IMPROVING STEADY STATE STABILITY OF A PENCIL BALANCING SYSTEM

eingereichte
STUDIENARBEIT
of Project Laboratory Neuro-Robotics

Benoit PHILIPPE
Pierrick AUSSERESSE
Shahin AMIRIPARIAN

Lehrstuhl für
STEUERUNGS- und REGELUNGSTECHNIK
Technische Universität München

Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss
Univ.-Prof. Dr.-Ing. Sandra Hirche

Projektleiter:	Prof. Dr. sc.nat. Jörg CONRADT
Betreuer:	M.Sc. Indar SUGIARTO
Beginn:	24.10.2011
Abgabe:	20.01.2012

Abstract

The purpose of this project is to improve the steady state stability of a pencil balancing system. The first phase of this project involves modeling and implementing of an inverted pendulum system. Aim is to provide a stable control loop to keep a pen in unstable equilibrium, including human pen positioning at startup. Complete analysis of mechanical system is provided through linearized model on operating point, using Matlab Simulink. Our control loop is closed through DVS cameras, providing angle (slope of the captured line) and base position of the moving object. Different types of controllers (from simple PID to self-adaptive controllers and Fuzzy) have been analyzed in detail. In order to reduce the extra motion of the cup and make the system more stable we implemented a low pass filter on the servomotors inputs. This filter can be manually in the configuration panel ajusted. Another point to be discussed is the startup conditions. The humain slow perception makes it hard to work with a system, which moves with a higher frequency than ours. By means of the implemented low pass filter during this project it is much easier to stablize the pencil.

Contents

Chapter 1

Introduction

The humain visual perception interprets information and dynamic world from the effects of visible light reflection to the eye. Sometimes the changes in environment are faster than our reaction. So that we always tried to design systems improving our sens of perception. For exemple the balancing of an object. As a desmonstration of controller design, the object balancing has been used in many fields. Therefore we need vision sensors with high frame rates. For holding a pencil on its lead with help of a rebot system we need motors and cameras, which can measure the pencil mouvements and bring it to a desired position.

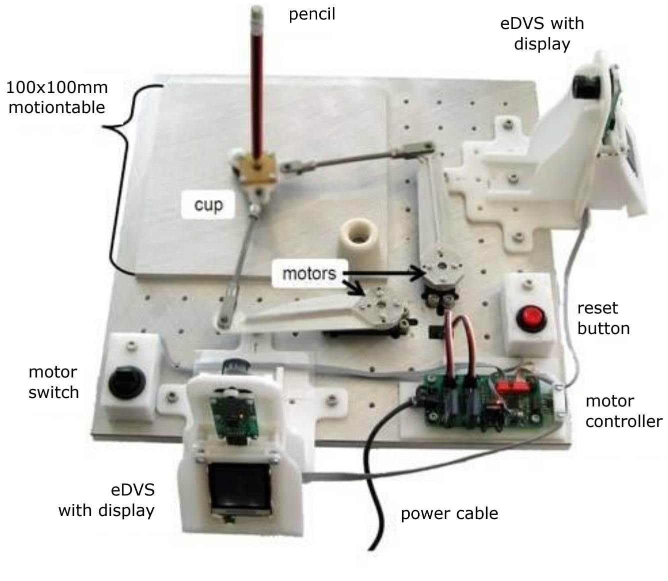


Figure 1.1: Pencil balancing system

The proposed system is composed of two dynamic vision sensors (DVS), two servomotors (BLS 451) and a microprocessor (ARM7). The both vision cameras are in a right angle to each other position. This configuration allows to generate a 3D estimation of the pencil position. The camera provides fast visual responses from the moving pencil on the actuated table. The microcontroller unit recives the data from the sensors and evaluate them for computing the desired cup position. This new position should reduce the slope in order to straighten the pencil.

