KTH Electrical Engineering

Wireless Inverted Pendulum

AITOR HERNÁNDEZ, JOAO FARIA AND JOSE ARAUJO

Stockholm May 6, 2013

TRITA-EE 2011:XXX
Version: 0.1

Contents

Contents i

Acronyms 3

1 Introduction 5

2 Requirements 7

3 Scenario 9
- 3.1 Common installation 9
- 3.2 Sensors, Controller and Actuator 13
  - 3.2.1 Installation 13
- 3.3 Sensors and Actuator 14

4 Results 17
- 4.1 Labview 17
- 4.2 Matlab/Simulink 17

References 19

Acronyms

GUI	Graphical User Interface. 11, 12
OPAM	Operational Amplifier. 5
SPI	Serial Peripheral Interface. 5
UPM	Universal Power Module. 1, 3, 8–10

Chapter 1

Introduction

As we saw in documents [5, ?], we have different processes used for academic purpose. In this document, we show the inverted pendulum provided by Quanser running wirelessly. It explains the tools, software and knowledge that you need to run the wireless inverted pendulum.

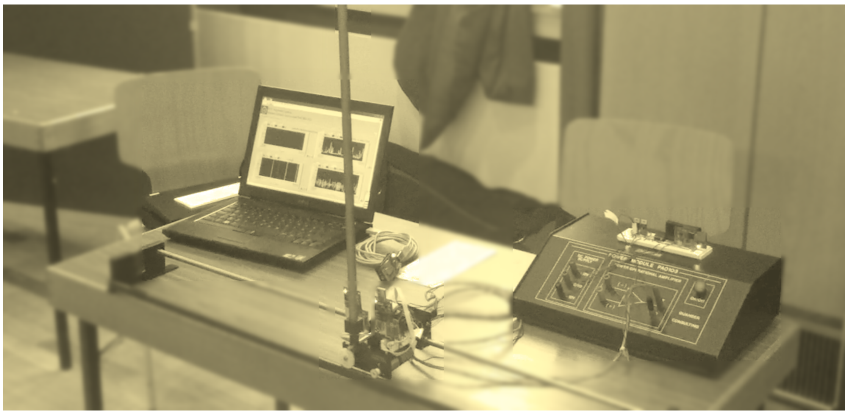

Figure 1.1: Inverted pendulum with the UPM and a computer to show the performance analysis

Figure 1.1 shows the scenario with the inverted pendulum, the Universal Power Module (UPM), and the computer to analyse the communication performance. The inverted pendulum has two motes connected to two sensors which are measuring the pendulum angle and cart position separately.

In this document, we do not focus on the modelling and control laws used and implemented in this scenario. For information regarding the modeling and simple control laws we recommend the following references from Quanser [7, 9, 8]. For more information check the following document [2, Sec. 6.1] or [1].

The code and software are available on the following URL:

http://code.google.com/p/kth-wsn/source/browse/trunk/kth-wsn/apps.inverted-pendulum

To compile the applications we recommend the use of the latest version of TinyOS that you could find on the following URL:

http://code.google.com/p/tinyos-main/

Chapter 2

Requirements

For this process we need special tools and hardware. The specific schematics for the hardware is provided in this document.

First of all, it is mandatory that you have read the following documents:

- *Getting started with TinyOS at the Automatic Control Lab* [4]
- *Communication between PC and motes in TinyOS* [3]

Below we show a list of platform requirements:

- Quanser Single Inverted Pendulum with its UPM
- 2 (+1) or 3 (+1) motes (Telosb, TmoteSky or MAXFOR) depending on the scenario
- 1 Zolertia mote for the controller
- TinyOS properly installed
- Counter board with the LS7366R [10]
- Adaptation circuit for the UPM - Mote
- Source code for the motes `http://code.google.com/p/kth-wsn/source/browse/trunk/kth-wsn/apps.inverted-pendulum/actuation-included/`
- LS7366R Driver implementation
- LabView with the Vi which shows the communication performance. `http://code.google.com/p/kth-wsn/source/browse/trunk/kth-wsn/apps.inverted-pendulum/LabView`

CHAPTER 3

Scenario

There are two possible scenarios to run the inverted pendulum wirelessly. The difference between them is the number of links that are involved. In the first case, we have the sensors sending data to the controller, and the controller sends the control voltage to the actuator. In the second case, we only have one link between the sensors and the controller, which is the actuator as well. In the following sections, we show them with more detail.

