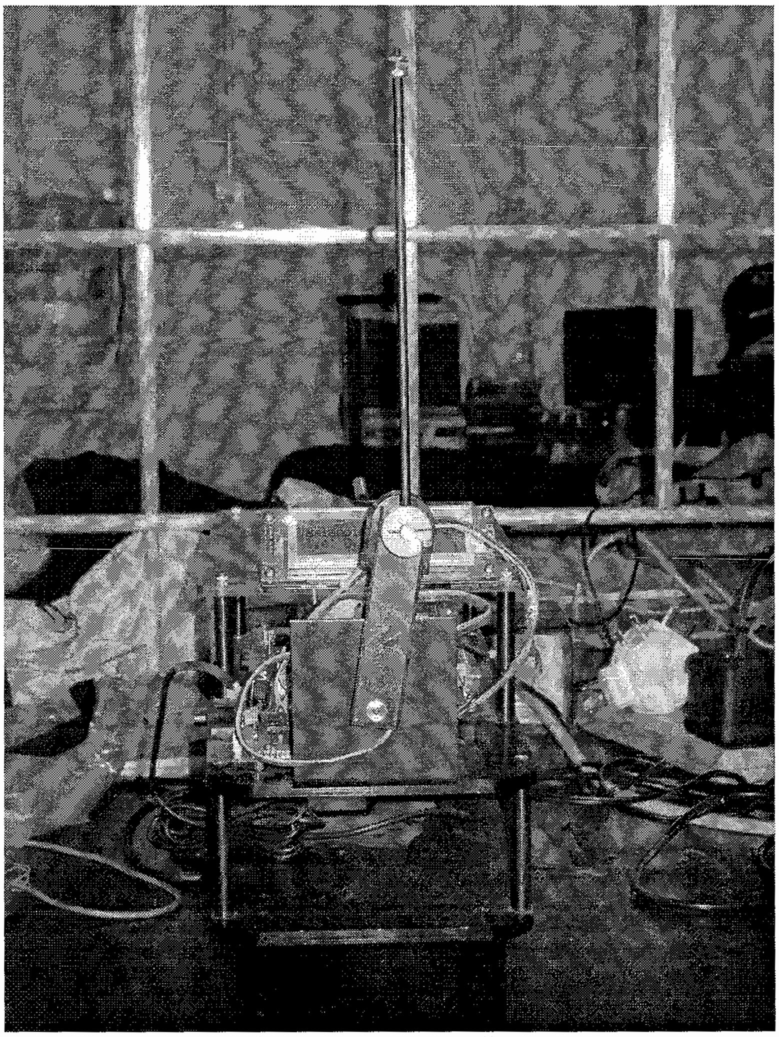


Figure A3: Picture of the pendubot at the fully-inverted position

Appendix 4

Derivation of the identification matrices of the furuta pendulum

Following the Olsen method, the following matrices can be derived.

$$\underset{\sim}{b} = \begin{bmatrix} -G_1 \\ -G_2 \end{bmatrix}$$

with

$$G_1 = \left(p_1 + 4p_3 \sin^2(b)\right)\ddot{a} + \frac{1}{2} p_2 \cos(b)\ddot{b} + 8p_3 \sin(2b)\dot{a}\dot{b} - p_2 \sin(b)\dot{b}^2 - \tau_1$$

$$+ p_5\dot{a} + p_6\left(1 - \frac{2}{\exp(30\dot{a}+1)}\right)$$

$$G_2 = \frac{1}{2}\cos(b)p_2\ddot{a} + 4p_3\ddot{b} - 4p_3 \sin(2b)\dot{a}^2 - p_4 g \sin(b)$$

