

## Chapter 3

# IMPLEMENTATION OF A MOBILE INVERTED PENDULUM

The balancing robot system is built as part of the thesis requirement for Kalman filter experimentation and to test the performance of linear state-space controllers in balancing an unstable system. Figure 3.1 shows the robot chassis designed using Solidworks.

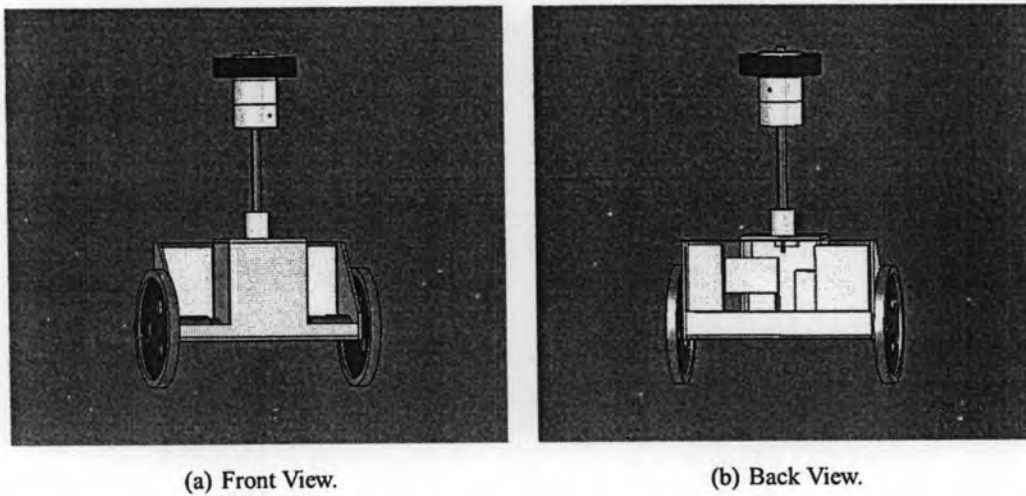


Figure 3.1: Mobile inverted pendulum body.

### 3.1 Robot Chassis

The robot chassis design is similar to Joe: mobile inverted pendulum. The mass of the body is 12.1 kg with 0.52 m in height and 0.45 m in width. The components (battery, monitoring battery, regulator, drive motor) are placed inside the body. The robot chassis is made from aluminium and wheel has been made from metal. The rubber on the top of body is used for protecting the body when the system becomes unstable. The robot wheel is made from metal and it is covered with rubber to increase friction. Radius of wheel is 0.093 m with a mass of 1.282 kg.

### 3.2 Controller

A DSPIC30F4011 is chosen as CPU (Figure 3.2). It is a micro-controller from Microchip and it is used as the brain of the balancing robot system. The controller consists of a powerful 16-bit micro-controller running at 120 MHz with 144 Kbyte ROM onboard. This allows relatively large and computing intensive programs to be executed quite easily. The programming languages used in this controller are C and Assembly.

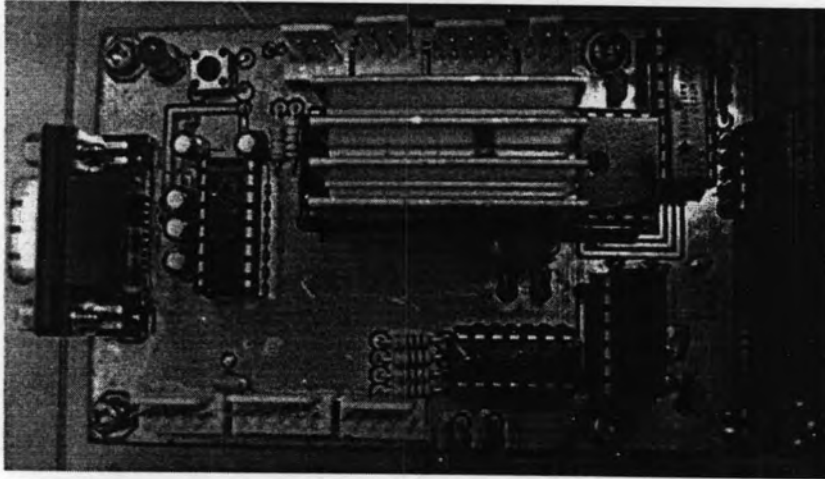
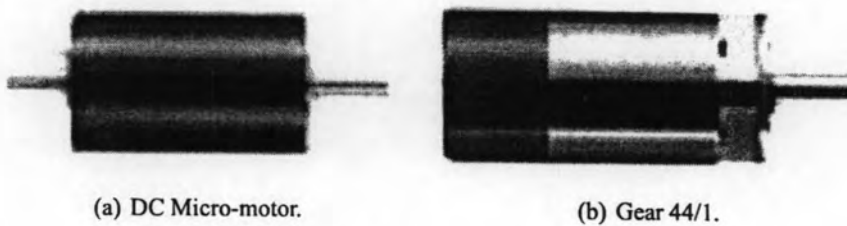


Figure 3.2: CPU DSPIC30F4011.