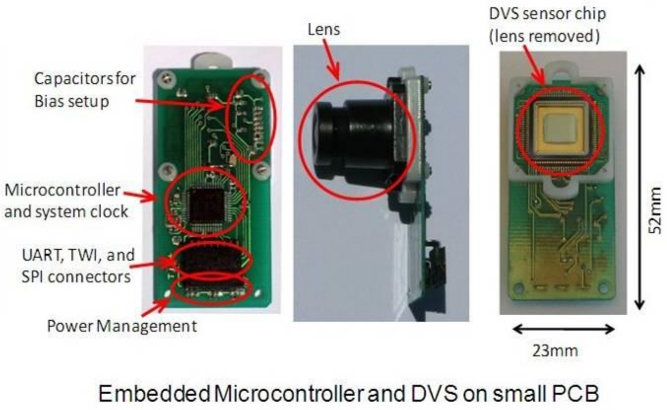


Figure 1.2: DVS camera with microcontroller

Chapter 2

Advencement and simulations

2.1 New coordinate transformation

Board embedded algorithm that calculate pencil position is in Cartesian coordinates, but actuator are motor that could only produce angular output. At first, it was supposed that as long as movements remains smalls, polar coordinates can be consider as Cartesian. Even if this approximation is quite effective (less than 10% error for +/-2cm span), this introduce coupling between X and Y axis, coupling that act as pure noise for other channels. One of the first visible effect in this situation is that it is impossible to place pencil base below its tip if you're not located at table center.

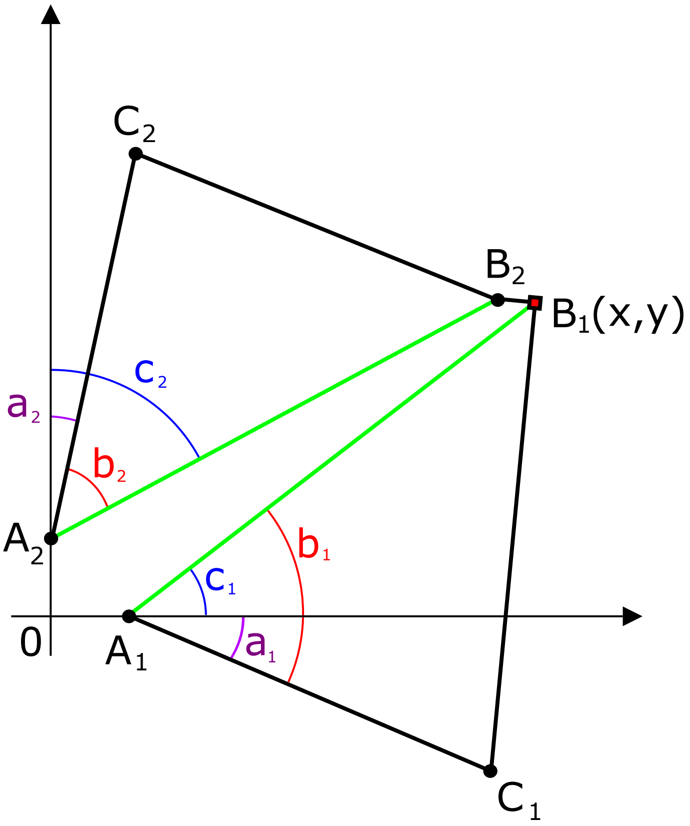


Figure 2.1: New coordinate calculation

In order to place base at the exact Cartesian coordinate required, we have to calculate servos controls angle in respect to the base position. This mean that for this algorithm, base position is an $A = \pi r^2$ input variable, constant and known. Next step is then to split the system in two. Indeed, setting base as a known parameter allow to totally dissociate the two servos controller calculation. Let's start with servo a_1 :