Consequently the B matrix will be equal to

$$B = \begin{bmatrix} 0 & B_{12} & B_{13} & -1 & B_{15} & B_{16} & B_{17} \\ 0 & B_{22} & B_{23} & 0 & B_{25} & 0 & B_{27} \end{bmatrix}$$

with

$$B_{12} = 8p_3 \sin(2b)\dot{b} + p_5 + p_6 \frac{\partial Fc}{\partial \dot{a}}$$

$$B_{13} = p_1 + 4p_3 \sin^2(b)$$

$$B_{15} = 8p_3 \sin(b)\cos(b)\ddot{a} - \frac{1}{2} p_2 \sin(b)\ddot{b} + 16p_3 \cos(2b)\dot{a}\dot{b} - p_2 \cos(b)\dot{b}^2$$

$$B_{16} = 8p_3 \sin(2b)\dot{a} - 2p_2 \sin(b)\dot{b}$$

$$B_{17} = \frac{1}{2} p_2 \cos(b)$$

$$B_{22} = -8p_3 \sin(2b)\dot{a}$$

$$B_{23} = \frac{1}{2} p_2 \cos(b)$$

$$B_{25} = -\frac{1}{2} p_2 \sin(b)\ddot{a} - 8p_3 \cos(2b)\dot{a}^2 - p_4 g \cos(b)$$

$$B_{27} = 4p_3$$

From this it follows that the regressor matrix is equal to

$$\underline{\phi} = \begin{bmatrix} F_{11} & F_{12} & F_{13} & 0 & F_{15} & F_{16} \\ 0 & F_{22} & F_{23} & F_{24} & 0 & 0 \end{bmatrix}$$

with

$$F_{11} = \ddot{a}$$
$$F_{12} = \frac{1}{2}\cos(b)\ddot{b} - \sin(b)\dot{b}^2$$
$$F_{13} = 4\sin^2(b)\ddot{a} + 8\sin(2b)\dot{a}\dot{b}$$
$$F_{15} = \dot{a}$$
$$F_{16} = 1 - \frac{2}{\exp(30\dot{a}) + 1}$$
$$F_{22} = \frac{1}{2}\cos(b)\ddot{a}$$
$$F_{23} = 4\ddot{b} - 4\sin(2b)\dot{a}^2$$
$$F_{24} = -g\sin(b)$$

Appendix 5

The energy based identification method

The energy based method uses the principle of balance of energy. The energy change inside the system has to be equal to the energy that is supplied to the system. This balance law is correct for the Mechatonics Kit because there exist no other energy sources other than the motor. Therefore the equation of interest becomes

$$T - V = L = L(t) - L(0) = \int_0^t (u(t) - f(t)) \cdot \omega(t) \cdot dt$$

With

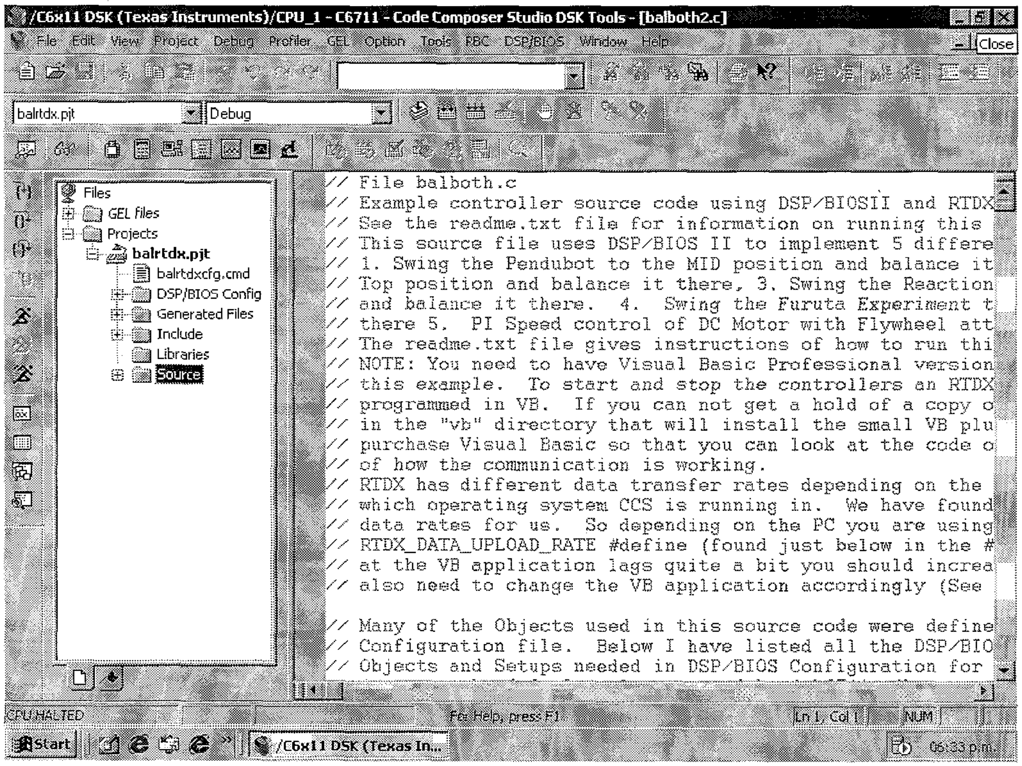
L = kinetic energy minus potential energy
u = control torque
ω = angular speed of the motor shaft
f = friction term