### 3.3 Actuators

The actuation to balance the robot is via two Faulhaber DC Micro-motors series 3863 made by Faulhaber (Figure 3.3), Germany. Each motor has a gear model 44/1 with reduction of 23:1 and a torque constant of 33.3 mN-m/A.



(a) DC Micro-motor.

(b) Gear 44/1.

Figure 3.3: DC Micro-motor and gear.

### 3.4 Sensors

A SILICON SENSING CRS03 (Figure 3.4 (a)) analog rate gyroscope and a ANALOG DEVICE ADXL202E (Figure 3.4 (b)) digital or analog accelerometer are installed on the balancing robot system to measure the tilt angle of the robot as well as the angular velocity. The

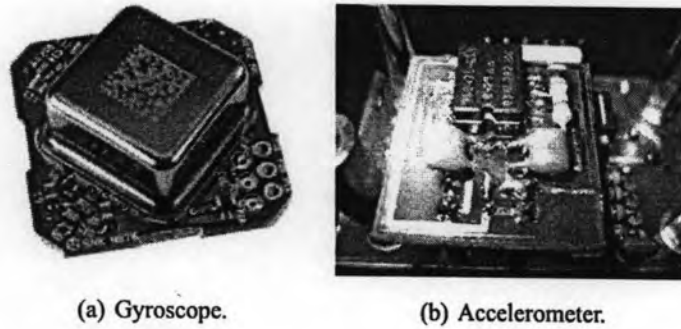


Figure 3.4: Sensor to measure angle and angular velocity of the MIP.

gyroscope returns the instantaneous angular velocity whose analog measurement is obtained through an A/D converter.

The accelerometer is a capacitive, liquid based sensor that generates a pulse width modulated output relating to the absolute inclination of the system. The output of the accelerometer is approximately linear with the measured angle. This sensor has an effective measuring range of only  $\pm 90$  degrees.

Besides the gyroscope and the accelerometer, the absolute encoders (Figure 3.5) built into the motors are also used to provide information on the robot's movement as well as the speed at which it is travelling. These encoders are capable of providing 1024 counts per one shaft revolution.



Figure 3.5: Encoders HEDM5500J.

### 3.5 Remote Control

To make an MIP move forward, backward, leftward or rightward, we need to use a remote control to set the reference signals. To build the remote control we use the radio frequency module from Radiometrix showed in Figure 3.6. It works on the 433Mhz frequency. The SILRX-433-5 integrates to a complete FM superhet UHF radio receiver on a small module. Together with the matching TXM-433-5 transmitter, a one-way radio data link can be achieved over a distance up to 200 meters on open ground. This remote control (Figure 3.7)

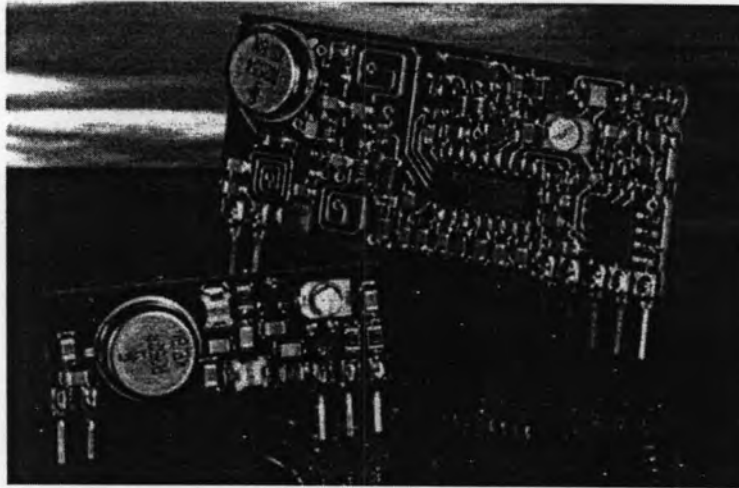


Figure 3.6: Left: TXM-433-5 transmitter Right: SILRX-433-5 receiver.

can make the MIP move forward, backward, leftward or rightward with a battery 12V.

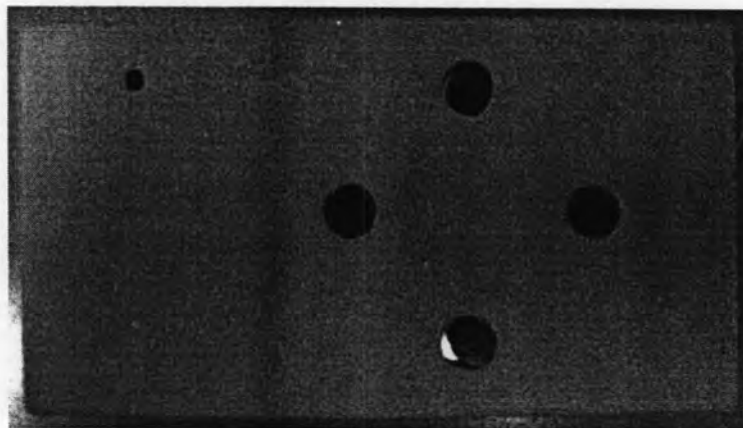


Figure 3.7: Remote control for the MIP.