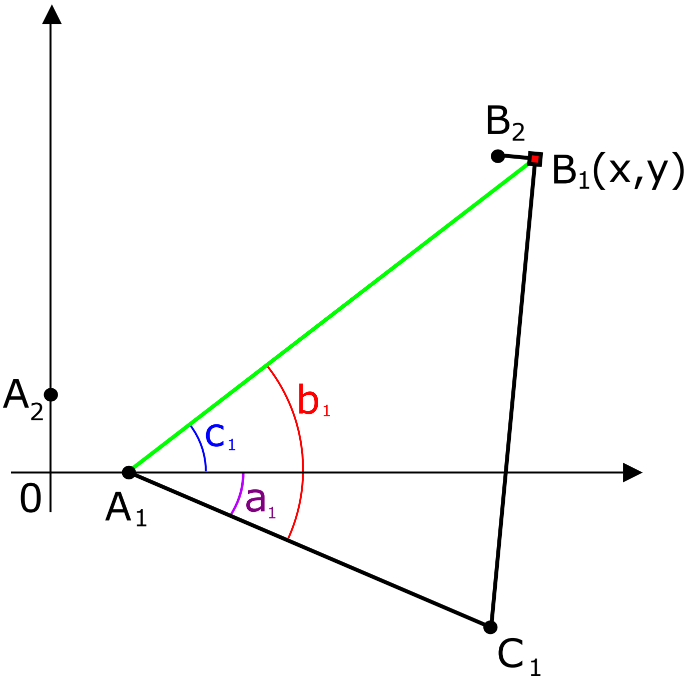


Figure 2.2: New coordinate calculation

Distance A_1 to origin is know, and will be labeled as d_1. Distances A_1C_1 and C_1B_1 are the robot arm, so they are also know. In triangle $A_1C_1B_1$, we can make use of Al-Kashi theorem :

$$C_1B_1{}^2 = A_1C_1{}^2 + A_1B_1{}^2 - 2.A_1C_1.A_1B_1.\cos(b_1) \tag{2.1}$$

Knowing B_1 position make it possible to calculate distance A_1B_1 as well, so only unknown is b_1 :

$$b_1 = \arccos\left(\frac{A_1C_1{}^2 + A_1B_1{}^2 - C_1B_1{}^2}{2.A_1C_1.A_1B_1}\right) \tag{2.2}$$

at the same time,

$$c_1 = \arctan\left(\frac{x - d_1}{y}\right) \tag{2.3}$$

So we get :

$$a_1 = b_1 - c_1 \tag{2.4}$$

Once calculation is done for first servo controller, a adaptation is to be made in order to calculate the second one. Indeed, for second servo, triangle is not defined using the base position itself, but a slightly off position B_2.

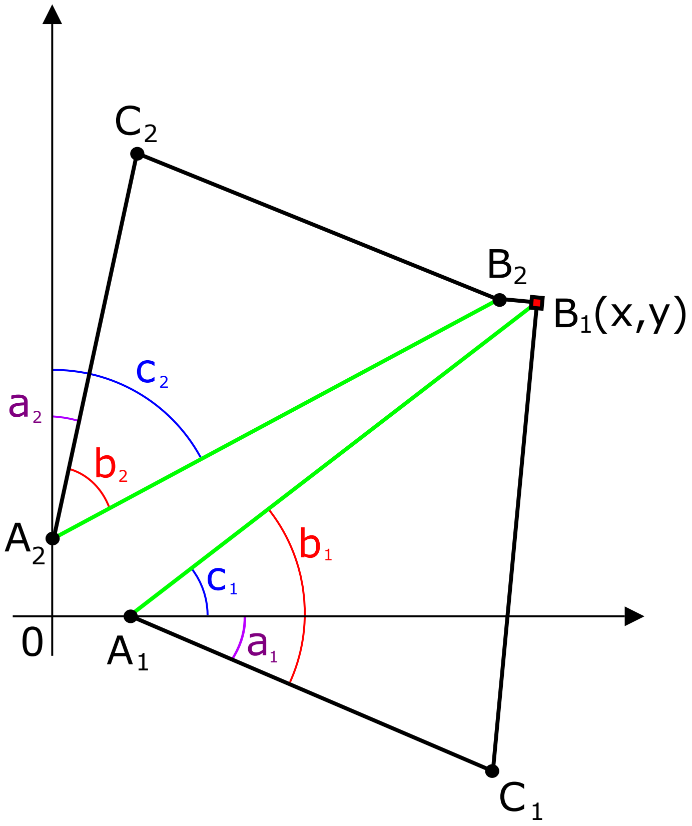


Figure 2.3: New coordinate calculation

In order to get B_2 coordinate, we have to make use of a_1 calculation. using sin and cos, C_1 position can be calculated :

$$C_1 = \begin{pmatrix} A_1C_1.\cos(a_1) + d_1 \\ A_1C_1.\sin(a_1) \end{pmatrix} \tag{2.5}$$