For the controller we need an special mote, with more computational capabilities. The Zolertia [6] uses the new microprocessors series and it is fully compatible with the rest of the code already implemented. If we do not use this mote, we could face an unexpected behaviour because we do not have enough time between sample to compute the Kalman filter and the LQR controller.

Before focusing in the different scenario, there are some installation parts that are common.

3.1 Common installation

Boards In this process, we need two different boards. The counter board is used to place the Serial Peripheral Interface (SPI) external counter (LS7366R [10]) and connect it to the mote.

Figure 3.1 shows the schematic of the circuit needed for the SIP external counter. The components are: (a) 3.3 Voltage regulator, FAN2504 (b) Operational Amplifier (OPAM) LM324 (c) The SPI counter LS7366 (d) Crystal oscillator, CMACKD 20.00 (e) Capacitors and resistors . The connector at the bottom (U13) is the expansion pack in the mote.

Moreover, we need to modify the voltage output of the mote to provide the voltage and current to the motor, and provide the 5V to the sensors and counter board. Figure 3.4 shows the schematic of this circuit.

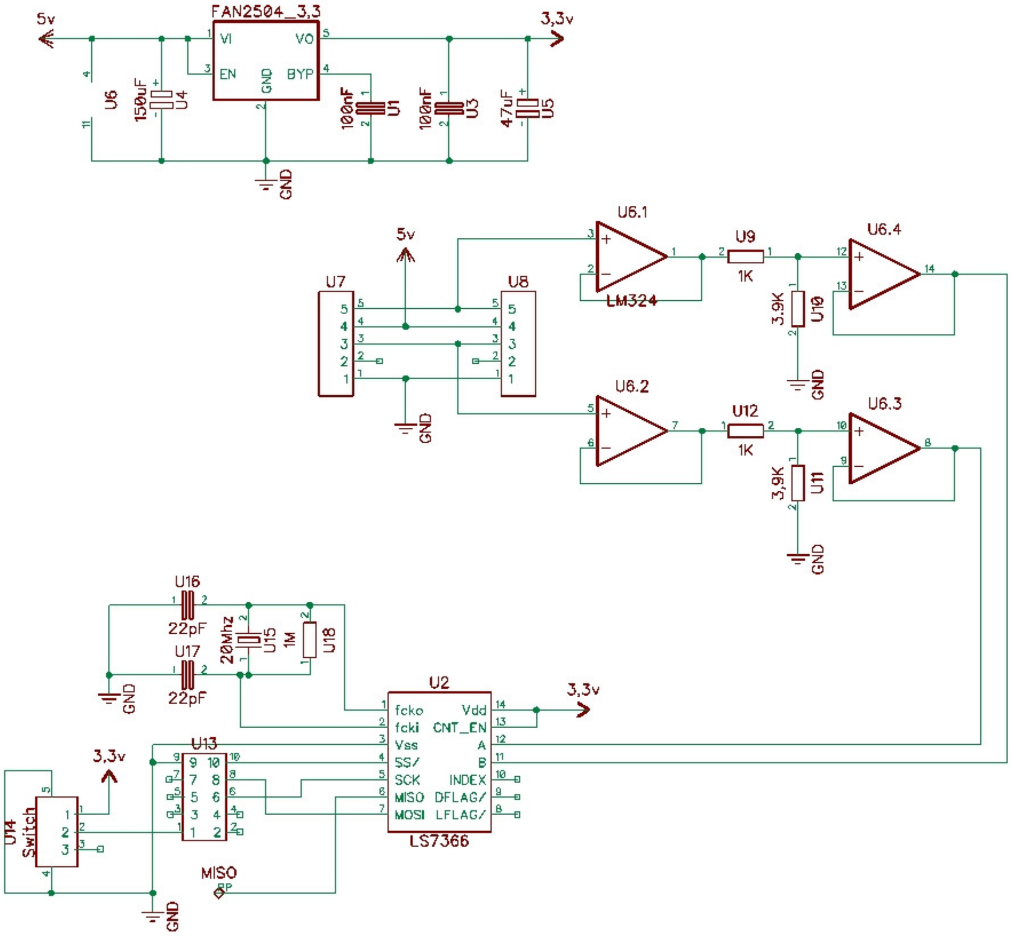

Figure 3.1: Schematic of the circuit

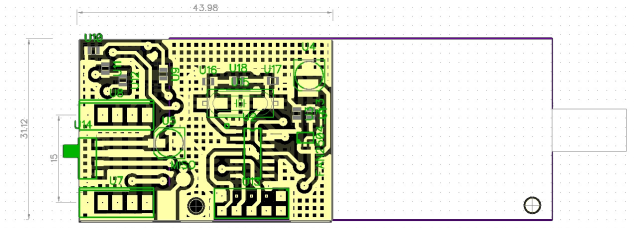

Figure 3.2: General layout with the board and the mote

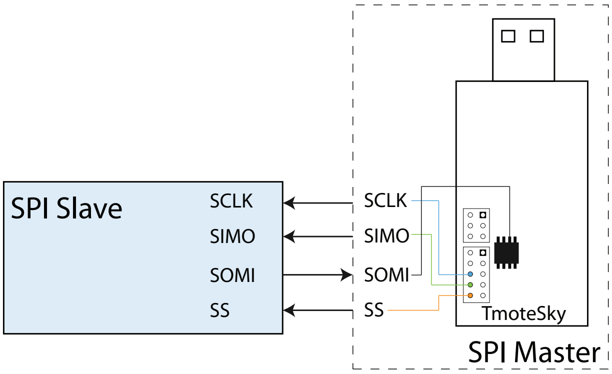

Figure 3.3: Connection between the counter board and the mote for the SPI interface.