### 3.6 Block Diagram for Implementation of the MIP

In this section, we show the block diagram of the concept for implementation of the MIP. This block diagram will help us to understand how to build the MIP with DSPIC micro-controller. The implementation of the MIP has been designed in away that it is simple to use.

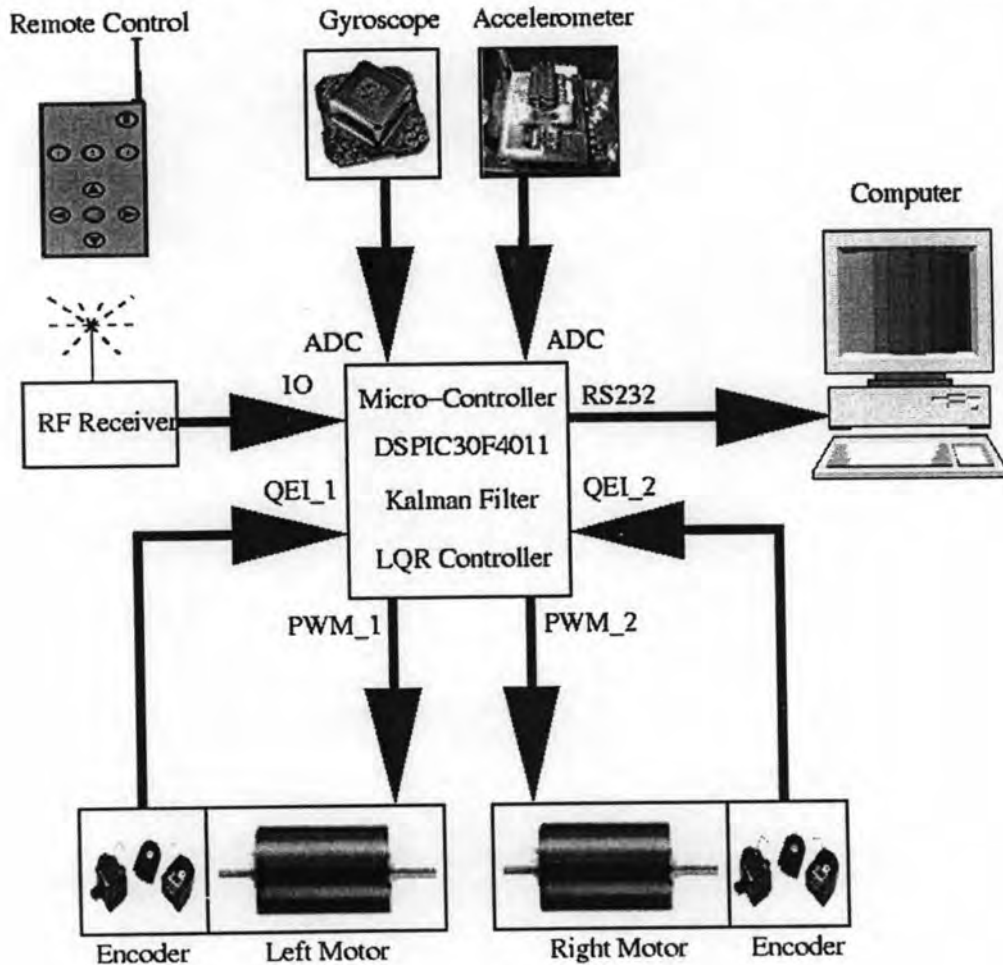


Figure 3.8: Block diagram for implementation the MIP.

It consists of a power-on switch mounted on the MIP.

When the MIP turned on, the Kalman filter calibrates the sensors around 2 seconds and turns-on the power drive for the motors. The control system will automatically balance the MIP in an upright position. The MIP can then be piloted by imposing a straight-line speed as well as a yaw rate on the proportional levers of the radio control unit. In case the pitch angle is higher than about 75 degree/s, the control system will automatically be switched off. The battery voltage is permanently monitored by the control system with 10 level display and an

LCD.