As $\overrightarrow{B_1B_2}$ and $\overrightarrow{B_1C_1}$ are orthogonal and have constant norm, we can calculate vector $\overrightarrow{B_1B_2}$ out of $\overrightarrow{B_1C_1}$ using rotation and scalling :

$$\overrightarrow{B_1B_2} = \overrightarrow{B_1C_1}.\frac{B_1C_1}{B_1B_2}.\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \tag{2.6}$$

Knowing B_1, we can then deduce B_2. Problem is then exactly identical to the first on, Al-Kashi theorem allow us to get c_2 in triangle $A_2B_2C_2$, and then a_2 after b_2 is calculated.

$$b_2 = \arccos\left(\frac{A_2{C_2}^2 + A_1{B_2}^2 - C_2{B_2}^2}{2.A_2C_2.A_2B_2}\right) \tag{2.7}$$

$$c_2 = \arctan\left(\frac{x_2 - d_2}{y_2}\right) \tag{2.8}$$

$$a_2 = b_2 - c_2 \tag{2.9}$$

These calculations give angles in radiant, but Servo controls are commanded through ON time modulation. There are also offsets on exact angle due to assembly variation of servo controllers wheels. To correct this, a trimming phase is then needed. To achieve this, we decided to use the already trimmed board, as there are no feedback signal that could allow us to trim angles without having the board running. Keeping the old regulator running, we provided in parallel the new coordinate transform system. By use of UART interface trimmed then servo offset and angles in order to match old and new system angle for position (0,0). Once achieve, we are sure that all servos offset are compensated. Radiant to ON time modulation ratio can be directly provided through system gain, so we used already provided MACRO to trim it. Old model and new one having similar gains, no action were required there.

2.2 Low pass on the angle

The system is composed of several functions, which receive the data from DVS cameras and after some evaluations forward them to the servomotor. The desired position of the cup will be calculated from the slope and base values received from the DVS cameras. The desired position is calculated in Cartesian coordinate. As the servomotors need inputs in angles, we have to transfer the Carstesian coordinates. Now we need to attenuate frequencies higher than current cutoff frequency. This means taht we have to use a low pass filter. We also programmed a function in the configuration panel, which makes it possible to change the low pass factor during the running system. If the value will be null, the low pass filter would be deactivated. By adjusting values bigger than defined range the system will stop working. The reason is, that the mouvement amplitude exceeds the geometrical limits on the table.

2.3 Simulations

At first we tried to modelise all the system with Simulink and so to use the simulation with diffferent kind of controllers (e.g. PID, FUZZY). We planned to modify the parameters of different controllers easier and faster and then select the best one of them for the system.

For the modeling we ajusted the system with an inverted pendelum. Actualy the pencil is seen by the both DVS cameras like an inverted pendelum. The first camera sees the pencil in the x,z direction and the second camera in the y,z direction. The purpose was to optimize the system and fit the reality. So we tried to simulate an inverted pendelum with simulink.