The basic idea is that in the energy equation the parameters of the system appear linearly. Then a lot of independent measurements are taken and the parameters are solved in a least square sense. The method is found to be very sensitive to noise. As can be seen from the relation above, the left-hand side of the equation contains only two elements. These are the values of the total energy considered at only two time instants. The presence of noise at these two time instants will produce large errors. These errors may be made less significant by averaging many measurements. Besides this negative effect, there exists no criterion that determines whether the excitation is a good one or not.

Appendix 6

Practical guide for operating on the mechatronics kit.

In principle the Mechatronics Kit has a rather advanced communication system. The communication goes through a board delivered by the company *Texas Instruments*. The board is capable of floating point operations on a very high speed. On the board there is a chip in which a program can be flashed. There is 128 kb of flash memory available. So the program cannot be any bigger than 128 kb. When the program is flashed onto the board it is possible to interface with the program by means of the top three white dip-switches on the board, the fourth one is just a dummy switch. Furthermore there is a reset button. Using the Mechatronics Kit this way might be very interesting for demonstration purposes, however it is not possible to record measurements. The program that is flashed into the flash memory at present time is bootflsh.c. See the readme.txt file inside the flash directory for more information on how to flash a different (for instance the same program with different controller gains or dynamical parameters) program onto the flash memory.
It is more useful to communicate with the Mechatronics Kit by means of a host computer. In this way data can be stored on disk space and later this data can be analysed off-line. For this way of communicating the manufacturer has also delivered some files. The main file is called balboth.c. The file is supposed to be run with the use of *Code composer studio,* from now on called CCS, and a supplemental Visual Basics interface, from now on called VP-application. Double clicking on the CCS-icon on the desktop gives figure 1. Starting the VB-application will give the screen in figure 2.



```
// File balboth.c
// Example controller source code using DSP/BIOSII and RTDX
// See the readme.txt file for information on running this
// This source file uses DSP/BIOS II to implement 5 differe
// 1. Swing the Pendubot to the MID position and balance it
// Top position and balance it there, 3. Swing the Reaction
// and balance it there.  4.  Swing the Furuta Experiment t
// there 5.  PI Speed control of DC Motor with Flywheel att
// The readme.txt file gives instructions of how to run thi
// NOTE: You need to have Visual Basic Professional version
// this example.  To start and stop the controllers an RTDX
// programmed in VB.  If you can not get a hold of a copy o
// in the "vb" directory that will install the small VB plu
// purchase Visual Basic so that you can look at the code o
// of how the communication is working.
// RTDX has different data transfer rates depending on the
// which operating system CCS is running in.  We have found
// data rates for us.  So depending on the PC you are using
// RTDX_DATA_UPLOAD_RATE #define (found just below in the #
// at the VB application lags quite a bit you should increa
// also need to change the VB application accordingly (See

// Many of the Objects used in this source code were define
// Configuration file.  Below I have listed all the DSP/BIO
// Objects and Setups needed in DSP/BIOS Configuration for
```

