University of Bristol

Faculty of Engineering

Sensors, Signals and Control

Part 2: Elevation Axis– Theory and Simulation with Control Requirements

Supervisors:
Andres Marcos

Email:
Andres.Marcos@bristol.ac.uk

Authors:
Alex Charles
Akash Ramineni

Candidate Number:
(ac13625)
(ar14120)

7th April 2017

1 P-I-D analysis

This report introduces the effects of proportional, derivative and integral gains, implemented by a conventional PID controller on a Quanser's behaviour [a]. This is assessed using time-domain and root locus results. A closed loop PID controller, is used to reduce tracking error (e) between the control input and the observed output [1]. Through correctly tuning the PID controller, it is desirable to achieve a sharp response to user input.

1.1 Proportional Feedback Controller

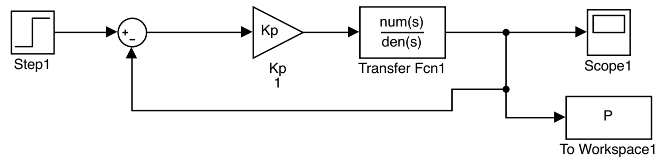

Figure 1: Proportional Feedback Controller

Proportional gain control uses the *Present* state of a plant to error correct. The correction applied is proportional to the deviation from the desired state of the plant. The result causes a highly oscillatory system response, as the present error evaluates the instantaneous state of the plant only. The simulation shown in Figure 1 found the effect of varying proportional gain (K_p) for the Quanser transfer function. The simulation executed an iterative loop with varying K_p from 0 to 0.1 in increments of 0.01. Figure 2 shows the time domain response of increasing K_p. When $K_p = 0$, a flat line was observed as there is no signal passing through the plant. By extending proportional gain, the correction factor became larger, increasing amplitude for higher K_p gains. The `stepinfo()` function revealed that an increase in K_p, reduced both rise time and steady-state error while raising overshoot.

The root locus plot in 3 allows observation of s-domain features with varying proportional gain. For pure variation in proportional gain, the (stable) poles are restricted to the same distance from the imaginary axis. By increasing K_p the poles moved away from the real axis in both directions, increasing natural-frequency (ω_n) and decreasing damping ratio ζ. The result verifies the observation noted in the time domain response, as increased natural frequency will lead to greater oscillatory behaviour, and reduced damping ratio increases the overshoot.

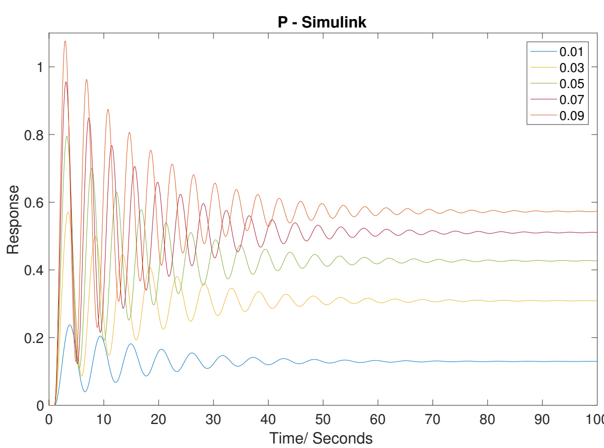

Figure 2: Response of Varying K_p

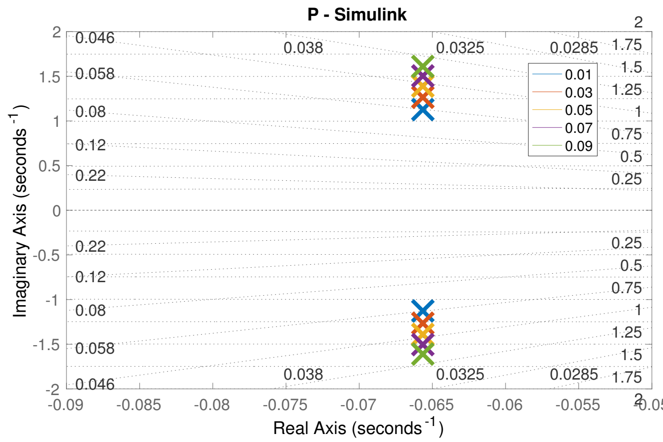

Figure 3: Poles of Varying K_p

1.2 Derivative Feedback Controller

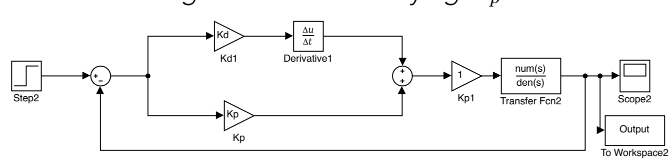

Figure 4: PD Feedback Controller

Derivative control utilises the expected *Future* state of the system to pre-emptively correct the error signal, e. This correction is proportional to the rate of change of the system state at a point in time. The system calculates the state of the plant in the future based on the rate of change conditions