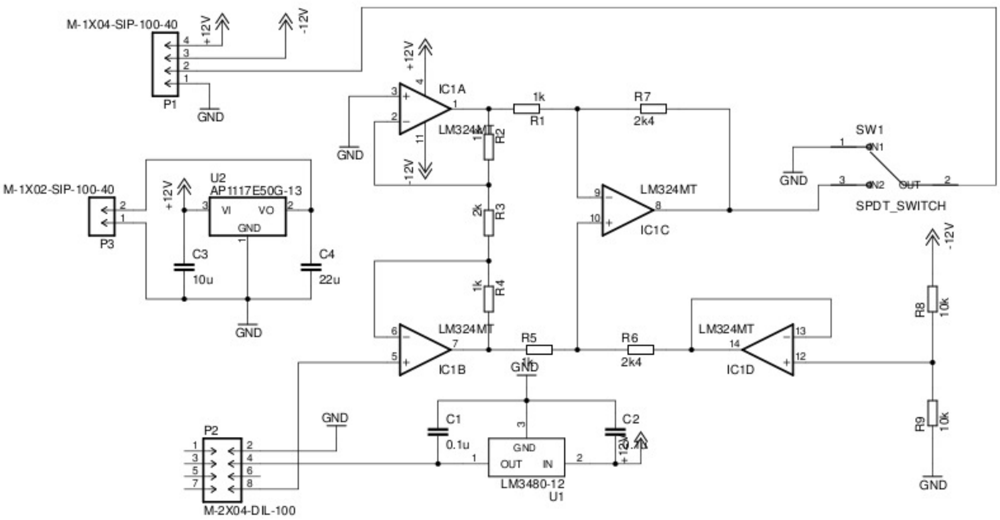

Figure 3.4: Schematic of the circuit

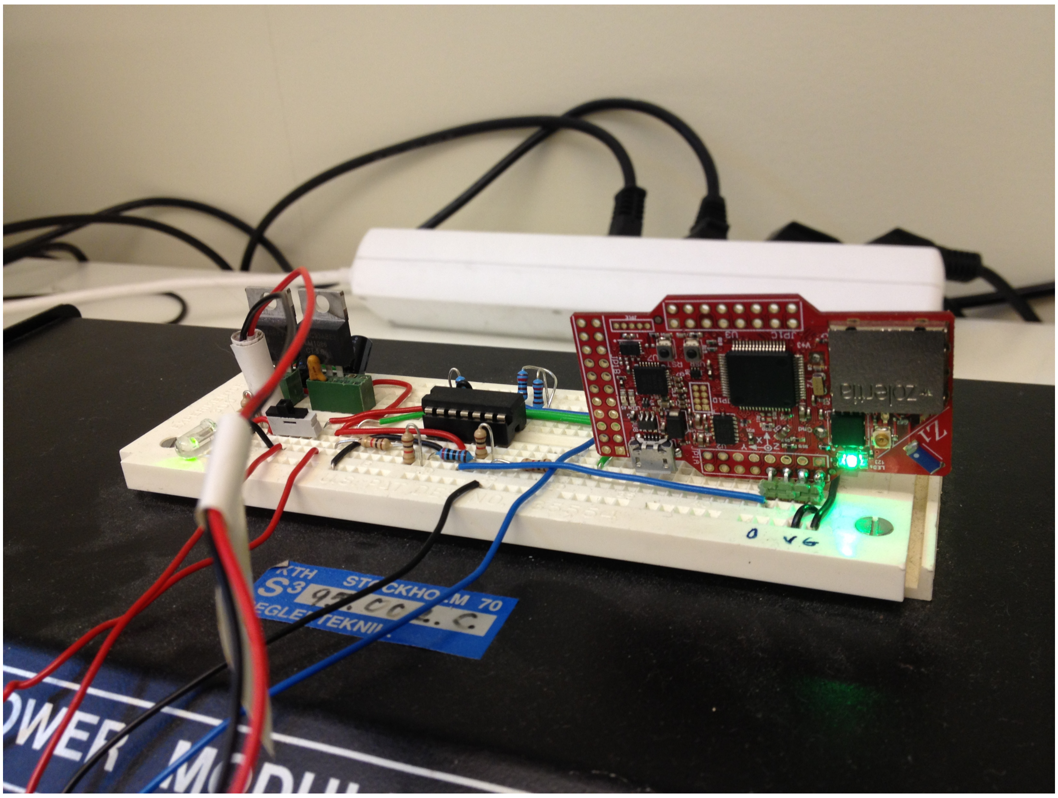

Figure 3.5: Mounted Circuit

After showing the different board and circuits, we explain which are the procedures to interconnect them. Below we have a list with the different steps:

1. Connect the counter board with the mote. There is one communication line that needs to be soldered manually. It is connected to the Flash memory in the mote. Figure 3.3 shows the connection.

2. Connect the encoders to the counter board with the white bus. Check that the ground is connected properly. It does not influence if we switch the cart position with the pendulum angle. We only need to be consistent when we program the motes.

3. Mount the circuit shown in Figure 3.4, in the prototype board that is placed in the UPM. Figure 3.5 and Figure 3.6 show the connections in prototype board.

4. Supply the circuit in the prototype board with the +12 V, GND and -12 V from the UPM. Before connecting the sensors or any mote, be sure that the