Figure 1: Code Composer Studio window

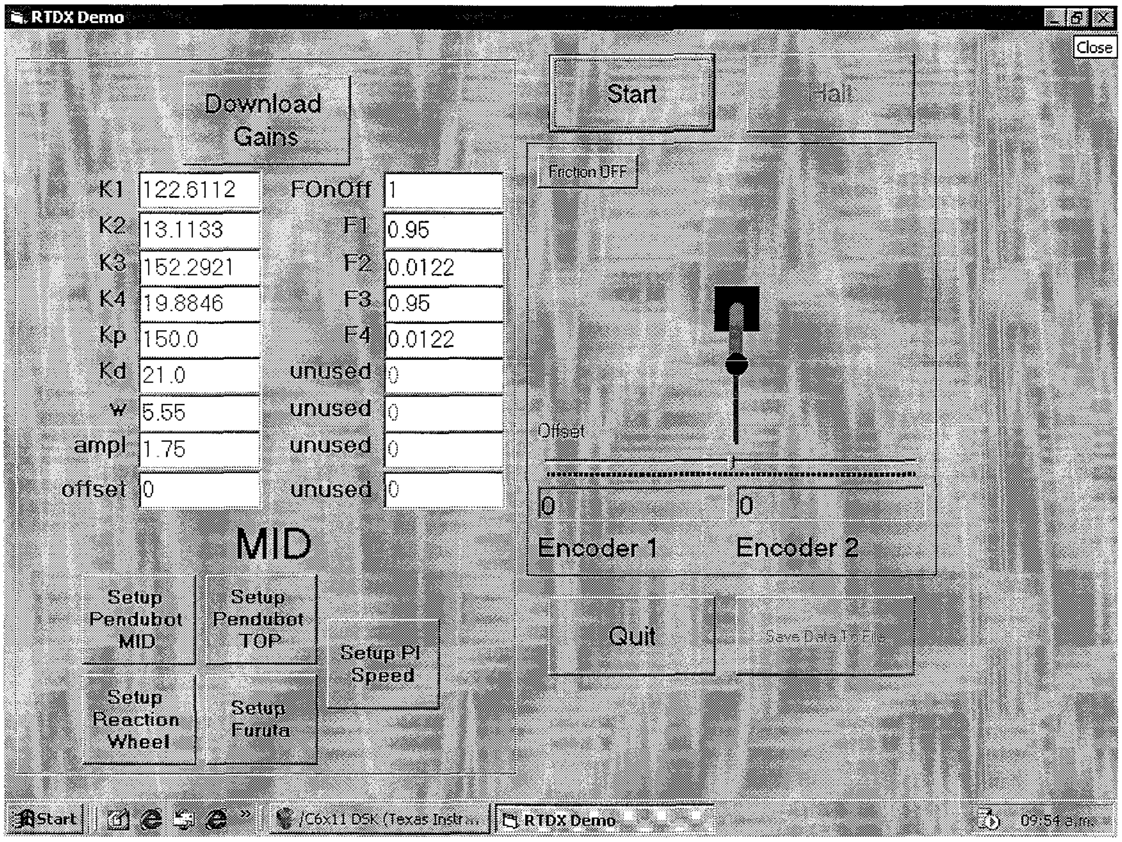
In figure 1 two different windows can be distinguished. The left window is of first interest. Whenever you want to start a project, just go to this window, and right-click on *Projects*, then click *Open Project.* The same action can be executed by going to the menu-bar in the top of figure 1 (go to *Project* and click *Open*). You can now choose a project to open. Look for the file *balrtdx.pjt.* CCS now loads the project. If you now click on the plus left from *balrtdx.pjt* inside the left window, the field will expand. As can be seen there will arise several folders. The *DSP/BIOS* folder contains the necessary settings for proper communication, as well as the *Gel files* folder (up in the top), another important file for proper settings of the host computer to communicate with the board is the file *balrtdxcfg.cmd.* It is highly advisable to stay out of these two folders and the file. The main prise of the Mechatronics Kit is for the building of these folders and their contents. In the files are all the addresses of ports and channels for communication. The settings for the *balrtdxcfg.cmd* were changed once, causing a re-installation because CCS did not work anymore. Certainly CCS is capable of much more than just controlling a mechanical system, but when you change one crucial fact that might seem unimportant to you, things might go wrong. Therefore only change these files after having read the tutorial, and when you know what you are doing. Furthermore there exists a folder named *Generated files.* In this folder are the files that were produced after building the project (see the sequel). Then there is a folder called *Include.* This is an important folder. In it are the files that are being referred to in the main c-file *balboth.c.* When you click on the plus next to the folder, you will see that there are many files included. The extensions of the files are *.h, .h62, .h64, .s62.* The *h* stands for the fact that the main c-file is able to use this file when it is needed. In these files are typical things like locations of addresses of ports and channels, but also specific (non standard c-language) commands are defined in these files, for instance the *rtdx*-commands. RTDX stands for Real Time Data Exchange. The host PC is able to communicate with the board, thus sending and receiving data at the same time, while it is also making computations and animations. The way of communication is said to be very advanced, the interested reader is referred to the tutorial. Especially have a look at the RTDX part of the tutorial. Inhere the hierarchy of tasks in the real time application is explained. What is important for now is that inside the *Include* folder are the functions that are used in the main c-file, and also the locations of ports and channels. Then there is only one folder left, the *Source* folder. In the *Source* folder are two files, *c6xdskdigio.c* and *balboth.c.* In *c6xdskdigio.c* code can be found for communication with the daughter card. The main c-file is *balboth.c.* Inside this file are 2225 lines of code. When you click on the plus next to the *Source* folder and then double click on the *balboth.c* file, then in the right window the code will appear. Everything you might want to change in controlling the mechanical system can be changed inside this file. Although the file is very long, there is much structure in it and it is full of comments.

