Assignment 2: Vehicle Stability Control (VSC) Controller Design

RO47017 Vehicle Dynamics and Control

by

Ben Halliwell

12/05/2023

Mechanical Engineering

Department of Mechanical, Materials and Maritime Engineering

Contents

1. Introduction

Electronic stability control (ESC) is important to ensure the driver remains in control of the vehicle even under high lateral loading conditions.
ESC can be used to ensure a vehicle can cope with an exceptional cornering manoeuvre, such as a *Sine with Dwell*.

At 60 km/h the car can achieve the desired input (Figure 1), however at 100 km/h the car becomes completely unstable and ends up travelling backwards from X ≈ 360 m onwards (Figure 2). The car experiences understeer up to Y ≈ 300 m, and then experiences a transition to oversteer as the longitudinal axis of the car turns more rapidly than the displacement of the car (See Appendix).

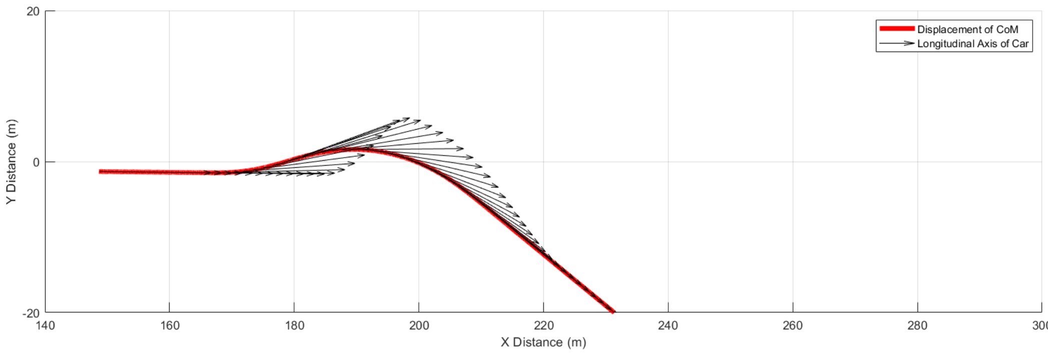

Figure 1: Performance without a controller at 60 km/h

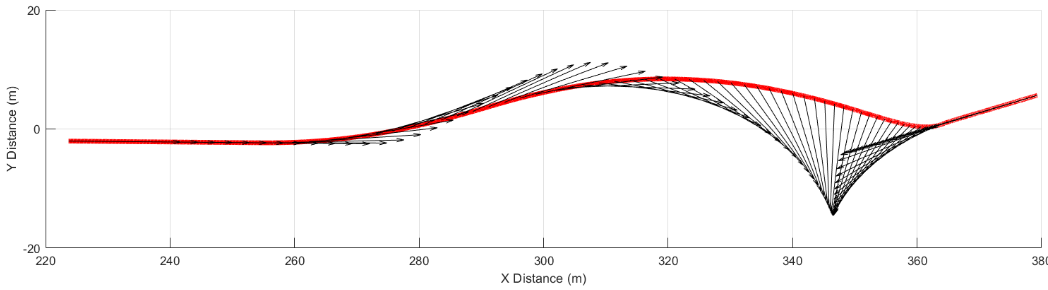

Figure 2: Performance without a controller at 100 km/h

Therefore, ESC should be implemented so that the car can perform this manoeuvre at higher speeds.

2. Vehicle Dynamics Controller

2.1 Reference Generation

Firstly, the steering angle (δ) input must be converted to a reference signal in terms of yaw rate ($\dot{\psi}$).

The steady state yaw rate is a function of the longitudinal velocity, the steering angle and the geometric properties of the car.

$$\dot{\psi}_{ss} = \frac{u\delta}{L + K_{us}u^2/g}; \tag{1}$$

However, following this signal exactly may lead to an impossibly high lateral acceleration. Therefore the reference signal must be saturated.

$$\dot{\psi}_{sat} = \begin{cases} \dot{\psi}_{ss} & \text{if } |\dot{\psi}_{ss}| < \dot{\psi}_{max} \\ +\dot{\psi}_{max} & \text{if } \dot{\psi}_{ss} \geq \dot{\psi}_{max} \\ -\dot{\psi}_{max} & \text{if } -\dot{\psi}_{ss} \leq -\dot{\psi}_{max} \end{cases} \tag{2}$$