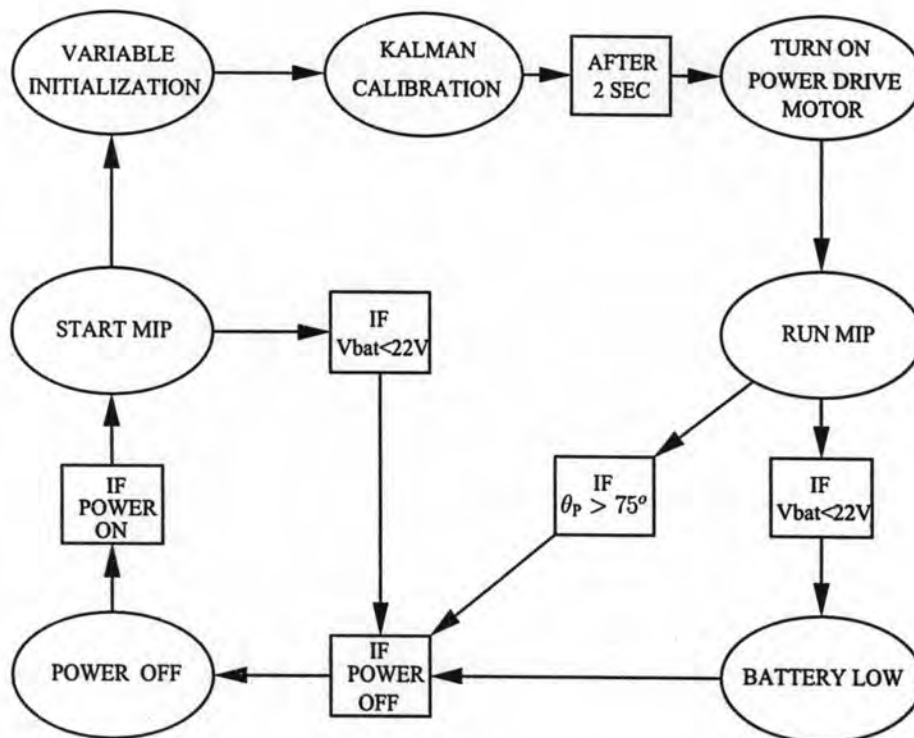


Figure 3.9: Flowchart of the control system's states.

## 3.7 Implementation of the Kalman Filter

### 3.7.1 Setting the servo test platform

The servo test platform (Figure 2.3) described in chapter 2 was used for Kalman filter experimentation, both sensors are rotated at angles between 0.5 radians. There are several tasks that need to be completed before the Kalman filter implementation is achieved. Firstly, raw readings of the gyroscope have to be scaled to rad/s and be on the same scale with the accelerometer readings. As unit conversion will cause inaccuracy in measurements, the raw accelerometer readings are differentiated and compared the gyroscope raw velocity readings to obtain the correct scale factor.

### 3.7.2 Setting the weighting matrices $Q$ and $R$

The Kalman filter is calibrated through the process covariance noise matrix  $Q$  and the measurement covariance noise matrix  $R$ . The forms of the matrices are given as:

$$Q = \begin{pmatrix} q_{acc} & 0 \\ 0 & q_{gyro} \end{pmatrix} \quad R = r_{measure}$$

Values of  $q_{acc}$  and  $q_{gyro}$  are set depending on how much faith one puts on the particular sensor. A lower value indicates the particular sensor is trusted more compared to the others. The value of the  $R$  matrix indicates the amount of noise expected from the measurement, a high value means the measurement is highly corrupted with noise. The calibration of the Kalman filter is usually by some complex statistical method which requires a lot of computation. In this thesis, the values are set by trial and error, a good understanding of the process is required to quickly determine the appropriate values for the  $Q$  and  $R$  matrix.

### 3.7.3 Algorithm for implementation of the Kalman filter

The Kalman filtering algorithm (Figure 3.10) is initialized as a background process running at every 10 milliseconds to facilitate multitasking of the DSPIC micro-controller so that the processor could perform other required routines.

## 3.8 Library of the MIP in SIMULINK

To use an MIP plant in simulation, a group of library in SIMULINK has been built. This allows the users to apply their controllers to the MIP plant.

Simulation is an important part the development of the MIP plant. The library consists of the following components:

- Nonlinear model MIP
- Linear model MIP
- MIP interface with XPC-Target
- MIP virtual reality

Figure 3.11 shows the library of the MIP.

## 3.9 Graphical User Interface

For simulations to be used more easily and more efficiently, it is important that a Graphical User Interface should be well functional. MATLAB has the power of handling large amounts