Let's have a quick look at the source file and devide it in several parts.

- The first 72 lines are comments, in the comments are the settings of the *DSP/BIOS* configuration
- 73 to 88 are for the inclusions of h-files, thus allowing the usage of specific commands
- 93 to 112 is code for the LCD-screen, so not really interesting
- 113 to 126 contain code for enabling data storage and data transfer. RTDX_Data_Upload_Rate should be increased if the animation in the Visual Basics application lags behind from reality, so this is not really important

- 128 to 160 define functions and variables. Important is the definition of the controller functions in 128-134
- 161 to 349 defining variables. When using the Mechatronics Kit for some time, all variables will become evident. 344 to 349 relate to the pressing of the buttons in the VB-application. If you want to add or remove a vector from being saved, you should modify lines 298-303. If you want to increase the sample frequency, you should change line 201. If you want to increase the measurement time, change line 292.
- 350 to 469 are mainly used for the LCD-screen. But also read-RTDX is enabled.
- 463 to 1000 here 15 cases can be distinguished. Case 1 represents the action when you push the "*Setup Pendubot MID*" button inside the VB-application, see figure 2. If you have a look at the code belonging to case one, you can recognise that everything is initialised, meaning that case 1 to 15 are only executed when certain conditions are met, see line 467 and 468 for the conditions for case one. Case 2 initialises the *"Pendubot to TOP position"*. Case 3 represents the *Start* button. Case 4 is for the *Download Gains* button. Case 5 is for the "*Reaction Wheel*" button. Case 6 through 12 are reserved for the "*Save Data To File*" button. Case 13 is for the "*Furuta*" button. Case 14 is also for "*Saving*" action, and case 15 finally is for the "*PI Speed*" button.
- 1001 to 1074 is just for letting the LED's blink and playing some music, so this is absolutely not important
- 1079 to 1170 is for assigning the right controller function and storing the data to the appropriate arrays. If you want to change the vectors that need to be saved, you should modify the file here. Also if you have added an extra controller button in the VB-application you should change the source file here.
- 1177 to 1941 is what it is all about. In these lines the control functions are specified. Five different controller functions can be distinguished.
  Controller_midISR starts at line 1177 and is for controlling the *pendubot* to the MID position.
  Controller_topISR starts at line 1326 and is for controlling the *pendubot* to the TOP position.
  Controller_ReactionISR starts at line 1468 and is for controlling the *reacton wheel.*
  Controller_FurutaISR starts at line 1606 and is for controlling the *Furuta pendulum.*
  Controller_MotSpdISR starts at line 1863 and is for controlling the *inertia wheel.*
- 1943 to 2225 contains very specialised code for sending data through the right ports and channels, but also for generating music and sending information to the LCD-screen. It is advisable to keep the code for these lines the way it is right now if you don't have much knowledge about the board and the communication between the board and the host PC.