Where the maximum yaw rate is:

$$\dot{\psi}_{max} = 0.85\frac{\mu g}{u} \tag{3}$$

However, this leads to a discontinuity in the gradient of the yaw rate, which needs to be addressed. This can be achieved by passing the signal through an estimate of the transfer function of the bicycle model.

$$\dot{\psi}_{ref} = \frac{\omega_0(u)^2(1 + \tau(u)s)}{s^2 + 2\zeta(u)\omega_0(u)s + \omega_0(u)^2}\dot{\psi}_{sat} \tag{4}$$

Natural frequency and damping ratio.

$$\omega_0(u)^2 = \left(\frac{C_{\alpha,f} + C_{\alpha,r}}{um}\right)^2\left(1 + \frac{K_{us}u^2}{gL}\right), \quad \zeta(u) = \frac{1}{\sqrt{1 + K_{us}u^2/(gL)}} \tag{5}$$

The time constant was chosen to be equal to the longitudinal velocity as this gave a good reference signal profile.

$$\tau(u) = u \tag{6}$$

Since the saturation level is inversely proportional to the speed (Equation **??**, even though the steady state reference signal at 100km/h has a greater magnitude than at 60 km/h, the final reference signal has a lower magnitude at 100 km/h.

2.2 Controller Saturation

Initially, it was very easy to optimise the controller to follow the reference signal almost exactly. This is because there was no limit on the maximum yaw moment the car could provide. According to Kobayashi, a 268 kg racecar could produce a maximum yaw moment of 3.5 kNm(Kobayashi et al., 2023). Using the same ratio, the 1380kg car in the simulation could produce 18 kNm of yaw

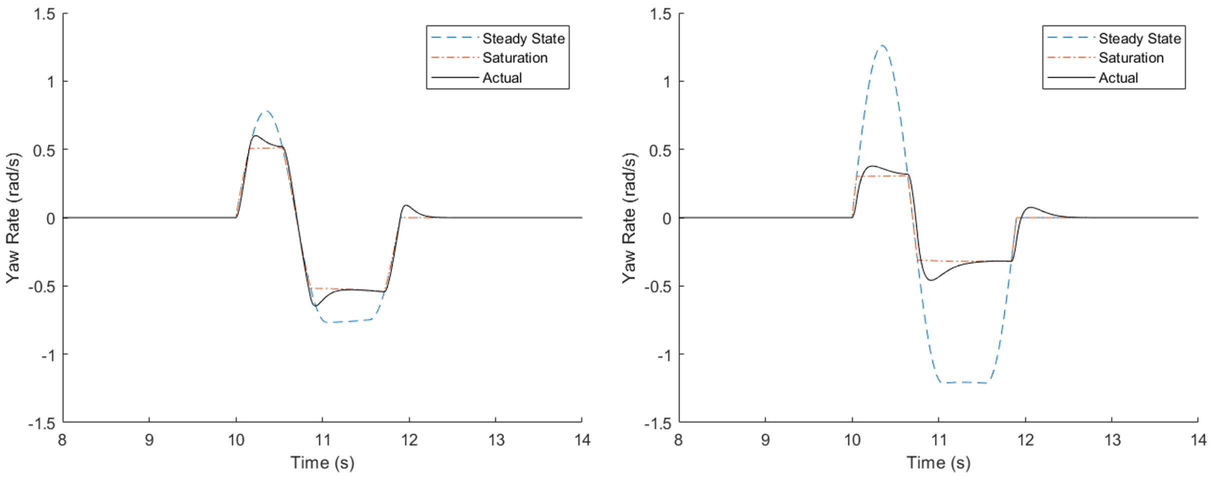

Figure 3: Reference Signal at 60 (L) and 100 km/h (R)

moment. However, the car in simulation is not a racecar so will not be able to produce the same yaw moment, so a safety factor of 1.5 was implemented to give a maximum yaw moment of **12 kNm**.