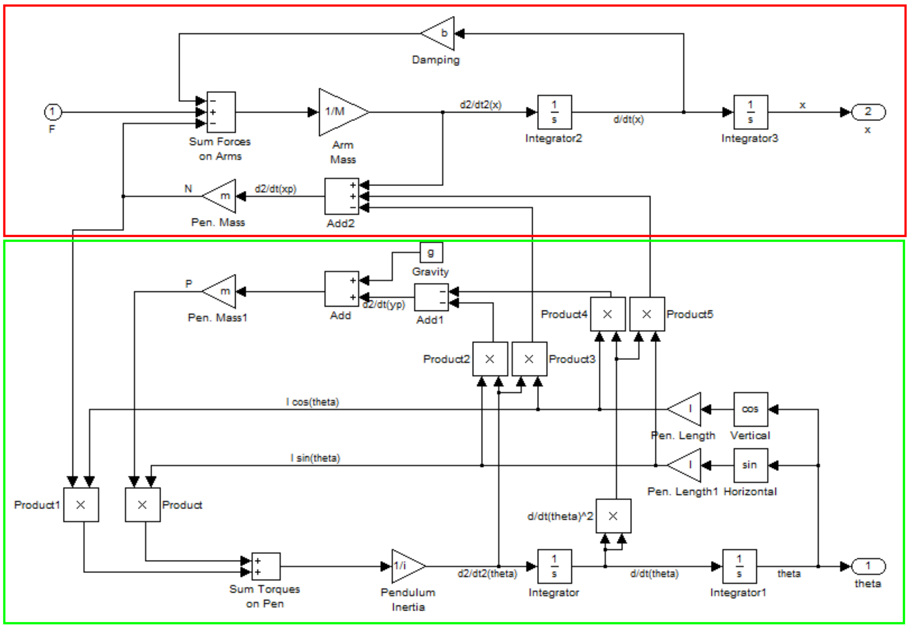


Figure 2.4: Simulation of an inverted pendelum

For the simulations we needed to couple mecanical equation of the pencil and the cup. We included the system into another one to close the loop with a PID controller.

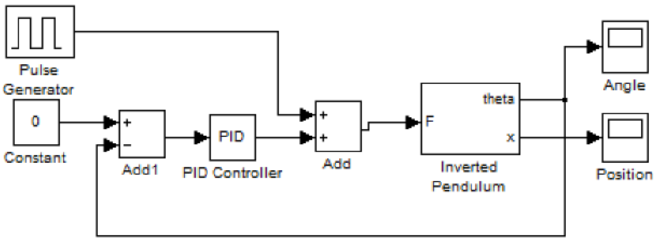


Figure 2.5: Simulation of an inverted pendelum with a PID contrller in the closed loop

There were too much parameters to fit the reality like for exemple the friction coefficient and the masse of the cup.

The system could be identified even without modelisation. We can split the system in two substystem. The first one is the system, which has two inputs (the position and the slope of the pencil) and two outputs (both servomotors). The second system characterise the pencil.
The following data of the pencil balancing system have been saved : the slope and the base position of the pencil from both DVS cameras, and inputs from both servomotors. With the system identification toolbox we created a nonlinear model with

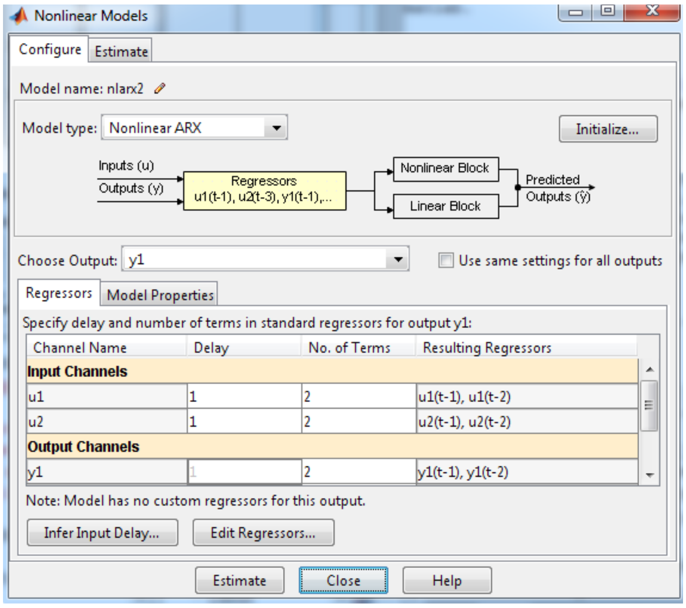
initials setting; number of pass inputs and outputs had been adjusted at 2.

List of inputs :

DVS0Slope	: Slope captured by the first camera
DVS0Base	: Base position of the pencil captured by the first camera
DVS1Slope	: Slope by the second camera
DVS1Base	: Base position by the second camera

List of outputs :

servoBSignal	: Commande for the first servomotor
servoTSignal	: Commande for the second servomotor



Model name: nlarx2

Model type: Nonlinear ARX

Choose Output: y1

☐ Use same settings for all outputs

Specify delay and number of terms in standard regressors for output y1:

Channel Name	Delay	No. of Terms	Resulting Regressors
Input Channels			
u1	1	2	u1(t-1), u1(t-2)
u2	1	2	u2(t-1), u2(t-2)
Output Channels			
y1	1	2	y1(t-1), y1(t-2)

Note: Model has no custom regressors for this output.

