Chapter 4: Verifying the Simulator

Before the model predictive controller could be designed, the simulator first had to be verified to ensure that the simulator is a good representation of a real milling circuit. The first step was to enter the transfer functions of a typical milling circuit into the program (the parameters of the transfer function are given in Appendix D). The simulator program can, however, only accommodate first order transfer functions (with immediate response) or integrators. It was decided to use this simple structure for the simulator since the response, combined with the dynamics of the control valve in the input module and further dynamics in the output module, should give an adequate approximation for the second order response. The transfer function for the milling circuit contains two second order transfer functions. The first second order transfer function (between the *Density* and the *Feed water*) looks as follows:

$$Density = \left[\frac{0.03053}{1+166.4s} - \frac{0.02593}{1+232.8s} \right] e^{-90s} Feed\ \ water \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.1)$$

while the other second order transfer function (between the *Particle size* and the *Feed water*) looks as follows:

$$Size = \left[\frac{-4.906}{1+140.3s} + \frac{4.85}{1+203.7s} \right] e^{-130s} Feed\ \ water \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.2)$$

The responses of these two outputs to a step change in the *Feed water* are shown in Figure 4-1.

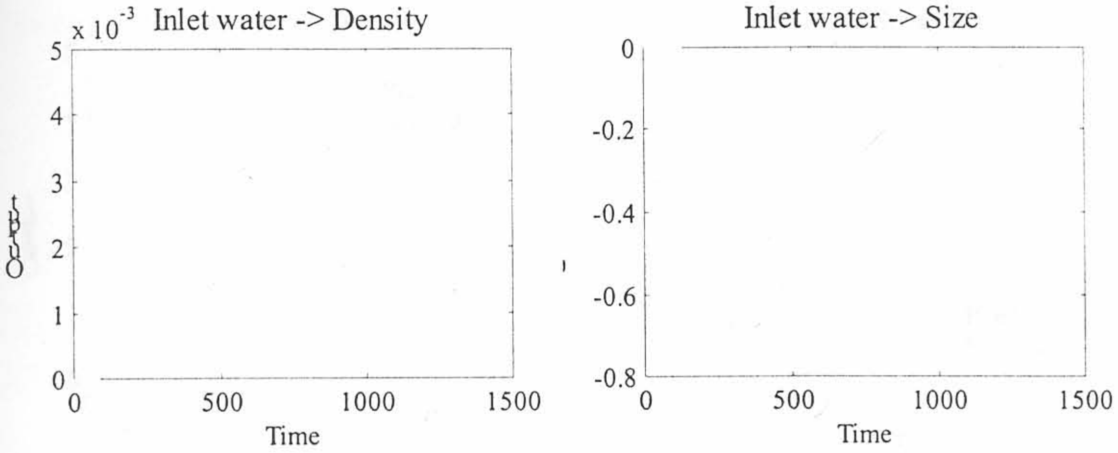

Figure 4-1: Second order plant step responses

The second order transfer functions were approximated as first order transfer functions with an initial response and an additional time delay. The response of this approximation will have a sharp edge at the initial response. This sharp edge will be rounded by the dynamics of the valve as well as by the output dynamics. This will make it look more like a second order response. The parameters of the first order initial response transfer functions were changed until the smallest error between the plant and the simulator was obtained. The least-squares error method was used as an error measure. The initial response transfer functions that gave the best approximations for the second order responses looked as follows:

$$Density = \left[\frac{0.001 + (0.005)(408)s}{1 + 408s}\right] e^{-252.6s} Feed\ \ water \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.3)$$

$$Size = \left[\frac{-0.056 - (0.8124)(350.2)s}{1 + 350.2s}\right] e^{-262s} Feed\ \ water \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.4)$$

The response of the simulator to a step change in the *Feed water* using these initial response first order transfer functions (with input and output dynamics) versus the second order response of the plant (also with input and output dynamics) is shown in Figure 4-2. From the figures it is clear that the approximations for the second order responses are acceptable for purposes of this investigation.