[a] See Control Coursework Part 1 for a derivation of the transfer function used

and accounting for the effects in the response. For this reason, a large damping effect is expected for increasing derivative gain (K_d). The simulation in Figure 4 was implemented to observe the effect of purely varying derivative gain with constant proportional gain, and the effect of varying both gains by the same amount; controlled using the additional gain block after the controller. Gain values in Figure 5 and Figure 6 were incremented from 0 to 1 in increments of 0.1.

Both time domain responses show that increasing K_d has a greater damping effect on oscillations, reducing overshoot while increasing rise time. For a $K_d = 0$, large oscillations were observed, decreasing with higher gain values. For gain values 0.6 and above, both high damping and quick settling times were observed. Increasing both K_d and K_p by the same amount shows a reduction in rise time, overshoot and steady-state error. The root locus S-domain metrics in Figure 7 show that as K_d increases the poles are shifted negatively in the real axis. The semi-circular shape represents the line of equal proportional gain, and the points represent varying K_d at that K_p level. As K_d is increased, the polar angle also increases. This implies poles closer to the real axis, (higher K_d) correspond to poles with greater damping; the natural frequency is therefore reduced decreasing oscillatory behaviour.

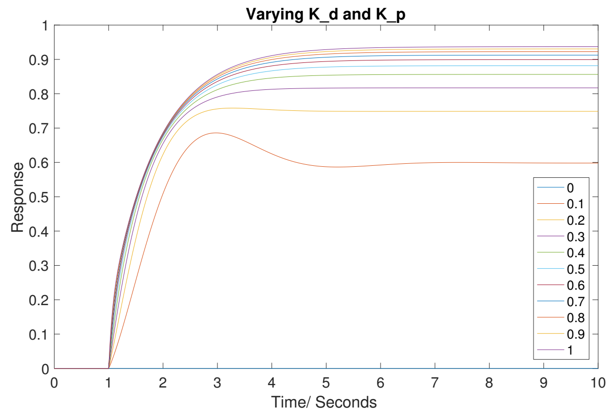

Figure 5: Time-Domain Response For Varying K_p and Varying K_d

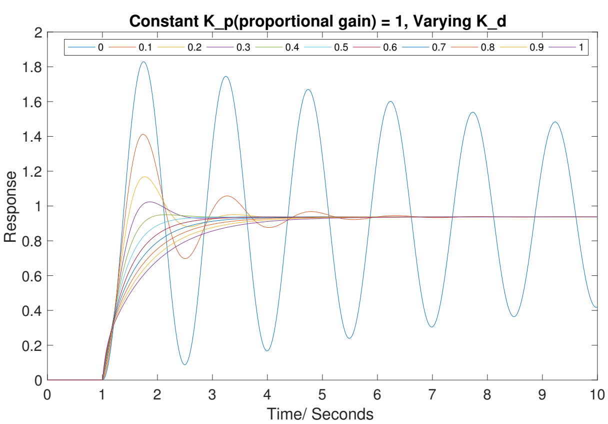

Figure 6: Time-Domain Response For Constant K_p and Varying K_d

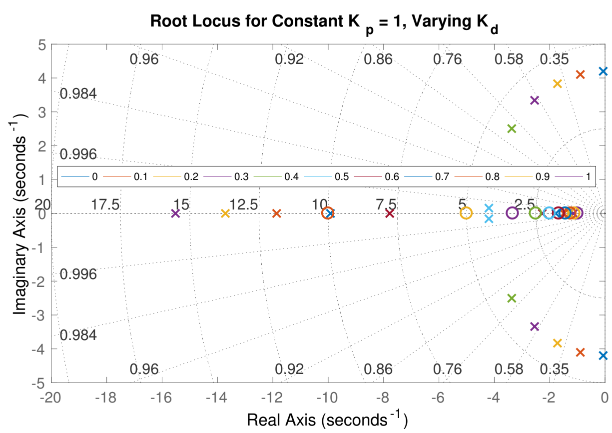

Figure 7: Poles of Constant K_p and Varying K_d

1.3 Integral Action

Integral control utilises the *Past* state of the system to produce an error-correcting signal. The integral controller sums the historical tracking error up to current time state, eliminating offset and leading to zero steady state error. The simulation of a full PID controller is shown in Figure 8. Figure 9 shows the variation of pure K_i with constant K_p and K_d. Figure 11 shows the variation of all three gains K_d, K_i and K_p by the same magnitude. This was implemented using a gain-scaling block after the controller. The time domain plots confirm the zero steady state error as the oscillations level to desired response level.