Figure 2.6: Setting for the nonlinear model in the System Identification Tool

However the model didn't fit the reel system (see bellow the fit-coefficient for the different output). The concordance between outputs from the model and reel outputs is too low. We changed the number of pass inputs and outputs, the delay, and other parameters to optimize the model estimation, but the result wasn't better.

DVS0Base	: 79,28%
DVS1Base	: 78,77%
DVS0Slope	: 49,88%
DVS1Slope	: 24,86%

Meanwhile we made some experiments with the system. We put our finger on the both servomotor and saw that the system was more stable. In fact we applied a small force with our finger, which slows down the system and reduces the amplitude. We though about a solution to modelise this effect. A low pass was the first solution.

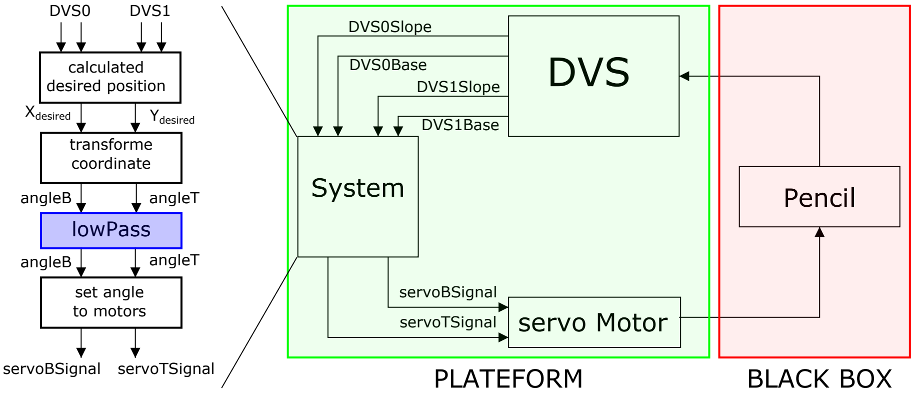


Figure 2.7: Functional diagram with implemented lowPass

We implemented the low pass on the angle (see above) and appointed different level. We could place the low pass on the servoBSignal and servoTSignal, but the transformation between angle and servoSignal is linear, so it would be have the same effect. For each established level we saved following data : the slope and base from both DVS cameras, inputs servomotor signal, and angleB angleT.

For each level of the low pass we computed servomotor inputs signal (servoBSignal and servoTSignal) with the fast Fourier transformation (FFT). We note lowPassXX with XX the low pass level.

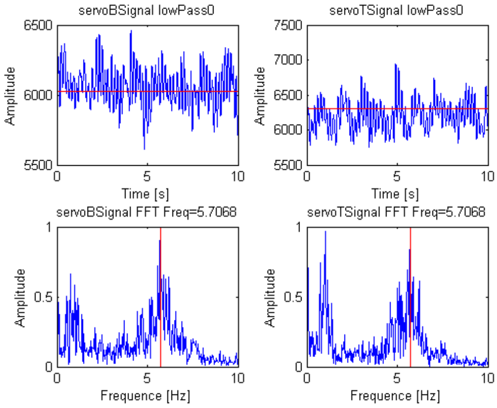


Figure 2.8: FFT without lowPass (lowPass 0)

