Chapter 5

REAL EXPERIMENTS

Those three experiments in simulation part then are applied into real experiment. We call the experiments as implementation, since all algorithms and programmings which used are similar. Users that already learn from experiments in simulation parts will be easier to follow the experiments in implementation part. By this procedure they also expected not to do something wrong or malicious which can harm the plants. Because certainly the safety of the plant should come first if we want to conduct a remote experiments. The only plant in our system which will be available for experiments is the flexible link plant.

5.1 System Analysis

Running the experiment from the host is enabled in default condition since this plant is already connected to the host computer. From the MATLAB environment we need to setup the experiment using file provided, that is `Set_SRV02_Exp5.m`. After that we have to compile the model in order to run the implementation. Firstly, build the code and the Wincon application will start automatically. The following figures is figure of Wincon application. As short explanation, it is actually support computer-to computer remote control. But in this case, the server and the client is in one computer, the host computer which directly connected to the plant.

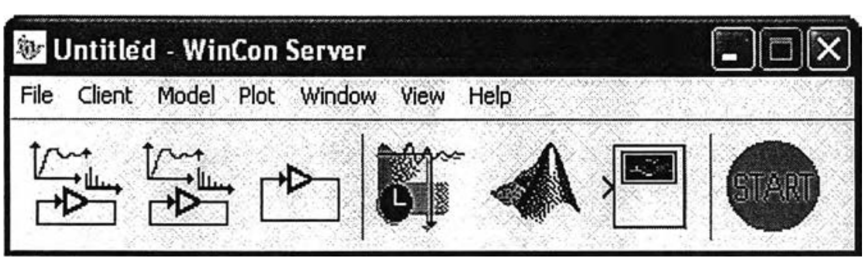

Figure 5.1: Wincon server.

A result of system analysis is shown in figure 5.3 and 5.4. This figure is taken from the scope of wincon client. The entered theta that is entered into the gain is 45^o, which means also as the expected value of flexible link amplitude.

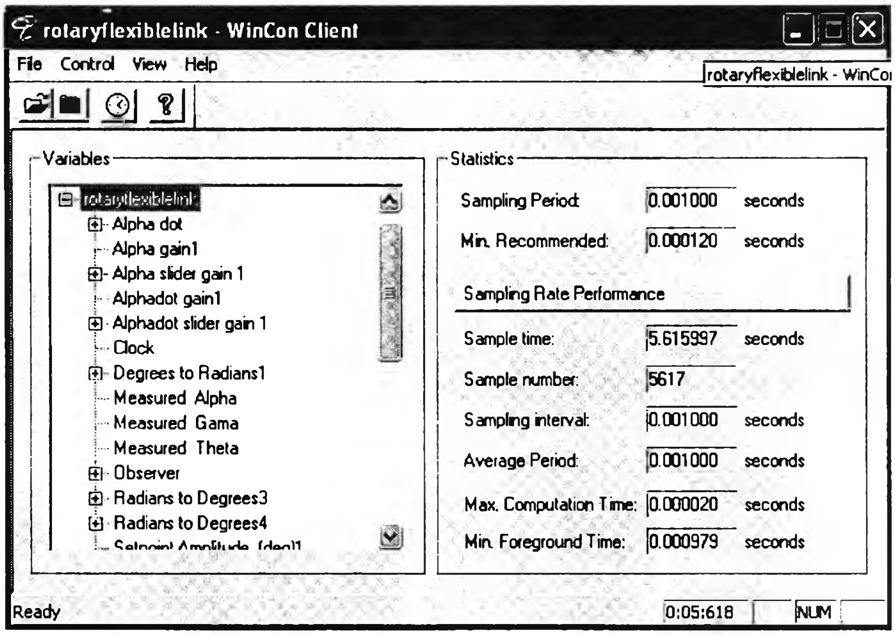

Figure 5.2: Wincon client.