2.3 Optimisation Parameters

The optimum controller would follow the reference signal as closely as possible.

$$e_1 = \sum_{i=i_{min}}^{i_{max}} |\dot{\psi}_{ref,i} - \dot{\psi}_i| \tag{7}$$

Also, the body sideslip angle (β) should also be minimised as it would be uncomfortable for the driver to experience a high amount of sliding during this manouevre. This is the oppposite of drift control which would want to maximise the body sideslip angle.

$$e_2 = \sum_{i=i_{min}}^{i_{max}} |\beta_i| = \sum_{i=i_{min}}^{i_{max}} \left| \psi_i - \arctan\left(\frac{V_{y,i}}{V_{x,i}}\right) \right| \tag{8}$$

Finally, due to the saturation the controller would rapidly oscillate between a maximum and minimum yaw moment at high gains (Appendix). This would require the brakes to rapidly change the brake pressure, which will lead to increased wear due to fatigue. Therefore the difference in yaw moment was minimised.

$$e_3 = \sum_{i=i_{min}}^{i_{max}} |\dot{M}_{z,i}| \tag{9}$$

The optimum controller would minimise these three errors.

$$e_{tot} = \prod_{i=1}^{3} e_i \tag{10}$$

2.4 PID Controller

The PID controller received the yaw rate error ($\dot{\psi}_{error}$) as an input and outputted the required yaw moment (M_z), and was gain scheduled based on the longitudinal velocity of the vehicle (u).

$$M_z = K_p(u)\dot{\psi}_{error} + K_I(u)\int \dot{\psi}_{error}dt + K_d(u)\frac{d(\dot{\psi}_{error})}{dt} \tag{11}$$

Firstly the P gain was optimised, followed by the D gain and then the I gain for both 60 and 100km/h (Appendix). A power law was then used to gain schedule between the two speeds.

$$K(u) = Au^B \tag{12}$$

$$B = \frac{\log(K_{100}/K_{60})}{\log(u_2/u_1)}, \quad A = \frac{K_{60}}{u_1^B} \tag{13}$$

This is beneficial as it allows the gain to approach zero when the vehicle comes to a stop, preventing any unwanted braking at these speeds when ESC is not required. However the controllers were optimised without gain scheduling, and then the gain scheduling was added between speeds. This is because it was assumed the decrease in longitudinal speed during the manoeuvre is negligible. The final longitudinal velocity is 51.2 and 90.6 km/h respectively for the two speeds, around a 10% decrease. It would require significantly more computational time to optimise gain scheduling at the respective speeds and would also require more computational power from the ESC unit as the gains would be of the form $K = A(u)u^{B(u)}$ with A(u) and B(u) being power-law functions themselves. Therefore this 10% error is acceptable.

A transport delay was also added to the controller (Appendix) to prevent any algebraic loops occurring as the yaw rate was fed back into the controller.

Table 1: PID Controller Parameters

Gains	60 km/h	100 km/h	Gain Scheduled
K_P	1.5e6	1.9e6	4.11e5 $u^{0.46}$
K_I	3e2	2e3	9.04e-3 $u^{3.7}$
K_D	3e4	1.6e5	2.79 $u^{3.3}$

2.5 LQR Controller

A linear-quadratic regulator (LQR) controller pre-calculates the the optimum gain from a differential equation (Equation 14 based on the theoretical system dynamics.

$$\underline{\dot{x}} = \mathbf{A}\underline{x} + \mathbf{B}\underline{u} \tag{14}$$

The "Bicycle Model" was chosen as a simple but comprehensive dynamic system.

$$\begin{bmatrix} \dot{u} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{C_{\alpha,f}+C_{\alpha,r}}{um} & \frac{-l_f C_{\alpha,f}+l_r C_{\alpha,r}}{um} - u \\ \frac{-l_f C_{\alpha,f}+l_r C_{\alpha,r}}{I_z u} & -\frac{l_f^2 C_{\alpha,f}+l_r^2 C_{\alpha,r}}{I_z u} \end{bmatrix} \begin{bmatrix} u \\ r \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha,f}}{m} \\ \frac{l_f C_{\alpha,f}}{I_z} \end{bmatrix} \delta \tag{15}$$

However, the B matrix was replaced with $B = [0, \quad 1/I_{zz}]^T$ as the input is a yaw moment instead of a steering angle.
Then the MATLAB function lqr(A, B, Q, R) was used to generate the look-up table of gains (K) for the LQR controller. The state-cost weighted matrix (Q) and the input-cost weighted matrix (R) were optimised, to provide the minimum error as stated in section 2.3. Since equation 15 has 2 outputs ($\dot{v}$, $\dot{r}$) and 1 input (δ), Q is a 2x2 matrix and R is a scalar.