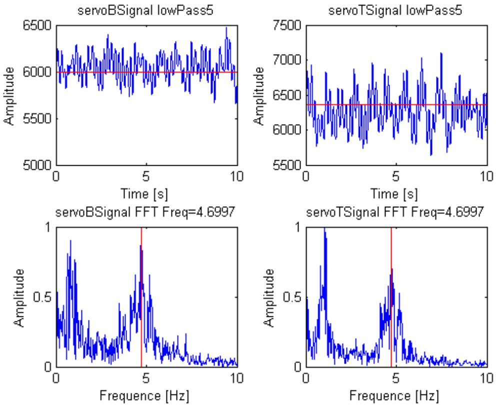


Figure 2.9: FFT with lowPass 5

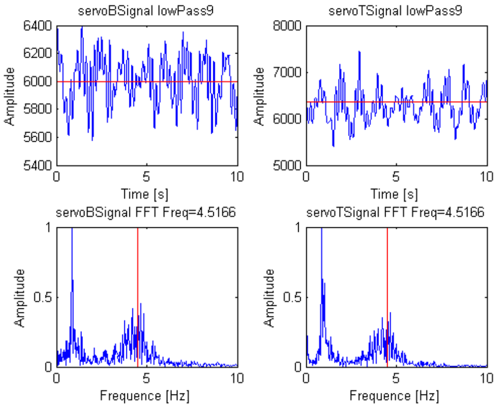


Figure 2.10: FFT with lowPass 9

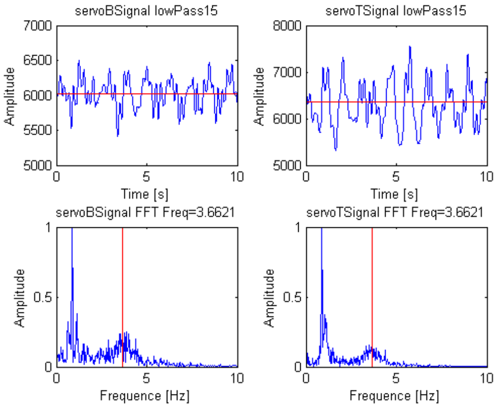


Figure 2.11: FFT with lowPass 15

We distinguished 2 frequencies, the first one by 1 Hz and the second according to the low pass level. The second frequency represents the motion speed of the cup. The more is the value of low pass, the slower is the cup motion. Furthermore the total energy decreases despite the level increases. By the way we constated that for low pass with level 15 the system isn't stable anymore.

However we can not affirme that the pencil is stable just because the frequency is lower. The slope of the pencil have to stay by null and the amplitude have to be as small as possible. We analyzed the histogram of the slope captured by the DVS cameras (DVS0Slope and DVS1Slope).

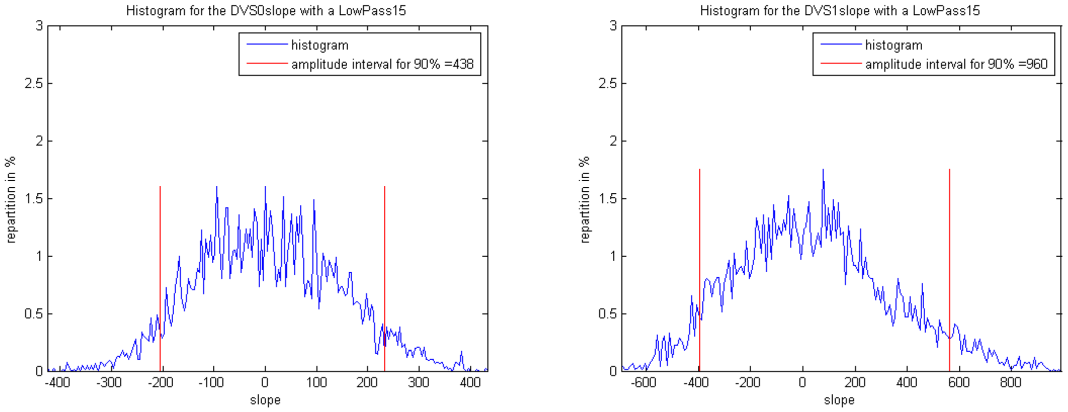


Figure 2.12: Histogram for DVS0Slope (left) and DVS1Slope (right) with lowPass 15

These histograms represent the slope repartition for different low pass level. The two red lines have been calculated in order that 90% slope values are present between the two lines. The distance between these lines define the slope amplitude, that is proportional with the standard deviation. We did it again for each low pass level.

