ENEL321 Lab Report

Liam Pribis 81326653

May 4th, 2020

1 Summary

During this lab, 5 different controllers were designed and simulated for use in controlling the roll angle or a model rocket. The controllers were first modeled to determine their probable response, and then run on a test rig. The response of each controller was measured and their performance was analysed. It was found that PID controllers performed better than PD or P controllers.

2 Methodology

Five controllers were used in this lab: one strictly proportional (P) controller, one proportional-derivative (PD), and three proportional-integral-derivative (PID) controllers. Specific gains were generally selected by fixing the other gains of the system and allowing the gain in question to vary over a range. The best controller in this range was selected. This was done using Python, SciPy, and MatPlotLib.

2.1 Proportional Controller

Multiple values for the proportional gain in this controller were tested. Lower values of K_p resulted in less overshoot and rise time, as illustrated in Figure 1, but also had a large negative steady state error. Large values of K_p achieved small steady state errors, but the overshoot quickly became too large as K_p increased. The large overshoot and rise time is due to the fact that when K_p is large and the error is large (at $t = 10$), the commanded angle of the canards will also be very large ($u_p(t) = K_p(r - x)$). The large canard angle will cause the rocket to gain rotational velocity very quickly (resulting in small t_r). Once the rocket reaches the commanded 90°rotation, it's rotational inertia means it will overshoot before beginning to turn back towards the commanded angle. This process may repeat if K_p is large, resulting in oscillation. The negative correlation between K_p and e_{ss} is due to the fact that there is an internal spring in the model of the rocket system. The K_p term is always trying to push the rocket's rotation toward the commanded angle. Larger K_p allows the canards to output more force (due to a larger angle of attack) to counteract this spring, and reduce e_{ss}.

Because of these constraints, a K_p of 0.3 was chosen, which produced a measured M_p of 25% and an e_{ss} of -13°.

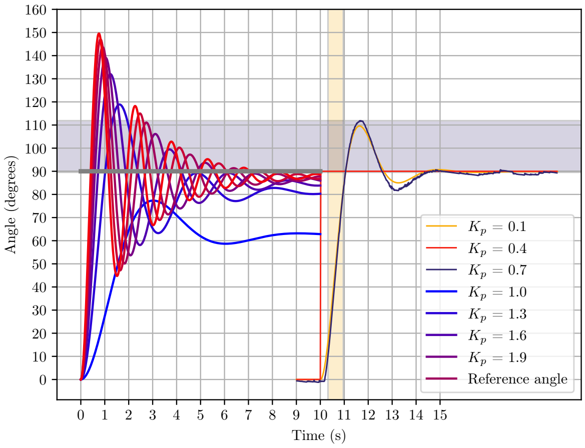

Figure 1: Modeled P control with varying K_p and step input.

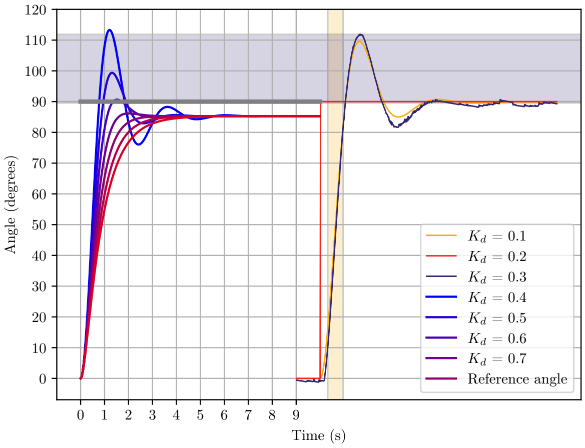

Figure 2: Modeled PD control with K_p = 0.8, varying K_d, and step input.

2.2 Proportional-Derivative Controller

When designing a PD controller, a K_p of 0.8 was chosen due to the tradeoff of $M_p\%$ and e_{ss} as previously discussed. Because a derivative term is present, the K_p can be higher than in the P controller without compromising on overshoot. The derivative term dampens the oscillations present in the proportional-only controller, decreasing the overshoot for equivalent values of K_p. The dampening effect is due to the fact that the K_d term always tries to counteract change in the commanded angle ($u_d(t) = K_d(0 - \dot{x})$). As with K_p, the K_d term also presents a tradeoff. Larger K_d values will add more damping, which decrease the overshoot or remove it entirely (resulting in an overdamped system). Smaller K_d values decrease damping, causing a faster rise time t_r, but more overshoot. This is illustrated in Figure 2. A value of 0.3 was chosen to compromise on rise time and overshoot.