Since the longitudinal velocity does change by 10%, the LQR gains need to change with velocity. Therefore R was varied for the possible velocities the car would experience (1 to 100 km/h). Changing the value of $Q_{1,1}, Q_{1,2}$ and $Q_{2,1}$ had a negligible or negative impact on the overall error, so values of zero was chosen for them.

$$R_i = u_i, Q = \begin{bmatrix} 0 & 0 \\ 0 & K_Q \end{bmatrix} \tag{16}$$

Using the previous error (Equation 10) led to an optimum gain of 1.5e9 and 2e9 at 60 and 100 km/h respectively, which equates to a power law of $1.5e8u^{0.56}$.

3. Simulation of Sine with Dwell manoeuvre

3.1 Reference Tracking

The PID controller follows the 100 km/h reference signal more closely than at 60 km/h with an error of 0.42% and 0.92% error (for 9<t<14 seconds) respectively. This is because the reference yaw rate is greater at 60 km/h due to the lateral acceleration limiting in the reference generation, however both errors are sufficiently small. The LQR controller performs slightly worse with an error of 1.18% for 60km/h and 0.55% at 100km/h.

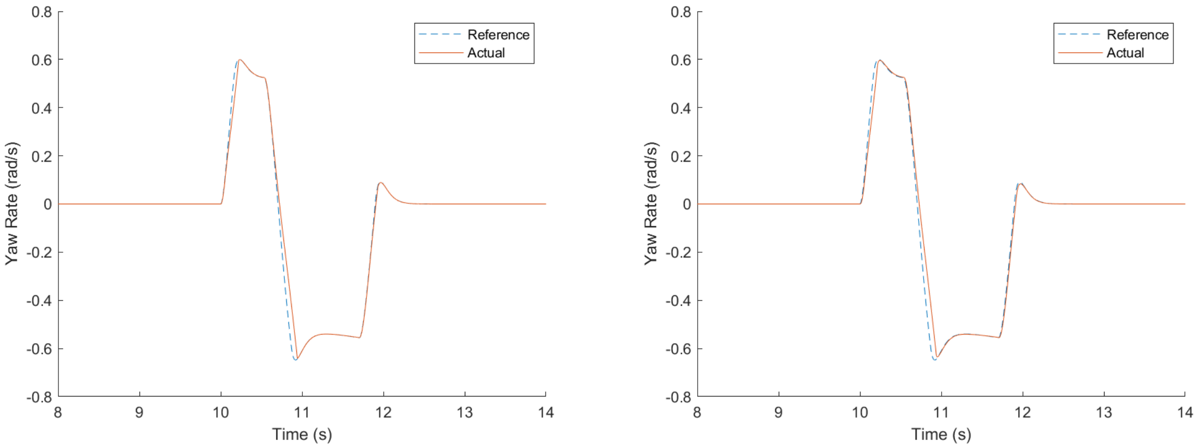

Figure 4: Reference Tracking at 60 km/h for PID (L) and LQR (R) controllers

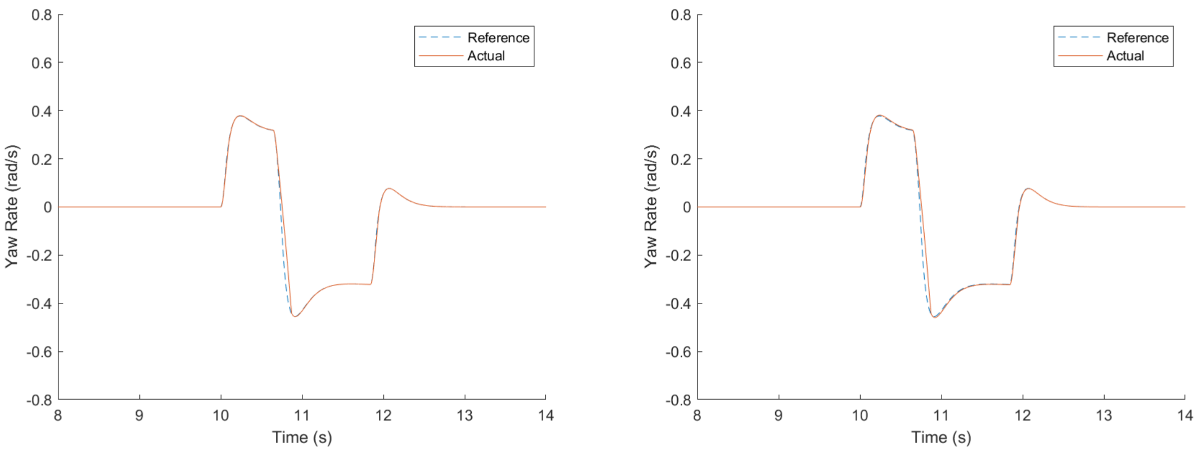