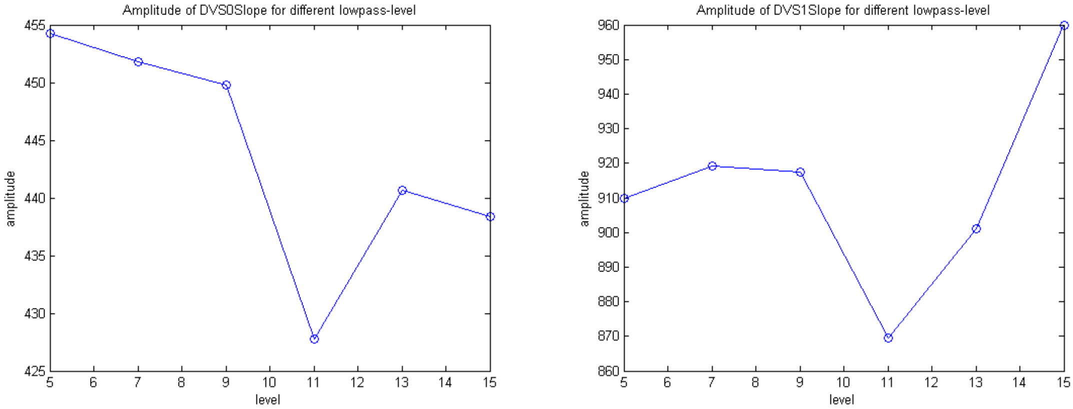


Figure 2.13: Amplitude of DVS0Slope (left) and DVS1Slope (right) according to the level of the low pass

The amplitude curve presents a local minimum for a level by 11. For this value we also saw that the system was more stable as other low pass level.

Chapter 3

Discussion

At the beginning of this project we were faced with a pencil balancing system, which could hold the pencil on its lead just for short time. It was even hardder to put the pencil on the cup of the robot table. So we decided to reduce the motion of the motors. For the first measurements we just mechanically put some loads on the motor arms. Then we saved the measured values from the sensors on the computer to build a simulation of the system. By comparing its behaviour before and after putting the loads we decieded to place a low pass filter. The filter we used was just a first order low pass. Because of the lake of time we couldn't test the other filter. We also had some problems with the system identification toolbox. The best fit-value between the model outputs and the reality was by 80% for the DVS-Base and by 45% for DVS-Slope. We tried to solve the problem by changing settings of the tollbox, but no results were better than the first one. We decided to analyze and optimise the stability of the low pass wihtout working with the system identification toolbox.

Chapter 4

Summary

During this project, we have improved the stability of the pencil balancing system. We have devided the tasks of this work into three phases : characterizing and modifying, implementing and evaluation phase. In the first step we have characterized the servo motors, then we tried to modify the PWM and actuator setups, in order to reduce the existing motions. After we found the statement of the problems with the help of our measurements and simulations we implemented some algorithms on the processor. After each implementation we saved the results to test and evaluate the system performances. Then we figured out that the algorithm of coordinates transformation could give us more satisfying outputs. But it couldn't fulfill our expectations. Therefore we designed another more effectiv solution which attenuate high frequencies. Finaly the conbination of these both algorithms could stabilize our system.

List of Figures

Bibliography

[BSM] L. Kucera B. Sprenger and S. Mourad. Balancing of an inverted pendelum with a scara robot.

[JCa] M. Cook J. Conradt, T. Delbrück. An embedded aer dynamic vision sensor for low-latency pole balancing. http://www.ini.ch/ conradt/projects/PencilBalancer/.

[JCb] M. Cook T. Delbrück J. Conradt, R. Berner. An embeddeb aer dynamic vision sensor for low-latency pole balancing.

[JCc] R. Berner P. Lichtsteiner RJ. Douglas T. Delbrück J. Conradt, M. Cook. A pencil balancing robot using a pair of aer dynamic vision sensors.

[JCd] R. Berner T. Delbrück R.J. Douglas M. Cook J. Conradt, P. Lichtsteiner. High speed pole balncing with only spike-based visual input. http://www.ini.uzh.ch/ conradt/publications/NIPS2008-JConradt.pdf.

[MA07] G. Moussouami and P. Aupoix. Etude du pendule inversé. 2006-2007.