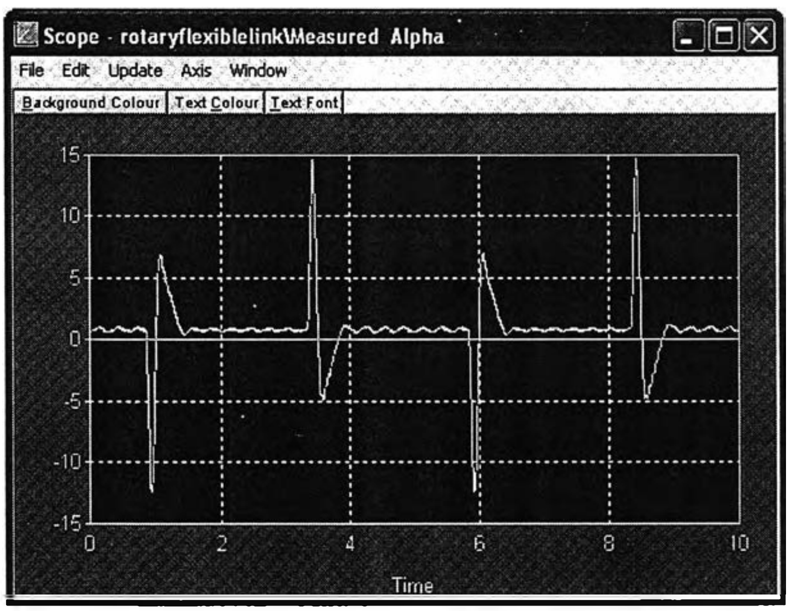

Figure 5.3: Experiment result of arm deflection, α.

5.2 Identification

To identify a system we need to have an open loop condition. This configuration in turn will impact another experiment, that is system analysis since it need a closed-loop sytem. In fact closed-loop system is the condition where most of the controller working on. Firstly,

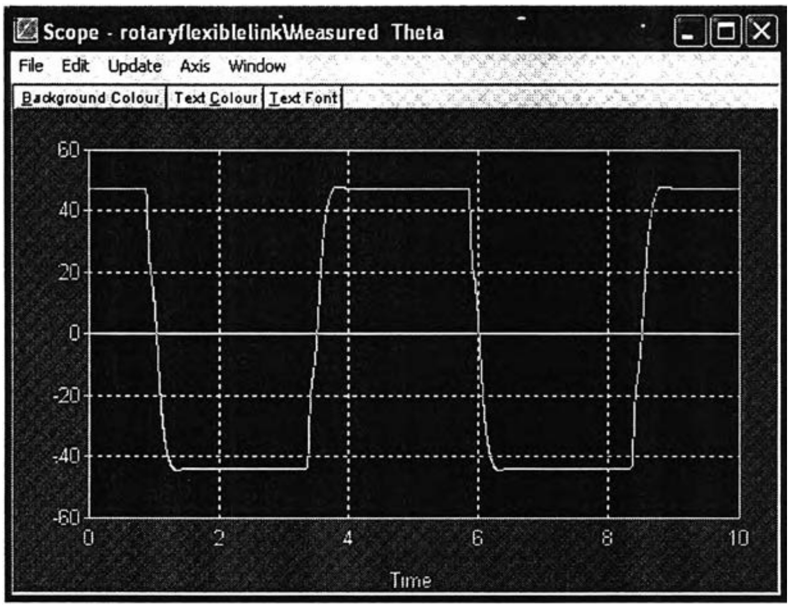

Figure 5.4: Experiment result of link position, θ.

we checked the step response of the plant. Just like the result from simulation, the real experiment show that any step input applied to the flexible link plant bring the plant into unstable position. So in this condition, we cannot use the step input to identify the flexible link plant.

Applying parametric identification, we got the result as shown in Table 5.1.

Table 5.1: Identification result using parametric method at flexible link plant.

Model type	Model
ARX	$y(t) = \frac{0.001699q^{-6}+0.001616q^{-7}}{1-1.275q^{-1}+0.2755q^{-2}}u(t) + \frac{1}{1-1.275q^{-1}+0.2755q^{-2}}e(t)$
State Space(n4sid)	$y(t) = \frac{0.0017s+0.0012}{s^3-1.9404s^2 0.9404s}$
State Space(pem)	$y(t) = \frac{0.0024}{s^3-1.9484s^2 0.9484s}$
ARMAX	$y(t) = \frac{0.0003239q^{-1}+0.001064q^{-2}-0.00216q^{-3}+0.0007867q^{-4}}{1-3.619q^{-1}+4.881q^{-2}-2.903q^{-3}+0.6413q^{-4}}u(t)$ $+\frac{1-2.957q^{-1}+3.238q^{-2}-1.585q^{-3}+0.3059q^{-4}}{1-3.619q^{-1}+4.881q^{-2}-2.903q^{-3}+0.6413q^{-4}}e(t)$