2.3 Proportional-Integral-Derivative Controllers

Three PID controllers were chosen. Controller 2 was chosen with the maximum K_p value allowed in the specifications, to decrease t_r as much as possible. In a model, K_d was then increased until it suitably removed the overshoot, and no more. Having a larger K_d than necessary could cause an overdamped system. Overdamping is undesirable because past the point of critical damping, there is no overshoot at all, and any additional K_d will only cause a decrease in t_r.

Controller 3 was chosen with a K_p of 1.0, which is outside of the range provided in the specifications. Controller 3 also had a smaller K_d than controller 2. Both of these decisions were made to minimise rise time, at the expense of other parameters.

Controller 5 was chosen with a K_p of 0.5 and a K_d of 0.12. Both of these values are smaller than the K_p and K_d values of the other PID controllers. In modelling, controller 5 had a shorter rise time, but also a large overshoot, which is due to its low K_d value.

For all three PID controllers, a K_i value of 0.1 was chosen. Through simulation, it was found that $K_i = 0.1$ reduced steady state error to almost zero. Larger values of K_i increase the overshoot because from $t = 10$ to $t = t_p$, the difference between the commanded and the measured angle is large. This positive difference is summed in each time delta by the controller ($u_i(t) = K_i \int_0^t r - x d\tau$, and $r - x$ is positive), adding to the positive angular momentum of the rocket. When the rocket reaches its commanded set point, the increased momentum means a larger overshoot, demonstrated in Figure 3.

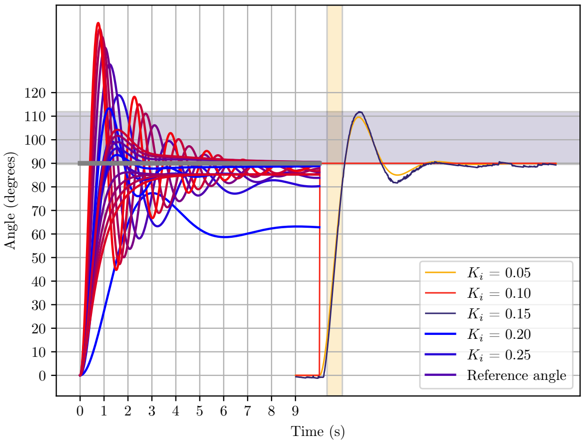

Figure 3: Modeled PID control with $K_p = 0.8$, $K_d = 0.6$, varying K_i, and step input.

3 Results

Table 1: Measured parameters for the controllers.

Controller	Type	K_p	K_i	K_d	t_r	t_p	t_s	$M_p\%$	$e_{ss}(°)$	ξ	ω_d	ω_n
1	P	0.3	0.0	0.00	0.80	11.90	19.88	25.15	-12.96	0.40	1.70	1.85
2	PID	0.8	0.1	0.60	1.34	14.52	19.72	4.00	1.28	0.72	*	*
3	PID	1.0	0.1	0.40	0.70	11.42	12.86	3.69	-0.24	0.72	*	*
4	PD	0.8	0.0	0.30	0.62	11.40	19.88	3.01	-6.81	0.74	*	*
5	PID	0.5	0.1	0.12	0.64	11.68	16.60	24.32	-0.44	0.41	1.90	2.08
specs		≤ 0.8			< 0.9			< 5	$< 1°$	> 0.4		

* Oscillations were not clear enough to determine ω_d or ω_n.

3.1 Controller 1: $K_p = 0.3$

Controller 1 used only proportional control. As expected, it produced a large overshoot and steady state error (from Table 1), due to lack of damping (K_d) and integral control (K_i). The integral term would reduce the steady state error because as time goes on, if there is a recurring positive or negative error, it will be summed by the integral, causing the K_i term to have a larger effect over time. The derivate control would reduce the overshoot and oscillation, because it counteracts and change (derivative) in the response.