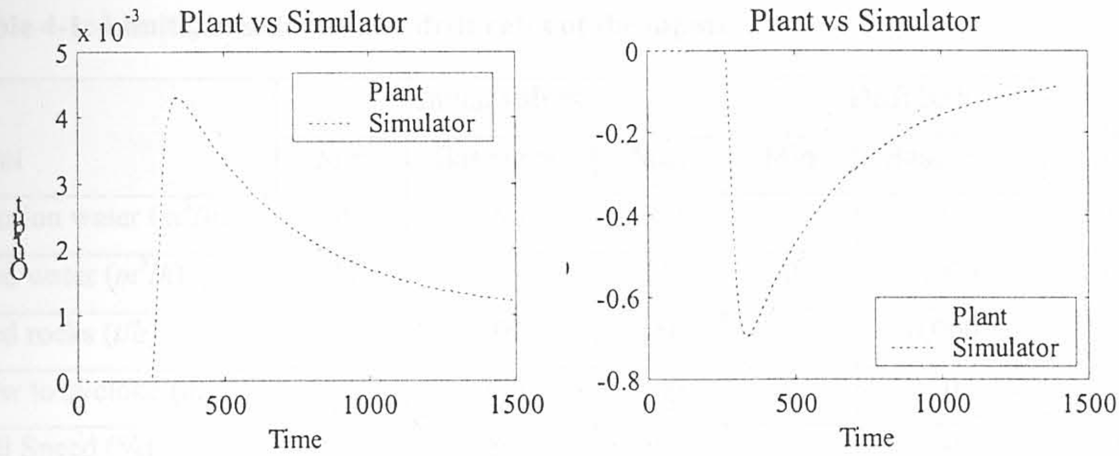

Figure 4-2: Step responses of the plant versus the simulator

After all the plant state dynamics were entered, the simulator was tested by applying a positive step change to each input variable separately. A time step of 1*s* was used for all the simulations and it was run for 10 000*s*. A time step of one second, which is much smaller than the shortest time constant, was chosen to ensure accuracy for the numerical integration technique. The simulation was run for 10 000*s*, which is also more than four times longer than the largest time constant, to ensure that the process will reach steady state. The results for these step changes on the inputs are shown in Appendix A. No noise, drift rates and other dynamics were entered at this stage so that the dynamics of the plant state module could be checked prior to the addition of these elements. The results showed that the simulator performed as expected, giving more or less the same response to step changes in the inputs as a real milling circuit would.

The parameters for the input module were entered next. The minimum and maximum value for each input variable as well as their base levels (steady state value) are shown in Table 4-1. The units of measurement for each input follow in brackets. The mill speed is given as a percentage of the critical speed. The critical speed is that turning speed of the mill that will cause total centrifuging. The drift rates of the minimum and maximum values as well as the base level are given as a change in the specific input value per second.

The magnitudes of the randomly generated noise and disturbances on the inputs are given in Table 4-2, as well as the time constants with which the random numbers are filtered.

Table 4-1: Limits, base levels and drift rates of the inputs

Input	Initial values			Drift Rates (s^{-1})		
	Min	Base level	Max	Min	Base level	Max
Dilution water (m^3/h)	0	150	300	0	0	0,01
Feed water (m^3/h)	0	7	15	0	0,0005	0
Feed rocks (t/h)	20	100	150	0	0,0003	0
Flow to cyclone (m^3/h)	200	300	400	0	0	0
Mill Speed (%)	60	80	95	0	0	0

Table 4-2: Noise and disturbance parameters for the inputs

Input	Magnitude of noise	Noise filter time constant (s)	Magnitude of disturbance	Disturbance filter time constant (s)
Dilution water (m^3/h)	2	30	1	30
Feed water (m^3/h)	0,1	30	0,05	30
Feed rocks (t/h)	10	50	5	50
Flow to cyclone (m^3/h)	10	20	5	20
Mill Speed (%)	0,2	30	0,1	30

The dynamics of each actuator on the inputs are given in Table 4-3. Only the valve for the *Dilution water* input was given a nonlinear factor to study the effect of a non-linear valve. The other valves were assumed to be linear.

Table 4-3: Actuator parameters for inputs

Input	Actuator time constant (s)	Actuator dead time (s)	Nonlinear factor	Slack
Dilution water	5	0	0,3	0,05
Feed water	12	0	0	0
Feed rocks	10	20	0	0,05
Flow to cyclone	8	0	0	0,03
Mill Speed	10	5	0	0

The dynamics of the actuator were used to design the local PI-controllers on each input. The parameters of the local PI-controller for each input were obtained. A minimum deadtime of 1s was assumed for design purposes. This is because a very small deadtime (near zero) will give rise to very large proportional actions. Large proportional actions will cause the noise to be amplified too much, resulting in an unstable controller. The parameters for each PI-controller are given in Table 4-4.

Table 4-4: PI-Controller parameters for the inputs

Input	Proportional action (K_c)	Integral action
Dilution water	4,58	2,35
Feed water	10,88	2,84
Feed rocks	0,533	14,69
Flow to cyclone	7,28	2,64
Mill Speed	1,88	8,3

No parameters or limits were entered for the state module. The limits, base levels and drift rates for the output module are given in Table 4-5. The sump level and mill load are measured as a percentage of the maximum level. The particle size is measured as the percentage of particles below 75 μm in diameter.

Table 4-5: Limits, base levels and drift rates of the outputs

	Initial values			Drift Rates		
Output	Min	Base level	Max	Min	Base level	Max
Sump level (% of max.)	0	60	100	0	0	0
Mill load (% of max.)	50	70	90	0	0,001	0
Mill power (MW)	1	1,9	2	0	0,00001	0
Density (t/m^3)	1,2	1,4	1,6	0	0	0
Particle size (% < 75μm)	60	75	90	0	0	0

The magnitude and noise filtering details for each output variable as well as the additional output dynamics are given in Table 4-6.