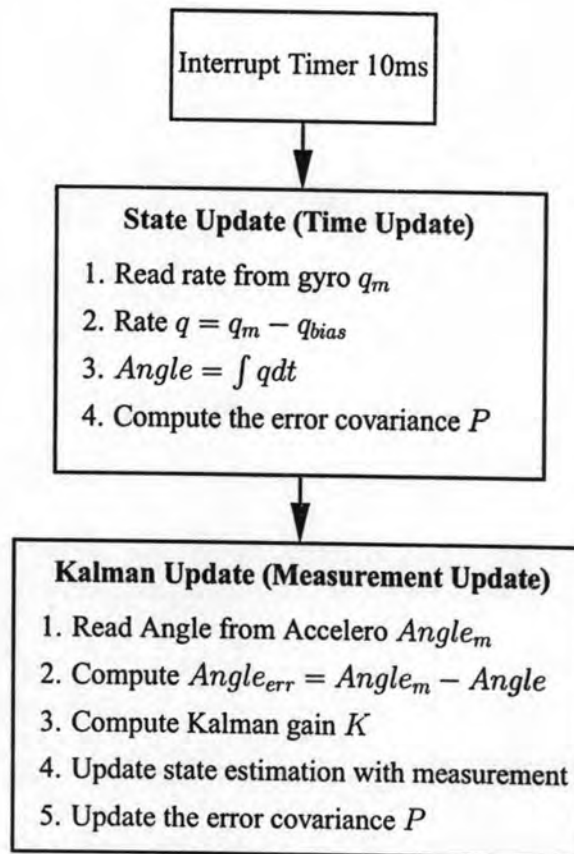


Figure 3.10: Kalman filter flowchart.

of data and performs necessary calculations and is therefore a good platform for a GUI.

The specifications for the most important content of the GUI are listed as follows:

- The GUI should handle all necessary preparations before the simulation. Different pendulum profiles, wheels, motors and gearbox are provided.
- Changes to the MIP specific parameters, e.g. mass of pendulum, moment of inertia, mass of the wheels, ... etc should be possible.
- By saving data and parameter values in specific files, the GUI should also be able to handle a list of separated data.

### 3.9.1 Ideas for the GUI

There are several demands on a simulation GUI. An investigation of the problems facing the GUI reveal the following ideas before starting working on the layout and implementing the code.



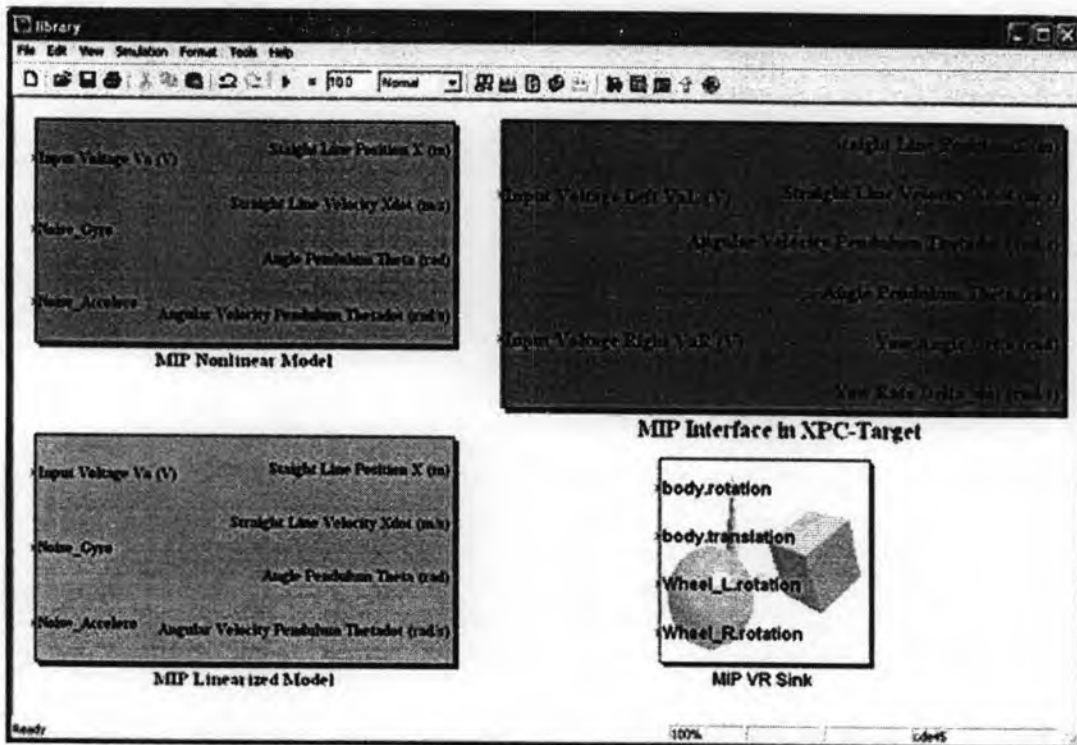


Figure 3.11: MATLAB library for the MIP.

- A fast operating GUI is preferred since slow loading of data and long time of simulation are irritating to the user.
- A nice visual appearance of the GUI is important. A boring and dull interface will not make the users eager to prepare the simulation properly.
- Easy preparation and efficient simulations are essential for a well functioning GUI. A regular users should be able to handle the necessary preparation before starting the simulation.
- Object-oriented code should be applied. This could be handy when the simulation is improved and other extensions to the program are needed. Object-oriented code can also be reused, which is efficient for the programmer.