Figure 5: Reference Tracking at 100 km/h for PID (L) and LQR (R) controllers

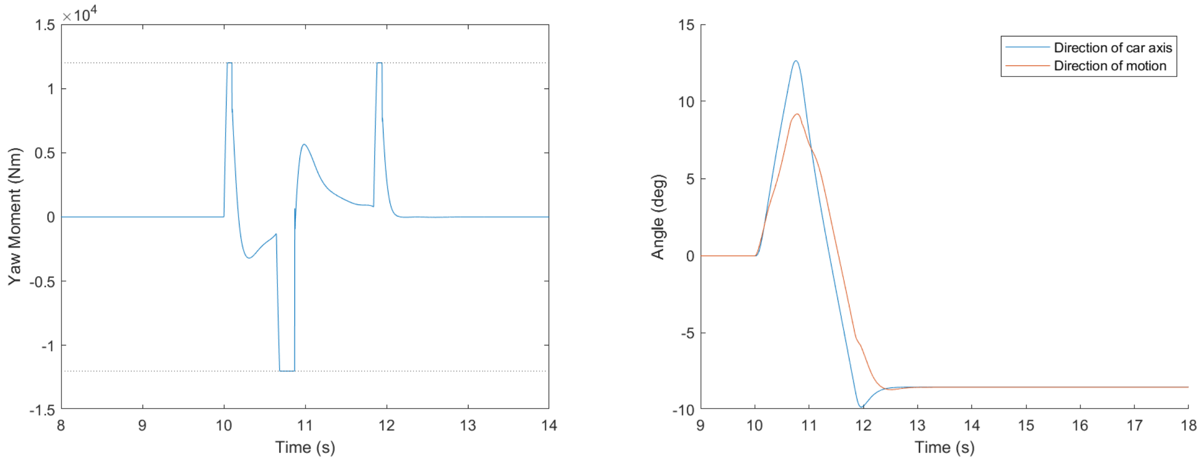

Figure 6: PID controller yaw moment (L) and LQR yaw (R) at 100 km/h

3.2 Other metrics

The applied yaw moment is slightly smoother for the LQR controller (Appendix), however the difference is negligible. Again, due to the lower acceleration saturation at 60 km/h, the yaw moment exceeds the 12000Nm limit for longer than at 100 km/h, although it is only occurs for a maximum of 0.3 seconds (Figure 6.1).

The body sideslip angle remains below 5 degrees for both PID and LQR controllers (Figure 6.2). Unlike the two previous metrics, both controllers perform worse at 100 km/h than at 60 km/h.

4. Controller performance with sensor noise

A yaw rate noise of ± 1 rad/s was added to the system. This has a significant impact of the reference tracking (Table 2) compared to without noise, but the two controllers perform almost identically.

4.1 Yaw Rate Performance Table

Table 2: Effect of Noise on Performance

Controller	Speed (km/h)	Noise	Yaw Rate Error (%)
PID	60	No	0.92
PID	100	No	0.42
LQR	60	No	1.18
LQR	100	No	0.55
PID	60	Yes	7.06
PID	100	Yes	7.14
LQR	60	Yes	7.07
LQR	100	Yes	7.15