So it can be concluded that the most important part of the very large source-code file is contained in the definition of variables part, lines 160-350, and the part for the controller to be used, lines 1177-1941. If you want to change a controller, make sure when you are using new volatile variables that you define them in the beginning of the source file, thus between lines 160-350. Else the debugger will let you know you made an error in editing the source file.



K1 122.6112
K2 13.1133
K3 152.2921
K4 19.8846
Kp 150.0
Kd 21.0
w 5.55
ampl 1.75
offset 0

FOnOff 1
F1 0.95
F2 0.0122
F3 0.95
F4 0.0122
unused 0
unused 0
unused 0
unused 0

Encoder 1 0
Encoder 2 0

Figure 2: VB-application window

Let's analyse one controller function. We take the first one, the controller for the *pendubot* to mid position. So we start analysing at line 1177. When you want to go to a specific line in the file, go to the right window of figure 1, right click in the field, and click *Go to.*

- Lines 1181 to 1196 give the user the opportunity to push the button "*Download Gains*" inside the VB-application. Obviously the gains can be downloaded during the experiment! Line 1197 defines the time the non-linear controller has to follow the reference sine.
- Line 1201 is the command for reading the encoders. The command is defined in the header file *c6xdskdigio.h* inside the *Include* folder. In this header file de constants pi and gravitational acceleration are also defined. Have a quick look at the file to get familiar.
- Line 1204 to 1210 scales the values of the readings of the encoder connected to the second link back to [0,2pi].
- Line 1213 and 1214 calculate the raw velocity, by numerically differentiating the positions of the first and second angle.
- Line 1216 and 1217 filter the raw velocities a little bit to smoothen out the noise.
- Line 1222 to 1299 is reserved for the actual control low. The relevant parts can clearly be recognised. Line 1222 says that when you are already using the linear controller you don't have to switch to it anymore. If the conditions for the linear controller are satisfied (see the comment lines 1223-1225) the control signal that is to be applied equals line 1229. This control signal is equal to the one on line 1275. If the conditions are not satisfied the non-linear controller is active. Lines 1242 to 1271 exactly describe the controller as reported in the document on the *pendubot* (and section 2.2 of this report). The applied control signal is equal to

line 1271. Lines 1289 to 1295 define the friction compensation. Line 1297 and 1298 define the saturation on the control signal. Lines 1302 to 1315 command the states and applied control signal to be stored.