### 3.9.2 GUI Design for the MIP

The Graphical User Interface for the MIP is shown in Figure 3.12. The GUI for the MIP has the advantages for the users as shown below:

1. Save configuration button  
It allows the users to save the variables of the MIP from a workspace by clicking on

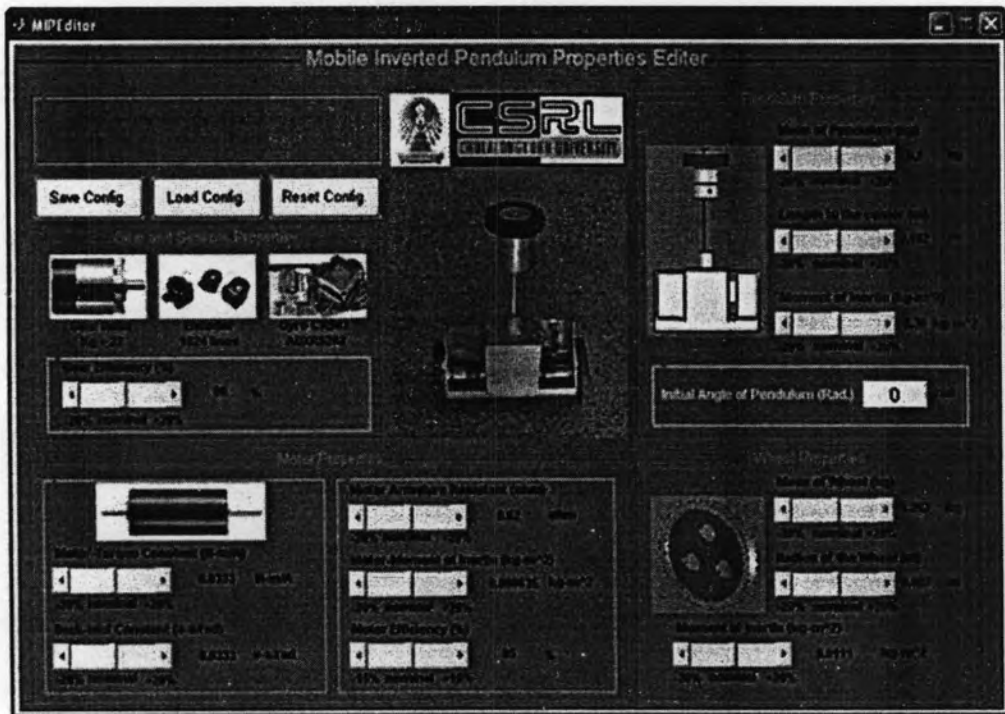


Figure 3.12: Graphical user interface for the MIP.

the button “Save Config.” using extension “.mat”.

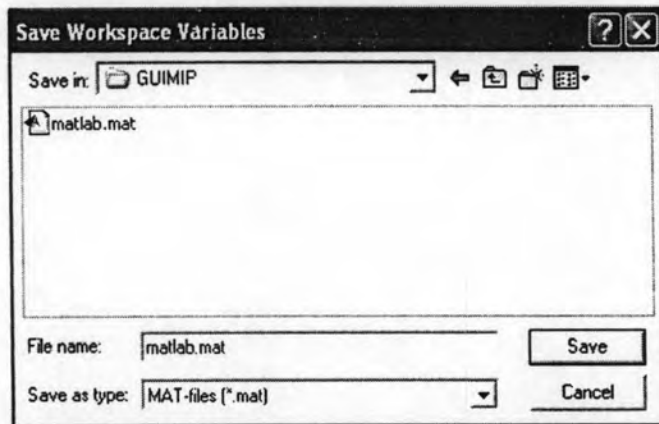


Figure 3.13: Save configuration in GUI for MIP.

## 2. Load configuration button

It allows the users to load their variables in the file extension “.mat” to the workspace.

## 3. Reset configuration button

Reset configuration is the command to load the default variables to the workspace. The

default variables are listed in Table 3.1.

Table 3.1: Default variables of the MIP.

Symbols	Values	Units
$g$ : Gravity constant	9.81	m/s <sup>2</sup>
$r$ : Radius of the wheels	0.093	m
$M_W$ : Mass of the wheels	1.282	kg
$M_P$ : Mass of the pendulum	9.6	kg
$J_W$ : Wheel moment of inertia	0.0111	kg-m <sup>2</sup>
$J_P$ : Pendulum moment of inertia respect to $X$ axis	0.36	kg-m <sup>2</sup>
$J_{P\delta}$ : Pendulum moment of inertia respect to $Z$ axis	0.563	kg-m <sup>2</sup>
$L$ : Length of center of the pendulum to cg	0.152	m
$D$ : Lateral distance between the contact patches of the wheels	0.35	m
$K_m$ : Motor-torque constant	0.0333	N-m/A
$K_e$ : Back-emf constant	0.0333	V-s/rad
$R_a$ : Motor armature resistance	0.62	$\Omega$
$\eta_m$ : Motor efficiency	85	%
$J_m$ : Motor moment of inertia	0.000635	kg-m <sup>2</sup>
$\eta_g$ : Gearbox efficiency	80	%
$K_g$ : Gear ratio	23	
$\theta(0)$ : Initial angle of pendulum	0	rad

#### 4. Gear efficiency slide bar

Gear efficiency slide bar is used for changing the variable of gear efficiency from 64% to 96%.