Table 4-6: Noise parameters for the outputs

Output	Magnitude of noise	Noise filter time constant (*s*)	Output gain	Output time constant (*s*)	Output dead time (*s*)
Sump level	5	30	1	5	0
Mill load	1	100	1	5	0
Mill power	0,05	30	1	20	0
Density	0,1	200	1	20	0
Particle size	5	20	1	15	5

After entering all these parameters, the simulator was run again for 10 000*s*. This time no step changes were applied. This was done to show the effect of the noise and the drift rates on the variables. The effect of the noise on an input (*Flow to cyclone*) is shown in Figure 4-3, where the actual input to the plant is plotted. The setpoint of the local PI-controller for this variable was 300 m^3/h. The variable is not kept exactly at the setpoint due to the random disturbance that is added after the PI-control action.

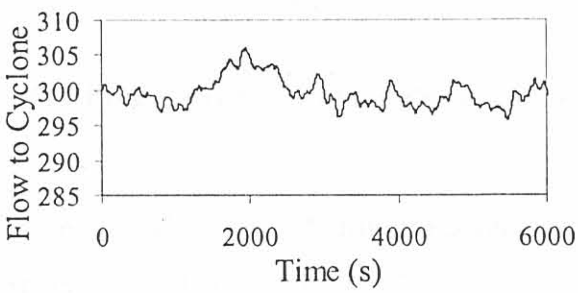

Figure 4-3: Effect of noise on an input (*Flow to cyclone*)

The effect of the local PI-controller is best shown when a nonlinear valve is used (for example the *Dilution water*). Due to the nonlinearity of the valve, the real input to the plant will not be the same as the input specified to the valve. The PI-controller will then compensate for this difference. The effect is clearly shown in Figure 4-4. The setpoint of the input is 150 m^3/h. Without the PI-controller, the input to the plant is much higher than expected, due to the nonlinearity of the valve. The input also drifted down due to the drift rate of the minimum value. When the PI-controller is used, the input is kept close to the setpoint of 150 m^3/h.

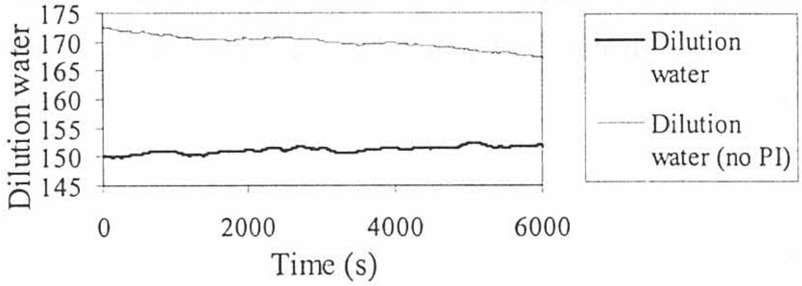

Figure 4-4: Effect of PI-controller on the input (*Dilution water*)

The effect of drift rate on the base level of an output (*Mill load*) is shown in Figure 4-5.

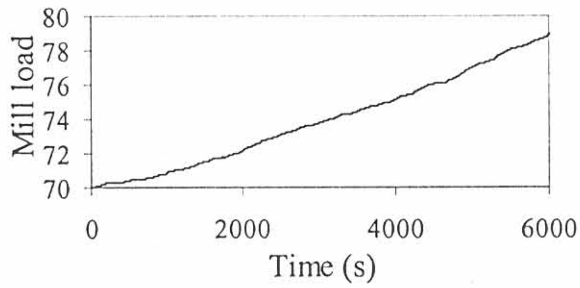

Figure 4-5: Effect of drift rate of base level on an output

The effects of random noise, disturbances and drift rates on the other input and output variables are shown in Appendix A. The variables will continue to change or drift around until they reach a soft or a hard limit. An example of this is shown in Figure 4-6 where the *Mill power* reaches its hard maximum of 2 after 10 000*s*. From 10 000*s* onwards the *Mill power* remains at 2 and will not drift away any further.

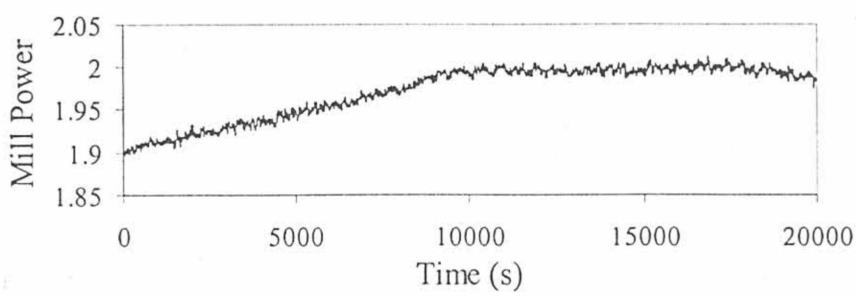

Figure 4-6: *Mill power* reaches limit at a maximum of 2

From these results it is clear that the simulator works and performs as required. The next step is now to design the model predictive controller using this simulator.