5.3 Controller Design

Controller design that are presented in this real experiments have the same basic controller and development. The main different here is that user cannot directly access the uploaded files they send. This fact majorly caused by the behaviour of the simulink model in experiment. Notice that this real experiment is using real time window target which need to be bulit first so that Matlab workspace can recognize the paramaters. Beside the actual requirement

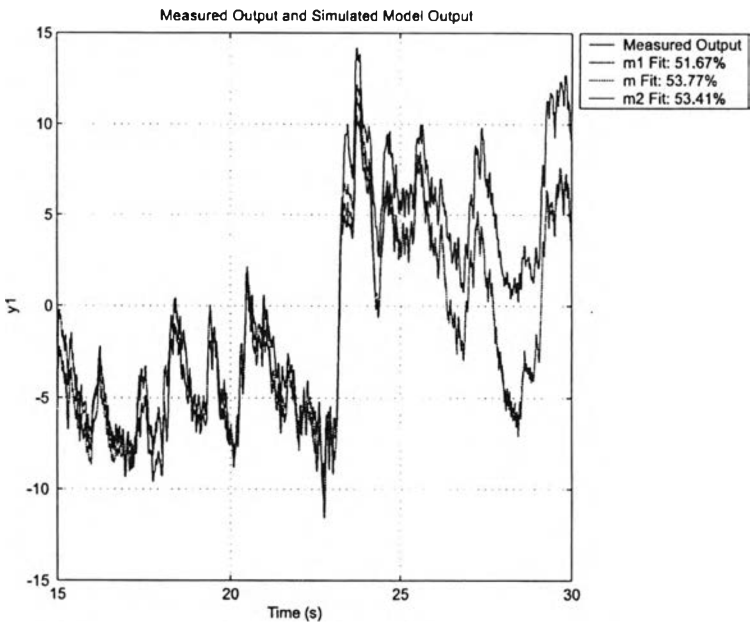

Figure 5.5: Comparison of the model type from identification.

that Matlab also need to compile this C-based code to support the real time environment. An interface that already prepared for this experiment is show by Fig. 5.6

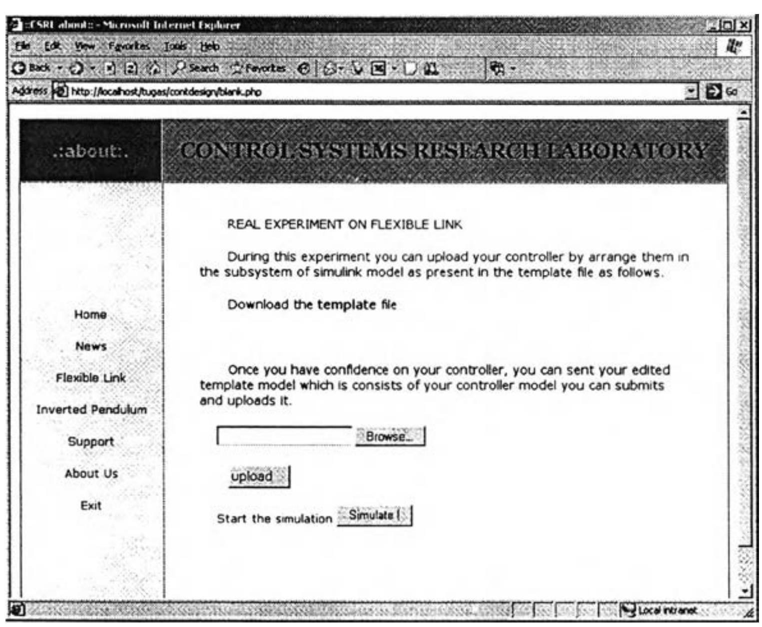

Figure 5.6: Real control design experiment page.

After doing experiment in simulation and real experiment, we found some facts that need to be underlined. Those facts are including the identification and the control design on the simulation base and the one in real experiment. For the identification, the experiment in simulation shows better fit in the validation. The figure of comparison in simulation as shown in Fig. 4.41 indicate the better result compare with the comparison on real experiment as shown in Fig. 5.2. It is can be understood since in the simulation we have less noise and disturbance which could lead the result model of identification differ away from the real one.