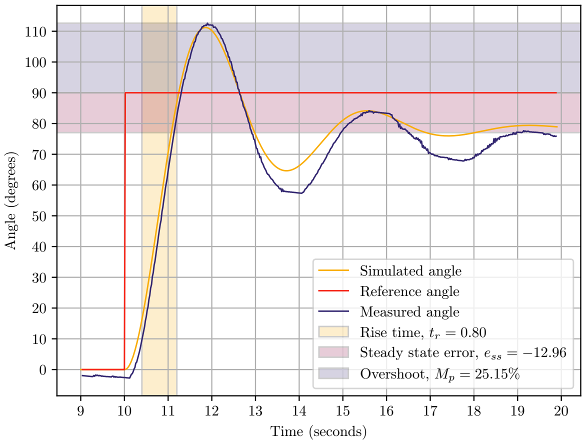

Figure 4: Controller 1 response.

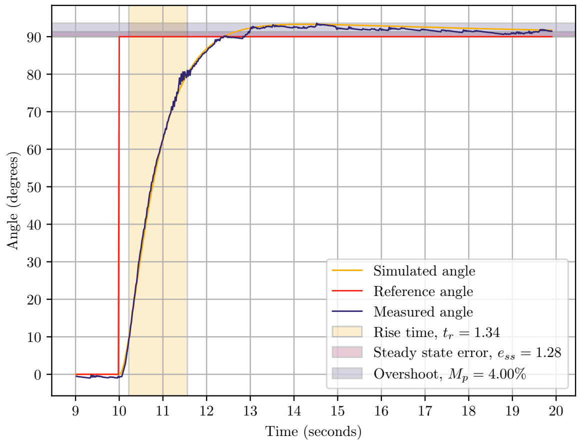

Figure 5: Controller 2 response.

3.2 Controller 2: $K_p = 0.8, K_i = 0.1, K_d = 0.6$

Controller 2 produced a low overshoot/$M_p\%$, a low steady state error, and a high rise time (from Table 1). The rise time and steady state error did not reach specifications. The high rise time, high steady state error, and low overshoot are due to the high K_d gain, causing damping in the response. As shown in Figure 5, controller 2 shows almost no oscillation and a reduced gradient as it approaches 90°. The damping also causes this controller to fail to reach a low enough steady state error, because the K_d term is resisting the change toward a 90°rotation. To optimise this controller, the derivative gain could potentially be reduced some, depending on how much overshoot can be tolerated.

3.3 Controller 3: $K_p = 1.0, K_i = 0.1, K_d = 0.4$

Controller 3 met all specifications except $K_p \leq 0.8$. It produced a fast rise time, very little overshoot, and almost zero steady state error. The main problem with this control scheme is that it saturates the angle of the rocket's canards. This is shown in Figure 6. At $t = 10$, the controller attempts to actuate the fins to >90°. This is in contrast to controller 5 (Figure 7), which never actuates the canards more than 50°.

While this response looks good in the model, in the real world the canards would stall and send the rocket in to an unpredictable state. Saturating the control surfaces also makes this controller very difficult to model accurately, because it becomes non-linear.

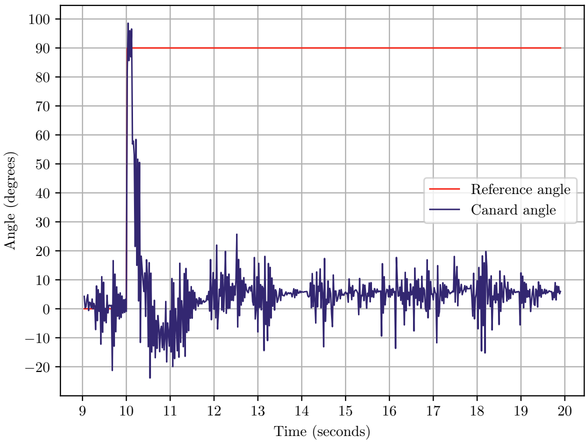

Figure 6: Controller 3 canard actuation.