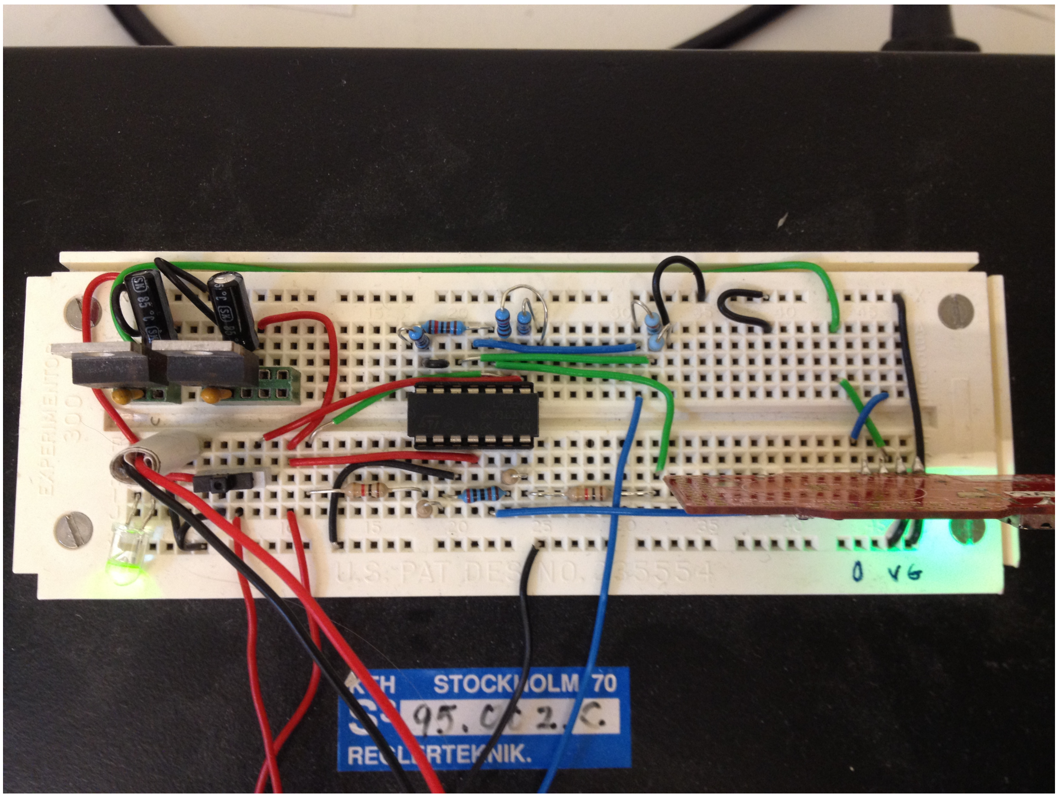

Figure 3.6: Mounted Circuit

voltages are correct: We have + 5V to supply the counter boards and 3.3V for the Actuator mote.

5. Connect the Actuator mote in the prototype board if the voltage is correct.

3.2 Sensors, Controller and Actuator

Figure 3.7 shows the scenario with four motes involved. There are two motes for sensing, one for the cart position and another for the pendulum angle. Another mote is the contoller, and finally the mote attached to the UPM which gives the voltage to the pendulum.

3.2.1 Installation

As we have seen, for this scenario we need to program two motes with the `EncLS7366RPerformance` application, one with the `ActuatorApp` and the Zolertia with the `Controller`.

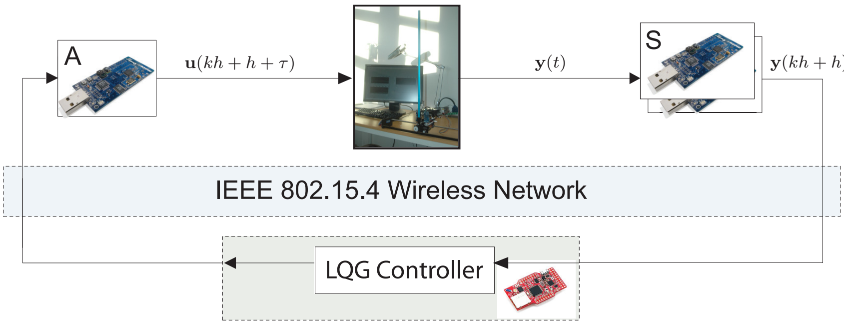

Figure 3.7: Inverted pendulum scenario with two links

Motes

The code for the motes is placed in the folder `actuation-included/nonbeacon-enabled/PerformanceApps`.

For the two links scenario we have the following applications:

Application	ID	Description
`2Links/Controller`	0	Controller "in your hand", the controller does not need to be placed in a certain place.
`EncLS7366RPerformance`	1	Sensor for X_c: cart position
`EncLS7366RPerformance`	2	Sensor for θ: pendulum angle
`2Links/ActuatorApp`	3	Mote connected to the UPM

3.3 Sensors and Actuator

Figure 3.8 shows the scenario where we have three motes involved. There are two motes for sensing, one for the cart position and another one for the pendulum angle. There is another mote acting as a contoller and actuator at the same time. It is the Zolertia mote, attached to the UPM which gives the voltage to the pendulum.