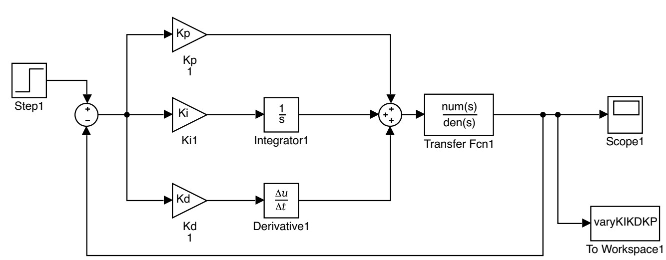

Figure 8: PID Feedback Controller

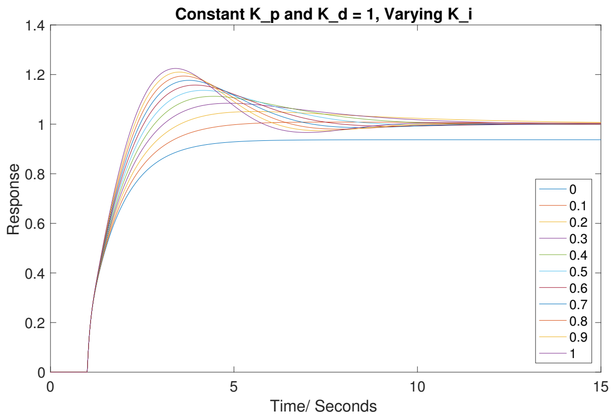

Figure 9: Time-Domain Response For Constant K_p, K_d and Varying K_i

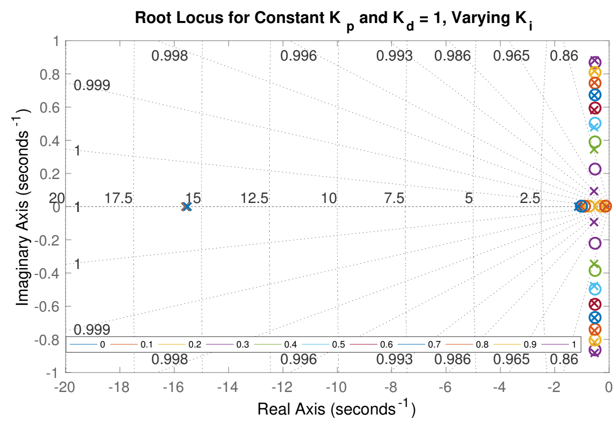

Figure 10: Poles of Constant K_p, K_d and Varying K_i

At $K_i = 0$ there is steady state error that exists in the system, but as integral gain increases the steady state error is eliminated.

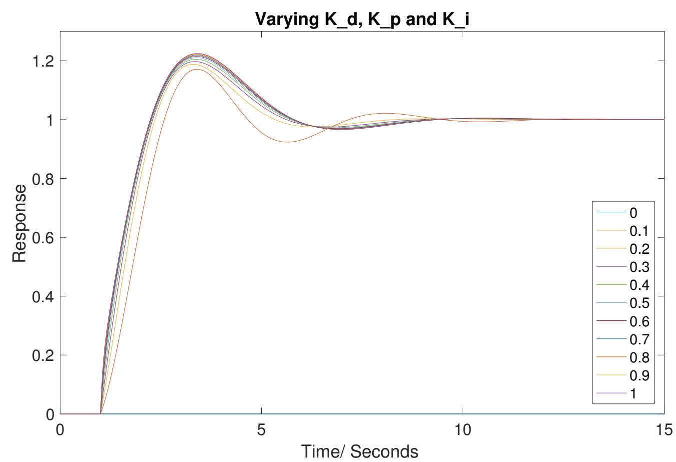

Figure 11: Time-Domain Response For Varying K_p, K_d and K_i

2 PID Design to Achieve Control Requirements

Through implementing the learning gained in part 1, a closed loop-feedback controller was then developed for the theoretical Quanser transfer function. This controller was tuned, through creating a script which outputted the step response of the controller into a table using the `stepinfo()`. Due to the steady state error response observed through using either a P or PD controller, it was decided to iterate different gains for a full PID, to obtain the optimum response. Initially all gains were iterated using the gain-scaling block (discussed in section 1.3). Through observing the different response, a rough estimate was found for all three gains, which could then be individually tweaked based on the parameters shown in Table 15.The first order transfer function obtained in Control part 1, was corrected using a more appropriate method for calculation of τ. $\tau = \zeta \times \omega_n$

2.1 Open Loop Transfer Function PID Controller

Table 1: Effect of Gains on Objectives

	Objectives			
	Tr	Ts	%OS	Ess
Kp	Decrease	Small Increase	Increase	Decrease
Kd	Small Change	Decrease	Decrease	Small Change
Ki	Decrease	Increase	Increase	Eliminate

As the proportional, integral and derivative gain values were adjusted to tune the PID, the characteristics for changing each gain were considered with reference to Table 15 . The controller was then tuned as necessary to best meet the requirements .Testing several PD controller structures, the resulting response showed poor steady state error. It was found that a PID controller with stated gain values produced optimal time domain metrics, which improved upon the provided specifications.