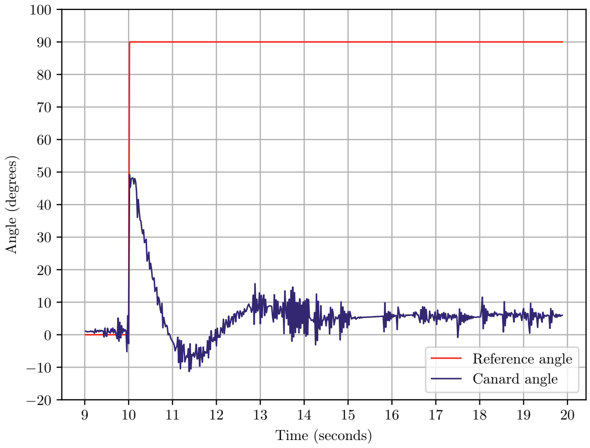

Figure 7: Controller 5 canard actuation.

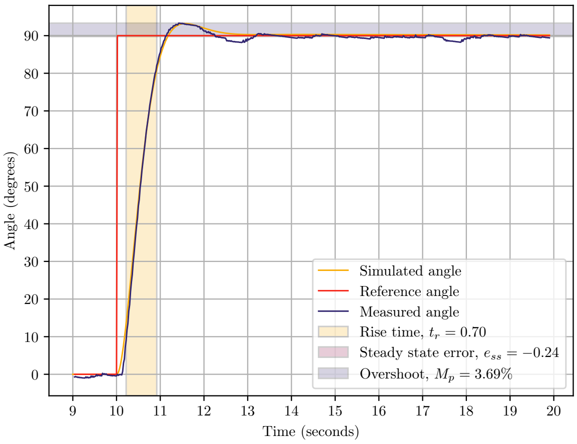

Figure 8: Controller 3 response.

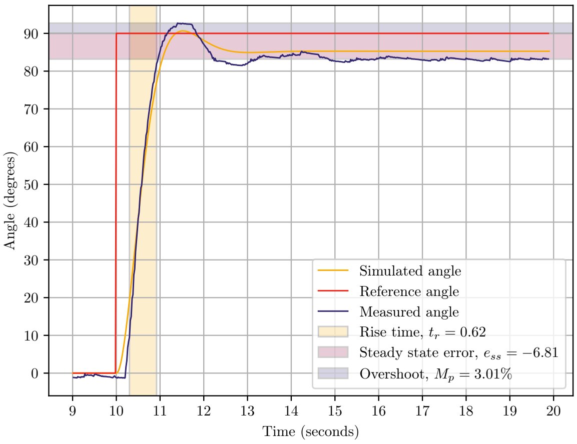

Figure 9: Controller 4 response.

3.4 Controller 4: $K_p = 0.8, K_d = 0.3$

Controller 4 meets all specifications except that of the steady state error. The large steady state error is due to the lack of an integral term. Since there is no integral to accumulate or sum the recurring error, it stays constant at -6.8°.

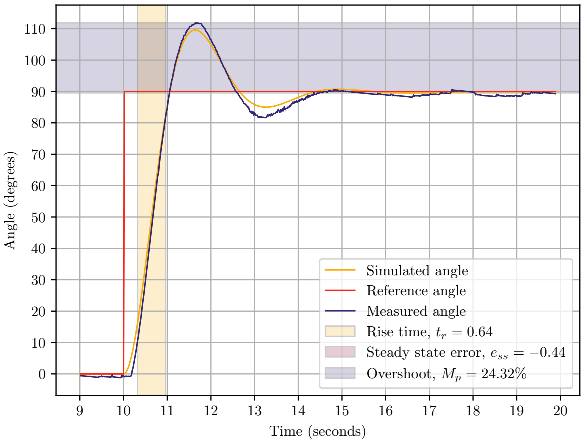

Figure 10: Controller 5 response.

4 Conclusion

The controllers that best met the specifications on the test rig were PID controllers. The strictly proportional controller had a large overshoot and oscillation. The P and PD controllers both had issues with steady state error. One of the PID controllers (controller 2) had a derivative gain that was too large. The excess of damping caused by the derivative gain resulted in a long rise time and relatively large steady state error. If a second iteration of this controller were to be designed, the derivative gain would be reduced.

One of the PID controllers had acceptable parameters, but its large K_p caused it to saturate the actuators. The saturation was not modelled for, causing the response of the system to look normal. If this controller were tested in a physical wind tunnel, it likely would not behave in the same way, showing the importance of physical testing for many control systems.