As we have seen, for this scenario we need to program two motes with the `EncLS7366RPerformance` application and the Zolertia with the `ControllerActuator`.

Motes

The code for the motes could be found in the folder `actuation-included/nonbeacon-enabled/PerformanceApps`.

For the one links scenario we have:

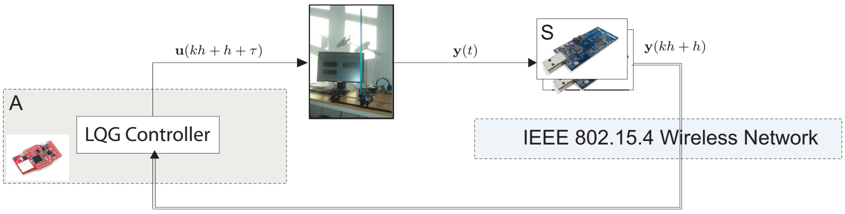

Figure 3.8: Inverted pendulum scenario with one link

Application	ID	Description
`ControllerActuator`	0	Controller and actuator mote connected to the UPM
`EncLS7366RPerformance`	1	Sensor for X_c: cart position
`EncLS7366RPerformance`	2	Sensor for θ: pendulum angle

Chapter 4

Results

To show the results we use another mote connect to the computer which works as a sniffer. The mote forwards all the messages with a certain payload length to the serial port.

Then, to show the results in the computer, we need to program the `PacketSnifferPerformance` application. To read the values from the mote we have two options implemented: Labview or Matlab/Simulink.

4.1 Labview

With the `PacketSnifferPerformance` application installed on the mote, we run the `sf` on the port where it is connected. Once the `sf` is running, it is time to run the LabView vi, provided in the folder `apps.inverted-pendulum/LabView`.

Figure 4.1 shows an screenshot of the LabView Vi which compute the performance evaluation. In the *Communication* tab, we see four charts that shows the reliability, delay, errors and inter sampling time for each of the links and the global results. In the *Control* tab, we can see the sensor values and the voltage applied to the motor, in case we ran the two links scneario.

4.2 Matlab/Simulink

Another option is Matlab with a Simulink model. This option is not that easier to modify and to show the results in a Graphical User Interface (GUI). It is more complicated and resource consuming.

Basically, with the `PacketSnifferPerformance` application installed on the mote, we run the `sf` on the port where is connected. Once the `sf` is running, we run the Matlab GUI provided in the folder `apps.inverted-pendulum/MatlabSimulink/ControlAnalysis`.