#### 5. Motor torque constant slide bar

Motor torque constant slide bar can be used for changing the variable of motor torque constant from 0.0266 N-m/A to 0.04 N-m/A.

#### 6. Back emf constant slide bar

Back emf constant slide bar is used for changing the variable of back emf constant from 0.0266 v-s/rad to 0.04 v-s/rad.

#### 7. Motor armature resistance slide bar

Motor armature resistance slide bar can be used for changing the variable of motor armature resistance from 0.496  $\Omega$  to 0.744  $\Omega$ .

8. Motor moment of inertia slide bar  
Motor moment of inertia slide bar is used for changing the variable of motor moment of inertia from 0.000508 kg-m<sup>2</sup> to 0.000762 kg-m<sup>2</sup>.
9. Motor efficiency slide bar  
Motor efficiency slide bar can be used for changing the variable of motor efficiency from 72.25% to 97.75%.
10. Mass of pendulum slide bar  
Mass of pendulum slide bar is used for changing the variable of mass of pendulum from 7.68 Kg to 11.52 Kg.
11. Length to center of CG slide bar  
Length to center of CG slide bar can be used for changing the variable of length to center of CG from 0.1216 m to 0.1824 m.
12. Moment of inertia of pendulum slide bar  
Moment of inertia of pendulum slide bar is used for changing the variable of moment of inertia of pendulum from 0.288 kg-m<sup>2</sup> to 0.432 kg-m<sup>2</sup>.
13. Mass of the wheel slide bar  
Mass of the wheel slide bar can be used for changing the variable of mass of the wheel from 1.0256 Kg to 1.5384 Kg.
14. Radius of the wheel slide bar  
Radius of the wheel slide bar is used for changing the variable of radius of the wheel from 0.0744 m to 0.1116 m.
15. Moment of inertia of the wheel slide bar  
Moment of inertia of pendulum slide bar can be used for changing the variable of moment of inertia of pendulum from 0.0089 kg-m<sup>2</sup> to 0.0133 kg-m<sup>2</sup>.
16. Initial angle of pendulum edit text  
Initial angle of pendulum edit text is used for changing the initial angle of the pendulum in radian.

### 3.10 MIP Interface with XPC-Target

This interface is used for connecting PC to the real MIP via serial port and National Instrument data acquisition card model PCI-6024E.

Figure 3.14 shows the MIP Interface with XPC-Target Block and Figure 3.15 shows the MIP



Interface with XPC-Target diagram.

With this block we can get all the states from the real MIP via serial port with baud rate



MIP Interface with XPC-Target

Figure 3.14: MIP interface with XPC-Target Block.

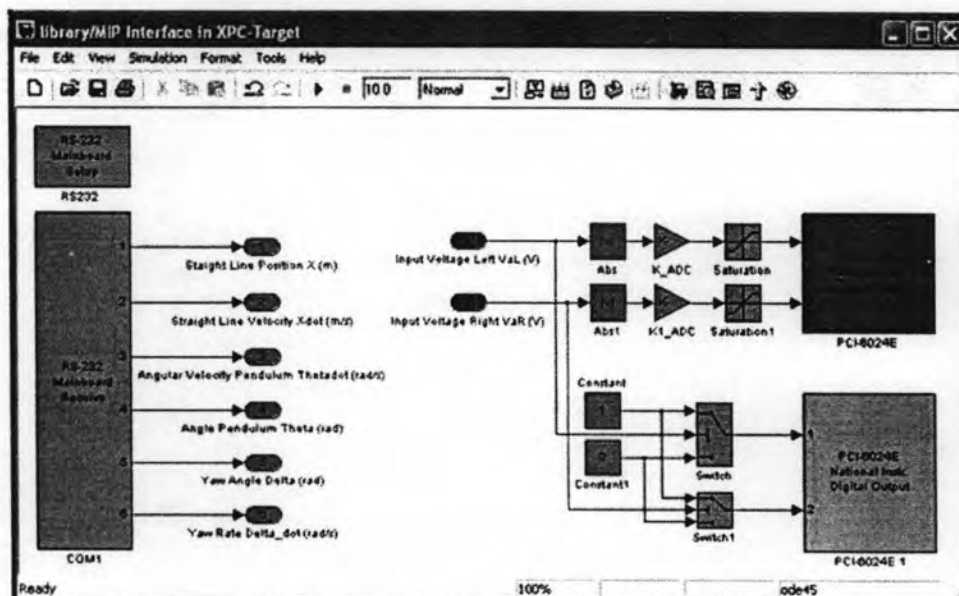


Figure 3.15: MIP interface with XPC-Target Diagram.

57,600 bits per second. We can also control the two motors via analog ports of PCI-6024E.