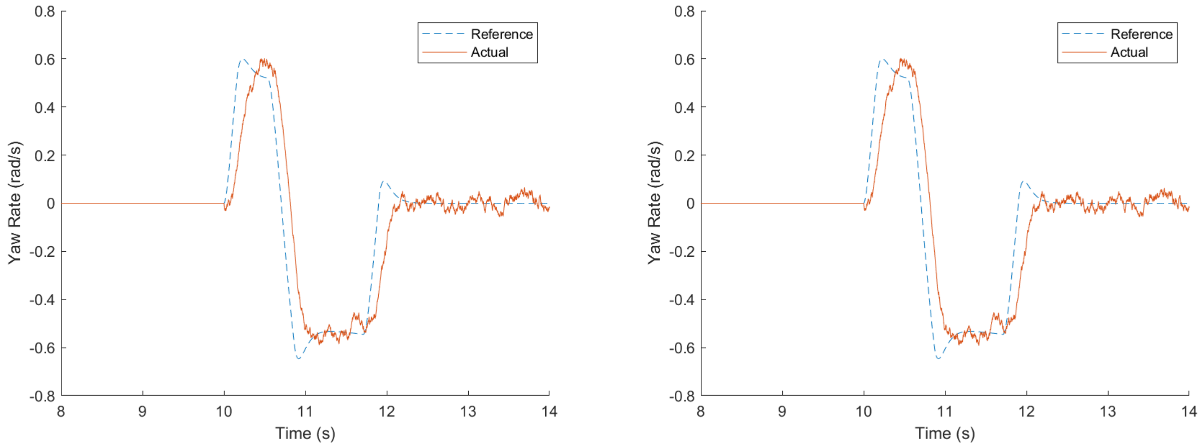

Figure 7: Reference Tracking at 60 km/h for PID (L) and LQR (R) controllers with noise

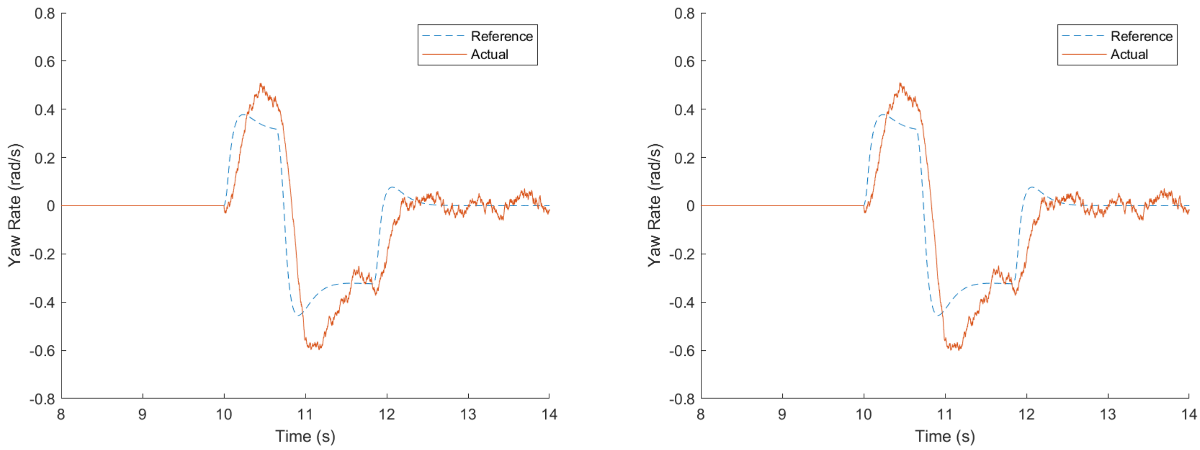

Figure 8: Reference Tracking at 100 km/h for PID (L) and LQR (R) controllers with noise

4.2 Other metrics

Unfortunately, the addition of noise causes the yaw moment to oscillate between the upper and lower limit (Figure 9) which is undesirable. Noise did not increase the maximum sideslip angle, but it did prevent the car from travelling in a straight line after the manouevre was completed. Both PID and LQR controllers performed very similarly.