Notice Fig. 5.7 which come from the result of simulation using LQR controller. If we compare this result with Fig. 5.1 which is taken from the real experiment using default controller, we can see that these responses are different. Both figures are taken by applying a step function with magnitude 43^o degree as the reference input. We can see that although the same reference is applied, the simulation result has some overshoot. The overshoot happens because there are two complex eigenvalues which are underdamped. We suspect that the difference in responses occurs because of the difference behaviour between the obtained model and the real plant. It is commonly happened in the simulation as it is certainly hard to find a model that precisely model the real plant in every aspect.

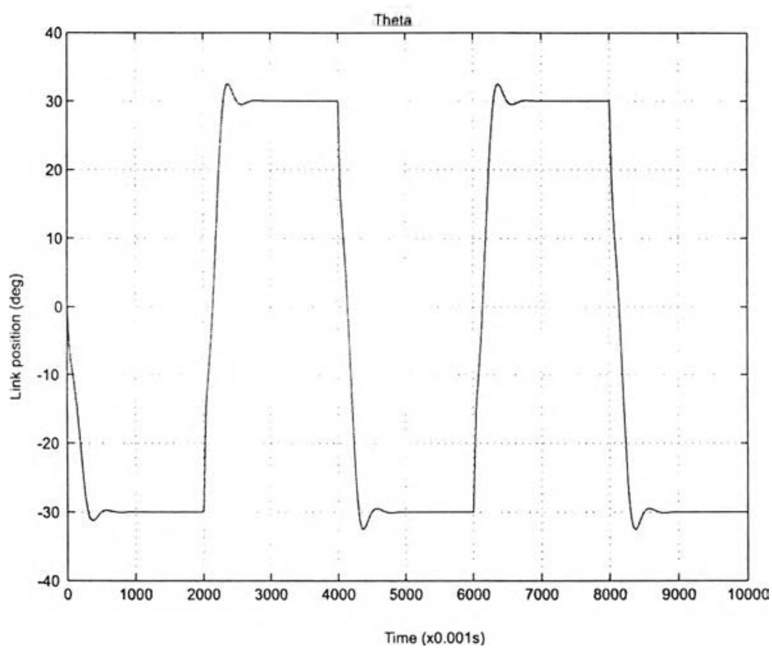

Figure 5.7: Simulation of link position, θ.

5.4 Problem and Obstacles

During the implementation, we found that the complexity of the designed hardware and software architecture dramatically increased. This fact especially occured when remote experiment through the Internet must be embedded in the system, compare with the simulation.

Some of the problems and obstacle that are faced when the process of the development being conducted can be described by the following explanation.

The development of the real experiment itself can be divided into three main parts. By the following item we explain what processes that actually being conduct for every step.

- Create the MATLAB code.
  If an application is required to be run remotely, then it also must work locally. This step is done by building the m-files. In fact this m-files can run the real experiment well, it can stop and run the application at the precise time.

- Develop the interface.
  This step is done by combining the interface that are used in the simulation experiment so that the remote experiment can interact with the plant.

- Publish to the Internet.
  Publishing the application to the Internet is meant that the user at the client computer can access and run the plant using the experiment interface. The inability of the connection between host computer of flexible link to the Linux server make it also imposible to publish the application. The Operating system that used in this configuration affect the server cannot access the resource, in this case is the directory of the host computer. This fact causes the directory that pointed by the `matweb.conf` cannot be accessed. In turn, it make the application for the real experiment could not work.

Some steps to resolve this problem already taken. One of them is by trying to upgrade the operating system of the host computer. The previous operating system installed in the host computer is Window XP Profesional without service package. This operating system is not ready to support for resource sharing in the network conection. It is proved by connecting a different computer with the same operating system and yields the same result, no shared directory can be accessed. Trying to increase the support of the operating system to the network connection, we upgraded it by installing the Service 2 Package. But this step also cannot work as expected.

Second step that had been taken was by reinstalling the plant on the different computer. This step facing obstacle of the license key for RTX installation. RTX ia a software that support MATLAB as well as Wincon to work in Real Time environment.