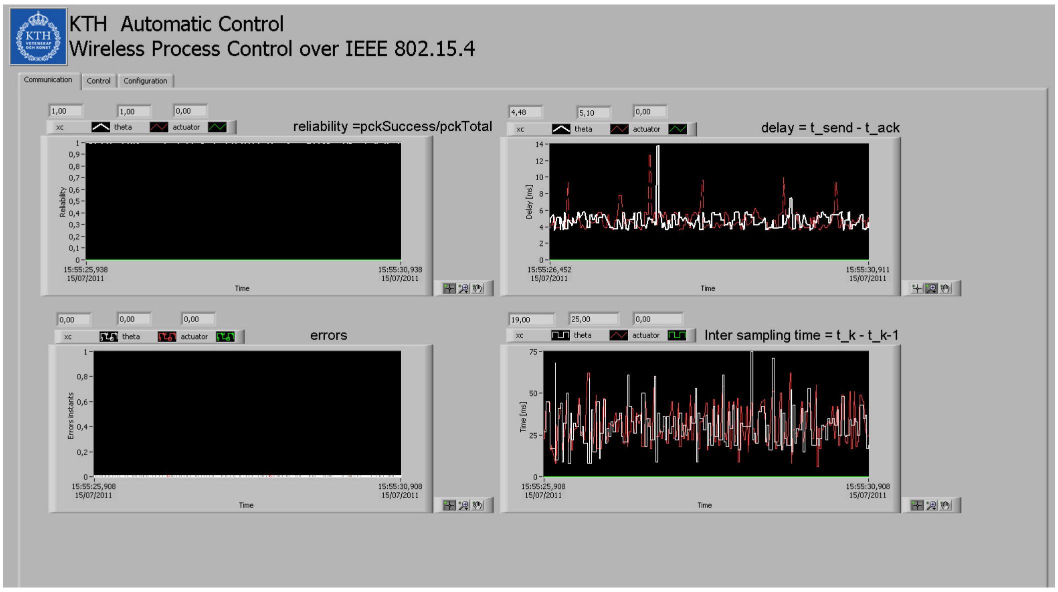

Figure 4.1: Screenshot of the LabView Vi running with the pendulum controller.

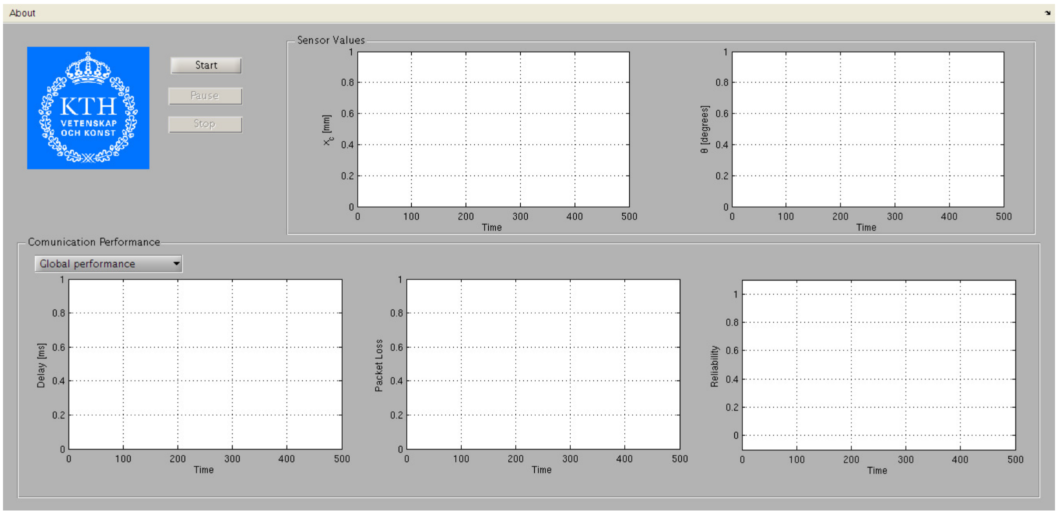

Figure 4.2: Screenshot of the GUI to show the inverted pendulum analysis.

References

[1] *Inverted Pendulum Control over an IEEE 802.15.4 Wireless Sensor and Actuator Network*, Germany, feb. 2011. European Wireless Sensor Networks (EWSN).

[2] Aitor Hernandez. Wireless Process Control using IEEE 802.15.4. Master's thesis, Royal Institute of Technology (KTH), November 2010. URL https://eeweb01.ee.kth.se/upload/publications/reports/2010/XR-EE-RT_2010_020.pdf.

[3] Aitor Hernandez. Communication between pc and motes in tinyos. Technical report, Royal Institute of Technology (KTH), July 2011.

[4] Aitor Hernandez. Getting started with tinyos at the automatic control lab. Technical report, Royal Institute of Technology (KTH), July 2011.

[5] Aitor Hernandez, Joao Faria, and Jose Araujo. Wireless water tanks. diferent scenarios and controllers. Technical report, Royal Institute of Technology (KTH), July 2011.

[6] Zolertia | Z1. URL http://www.zolertia.com/products/Z1.

[7] Quanser. *Single Inverted Pendulum - Instructor Manual*.

[8] Quanser. *Single Inverted Pendulum - Sudent handout*.

[9] Quanser. *Single Inverted Pendulum - User Manual*.

[10] Systems, LSI Computer. LS7366R - Quadrature counter with serial interface. Technical report. URL http://www.lsicsi.com/pdfs/Data_Sheets/LS7366R.pdf.