### 3.11 Virtual Reality

The Virtual Reality is a solution for interacting with virtual reality models of dynamic systems over time. It extends the capabilities of MATLAB and SIMULINK into the world of virtual reality graphics.

- Virtual worlds create virtual worlds or three-dimensional scenes using standard Virtual Reality Modeling Language (VRML) technology.
- Dynamic systems create and define dynamic systems with MATLAB and SIMULINK.
- Animation views moving three-dimensional scenes driven by signals from the SIMULINK environment.
- Manipulation changes the positions and properties of objects in a virtual world while running a simulation.

The Virtual Reality Toolbox provides a flexible MATLAB interface to virtual reality worlds. After creating MATLAB objects and associating them with a virtual world, we can control the virtual world by using functions and methods.

From MATLAB, we can set positions and properties of the VRML objects by creating callbacks from graphical user interfaces (GUIs), and mapping data to virtual objects. We can also view the world with the VRML viewer, determining its structure, and assigning new values to all available nodes and their fields.

In this thesis we create a virtual reality for the MIP that provides the users with animation accessibility in SIMULINK. Figure 3.16 shows the MIP virtual reality sink used in SIMULINK and Figure 3.17 shows the MIP virtual reality screen for animation. We have four following

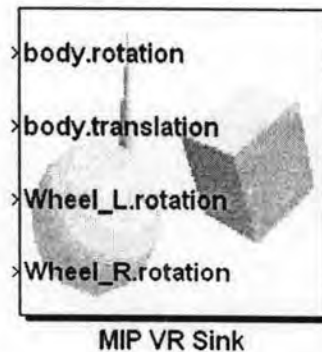


Figure 3.16: MIP virtual reality sink.

inputs for the MIP sink.

1. body.rotation is angle of pendulum ( $\theta_P$ )
2. body.translation is straight line position of the MIP ( $X$ )
3. wheel\_L.rotation is rotation of the wheel left
4. wheel\_R.rotation is rotation of the wheel right

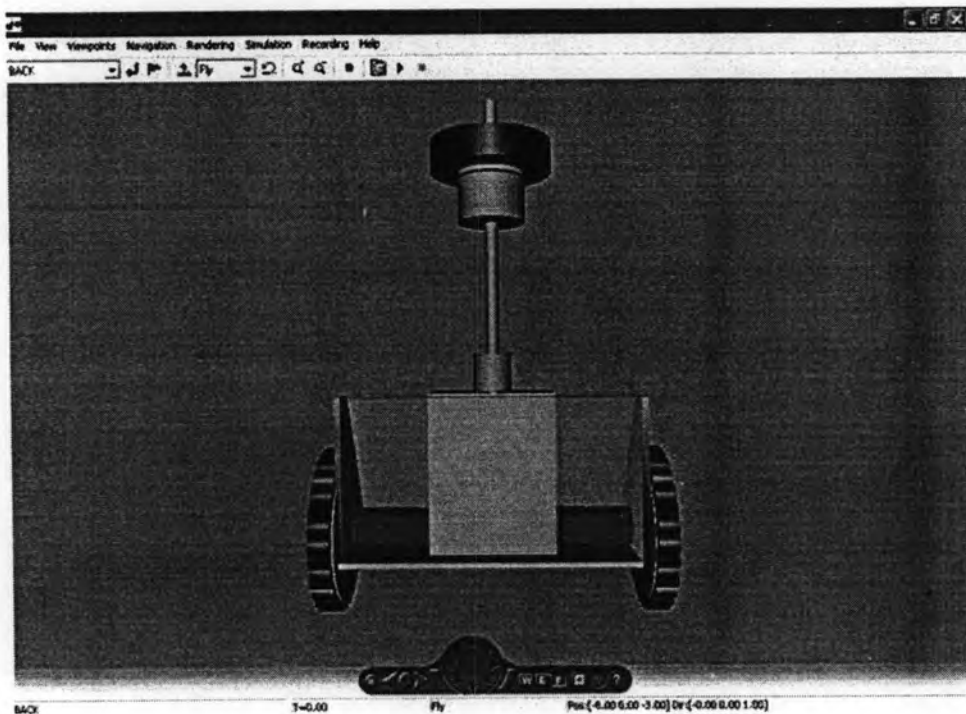


Figure 3.17: MIP virtual reality screen.

### 3.12 Conclusions

This chapter, the first part has presented the implementation of hardware that is used in the MIP. We have tried the MIP and have found that it can move forward, backward, leftward and rightward with an LQR controller that provides the best performance and robustness. With Radiometrix RF module, we can control the MIP more than 30m without losing the signal. The second part, we has created the MATLAB library such as nonlinear model, linearized model, graphical user interface that can be used in simulation, The MIP interface with XPC-Target that can be used to embed controller in MATLAB for balancing the real MIP plant via serial port and NI data acquisition card. Finally, we have built a virtual reality animation for the MIP plant that can be used with simulation and the real plant.