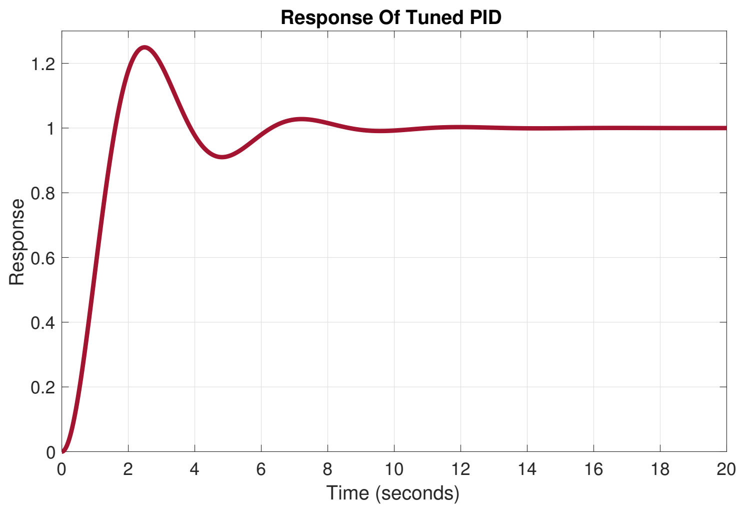

Figure 12: Tuned PID Response

		Task	Tuned PID
Objectives	Tr (s)	12	0.76
	Ts (s)	10	4.6387
	OS (%)	20	5.211
	Ess (%)	5	0.3
Selected Gain Values			
Kp = 0.1, Kd = 0.1, Ki = 0.1			

Figure 13: Tuned PID Transfer Function Values

2.2 Tuned Quanser Controller

A PID controller with the stated proportional, integral and derivative gain values was created inside the SSC17_QuanserPart2_PIDdesign.slx `controller block. The transfer function identified previously was input into the empty `PLANT' block. This allowed observation of the expected Quanser response for the chosen PID and plant configuration. In addition the two other `extreme' plants were also tested to ensure the controller was working effectively. The extreme plants were also used as a procedure for modifying our Controller structure.

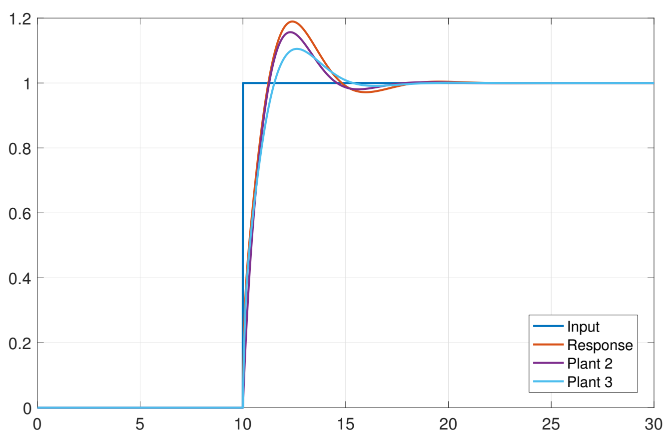

Figure 14: Expected Quanser Repsonse of PID Controller

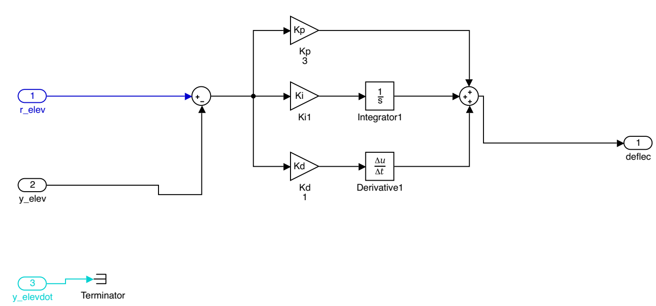

Figure 15: Simulation Used for Theory Controller

References

[1] (2012). Control tutorials for matlab and simulink - introduction: pid controller design, [Online]. Available: http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID (visited on 07/04/2017).