Now you might want to change the control law. If you just want to change the parameters, you can do this inside the VB-application. If you changed the parameters inside the VB-application, you just press the button "*Download gains*" and then press "*Start*". If you want to change the structure of the control law, you have to modify the source-code. If you want to apply an open loop input for instance, you just change line 1271 for "u=3", meaning that an open loop signal of amplitude 3 is applied, as long as the conditions for the linear controller to take over are not satisfied. After changing this line, you want the change to have effect. Therefore the new execution file has to be built. So when you are ready with modifying the source-code, you must click on the *"Rebuild all"* Icon, or go to the menu bar, select *Project* and then *Rebuild all.* Because the source code is rather long, this will take a few seconds. When building there appears a new window. This is the Log-command window. During the experiments several messages will appear in this window. If the building is completed and no errors are detected, you have to go to the menu-bar and select "*File*" and then "*Load Program*". The name of the program to be loaded has the same name as the project, the extension is .out, it is contained in the subfolder "*Debug*". Loading the program takes a few seconds. When this is finished, you have to go to the menu-bar and select "*Debug*" and "*Run*", or just press the upper blue man totally on the left in figure 1. Now the LCD-screen says "Waiting for VB start", so we go to the Start-menu of windows, and open the program *RTDXproj*. The VB-application will appear. Pressing the button "*Setup Pendubot MID*", followed by pressing "*Start*" will cause the pendubot to produce the response to the open loop signal of amplitude 3. After pressing the button "*Halt*", when we have seen enough of the experiment, we have the opportunity to save the measured data. Obviously you then have to push the button "*Save data to file*". You might want to give the file the extension *.txt* because then you can open the file in Windows Notepad, select all, and copy it into an m-file of Matlab, run this m-file and plot the results. The vectors that are saved are put into a matrix called Y.
When we want to perform another experiment, we first have to press the button "*Setup pendubot to MID*", or the button "*Setup Pendubot to TOP*" if you wish to change your goal. When you are through with experimenting, just push the button "*Quit*", and the VB-application closes. CCS however is still running. You can see this from figure 1. In the left down corner it then says "*CPU Running*". To Halt the running of CCS, just go to the menu-bar and select "*Debug*" and then "*Halt*", or just push the middle blue man totally in the left part of the CCS-window. If you don't change the c-file and you want to run the program again after a few minutes, you don't have to rebuild and load the program, but you can just push the blue man which stands for "*Run*"

Quick start

- Open project
- Edit source-file
- Build program
- Load program
- Run
- Start VB-application

- Initialise
- Start

Practical aspects

Pendubot

Building up the *pendubot* mechanically is not a very demanding task. The thing to take care of is connecting the second link to the first link before you attach the first link to the motor shaft. When you do it the other way around, the motor might get damaged. Another point of concern is the routing of the cables. The *Mechatronics Kit user's manual* gives good instructions for routing the cable considering the *pendubot* configuration. Before starting any experiment, make sure that the links are in the absolute downward position. This is rarely the case and thus many times the cause of bad performance.

Reaction Wheel

When building the *Reaction Wheel* configuration first get the motor off the base plate, then attach it to the big pendulum. Attach the encoder to the base plate. The pendulum should not be attached to the encoder shaft yet! There should be only one possible way of attaching the motor to the pendulum. Then attach the *inertiawheel* to the motor shaft. When this is done attach the pendulum to the encoder shaft. Next thing to do is to make sure that the encoder of the motor is clicked into the hole that says "encoder2". The encoder that is fixed to the base plate should be clicked into "encoder 1". For all the other experiments this is just the other way around! The routing of the cables described in *Mechatronics Kit user's manual* is not good. You'd better try the way of figure 3. Make sure that you don't tape the cables to close to the pivot (encoder axis of the base plate). To check weather the cables are fixed in a right way push the cable from left to right with your finger. If the pendulum does not move too much, the routing is good. Be careful that when the reaction wheel is turning its rounds, the cables do not get stuck around the corners of the base plate. Check this before you experiment, by lifting the pendulum with your hand and determining that the cables indeed don't get stuck at the corner of the base plate when the pendulum moves back.

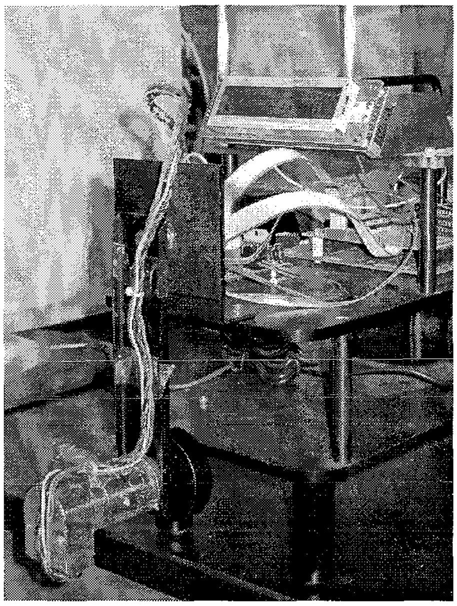


Figure 3: Picture of the Reaction Wheel

Furuta Pendulum

For building the *furuta pendulum*, you first have to take off the base plate and replace it by the base plate of the *furuta pendulum*. The long side should be directed in the vertical position. Screw the original base plate on top of the base plate of the *furuta pendulum*, and attach the motor from beneath. Make sure the cables of the motor are outside the range of the pendulum's movement. This can be achieved by turning it around the down left corner when standing in front of the set-up. The cables are just long enough for doing this. Tape the cables so that they can not slip away. The cables of the encoder can be routed in the way described in *Mechatronics Kit user's manual*. Again be sure to connect the pendulum to the encoder before you connect the arm to the motor shaft. You might have to use some extra power to screw the pendulum to the encoder shaft. The fitting is not very precise.

General

It might happen that the cables of the power supply of the motor slip out of their holes. As you then will notice it is very difficult to put them back in. Therefore first remove the LCD-screen, and then unscrew the screws that are connected to the holes in which the cables should be put in. When the screws are loose, put the cables in. Black should be in hole 1, red should be in hole 2.Then turn the screws back in. Then put the LCD-screen back in its place. To prevent the cables from slipping out again a tape has been put over the cables.

When taping the cables of the encoder to the base plate and the links of the system under consideration you should always be careful that the cables don't slip against each other or make collisions with the base plate. Unfortunately in some applications this is not avoidable.

Practical aspects on the identification.

During the identification of the *pendubot*, several m-files were created for analysing and processing the data. Inside the m-files comments can be found, that together with the description of the algorithm (which can be found in this report) describe the actions properly.
Whenever you save a measurement from the VB-application, you should give it the extension .txt. Then open the file in an editor for txt files, select all, copy all, and paste it into the m-flile exciter2.m, which can be found in the experiment-*pendubot* directory (this can also be an empty m-file). Execute this m-file and perform the following command for plotting the results of the measurement:
Plot(Y(:,1),Y(:,2)); This will plot the position of the first joint as a function of time.

The global action for the estimation of the *pendubot* (simulation data) is the following:

- Use test14.m together with test2.m to generate the measurement data, and the determination of the condition number of the scaled regressor matrix. When you think the condition number is low enough, you have to create a structure in the command line of Matlab: *name=structure('data',Z,'time',t);* Then save this structure: *save name name.*
- Now go to test13.m to process the data. Test13.m needs the just saved structure, and gives a structure with name2 as output. When test13.m is finished, save the structure with
  Save name2 name2.
- Go to penduest.m, this file needs the structure name2, and computes everything needed for the optimization, and also performs the optimization. It also needs valdata.mat, if you want to compare results, plus the file valtest.m
- The same procedure can be used for the other configurations as well. However as is mentioned in the report, the *reaction wheel* configuration is not suited for this estimation routine.
- Estimation on real data can be found in the experiment directory. The files work exactly the same as the simulation data files. If the files do not run properly, this is most probably due to the fact that you do not have measurement data available.
- Files ending with a 5 are the good ones. Files ending with a 6 are used for estimating with a slightly different friction model. No significant better results were achieved with this other model, however, the computational time is much higher.

Furthermore the optimization of the trajectory is implemented
Needed files:

- Sweverhoofd.m
- Costdet.m
- Dynamlos123.m
- Traj123.m

Comments belonging to these files can be found inside the m-files.

For the optimization of the single pendulum parameters there are the files ending with –single.

Estimation of the friction parameters (based on measurement data) can be found in the files friction.m and frictionest.m.

For questions, feel free to ask me at j.a.c.meesters@student.tue.nl.