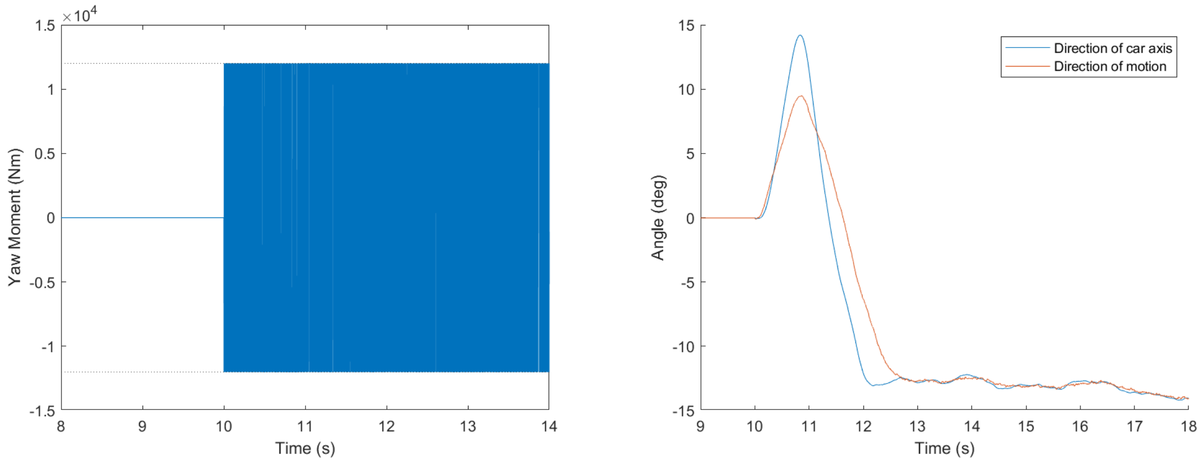

Figure 9: PID controller yaw moment (L) and LQR yaw (R) at 100 km/h with noise

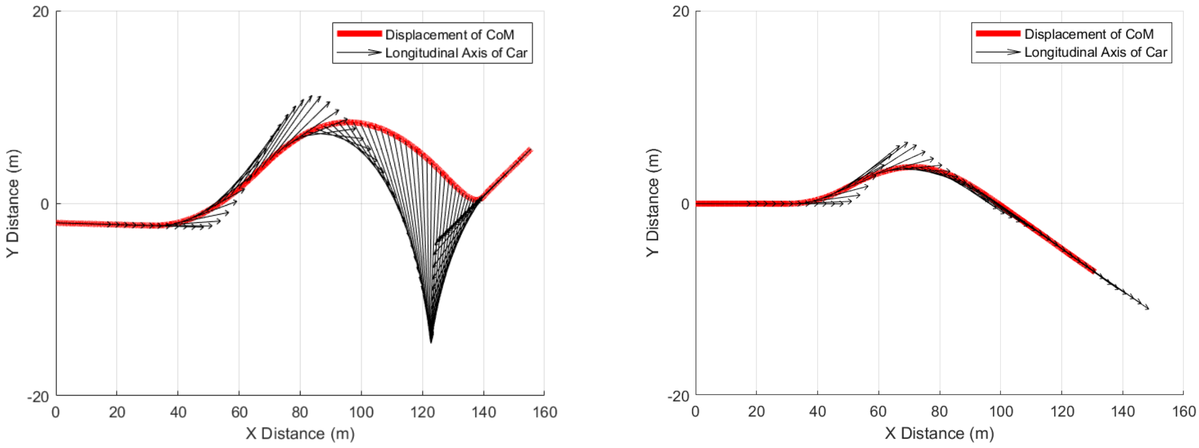

Figure 10: Planar View without no controller (L) and PID controller with noise (R)

5. Reflection

The most challenging part of this project was implementing LQR as I had not used it before commencing this project. This helped broaden my understanding of control methods other than PID and allowed me to test my ability to synthesise information from academic papers and MATLAB's website.

I found optimisation of the controller parameters was quicker, mainly because I utilised the optimisation plan and functions from the previous homework assignment. It also helped that there were less parameters involved which reduced the computation time.

I also learned about reference generation instead of using a step or ramp signal, which will be useful for more real-world applications of control that I might face in the future.

6. Conclusion

Both controllers were successful in preventing the car from entering the non-linear regime and becoming unstable which occurred when no controller was implemented which is the main purpose (Figure 10). Overall, the PID controller is a more effective controller than LQR, but it is not significantly better. LQR also requires significantly less computation power, as it only requires lookup table and multiplication of the corresponding gains whereas PID requires differentiation and integration of the error signal.

The reference signal has a larger impact on the stability performance than the controller, so it would be more effective to spend time optimising reference generation than more time optimising the controller.

However, both controllers suffer with sensor noise. This can be reduced by tuning down the derivative and proportional gains and increasing the integral gain to compensate. This can reduce the applied yaw moment so that it is not varying between the saturation limits. For example, by reducing the P gain by 150 and increasing the I gain by 3 times, the yaw moment profile can be improved with the reference tracking only decreasing by 2%. This could be a useful solution if continually applying the maximum yaw moment is a major issue.

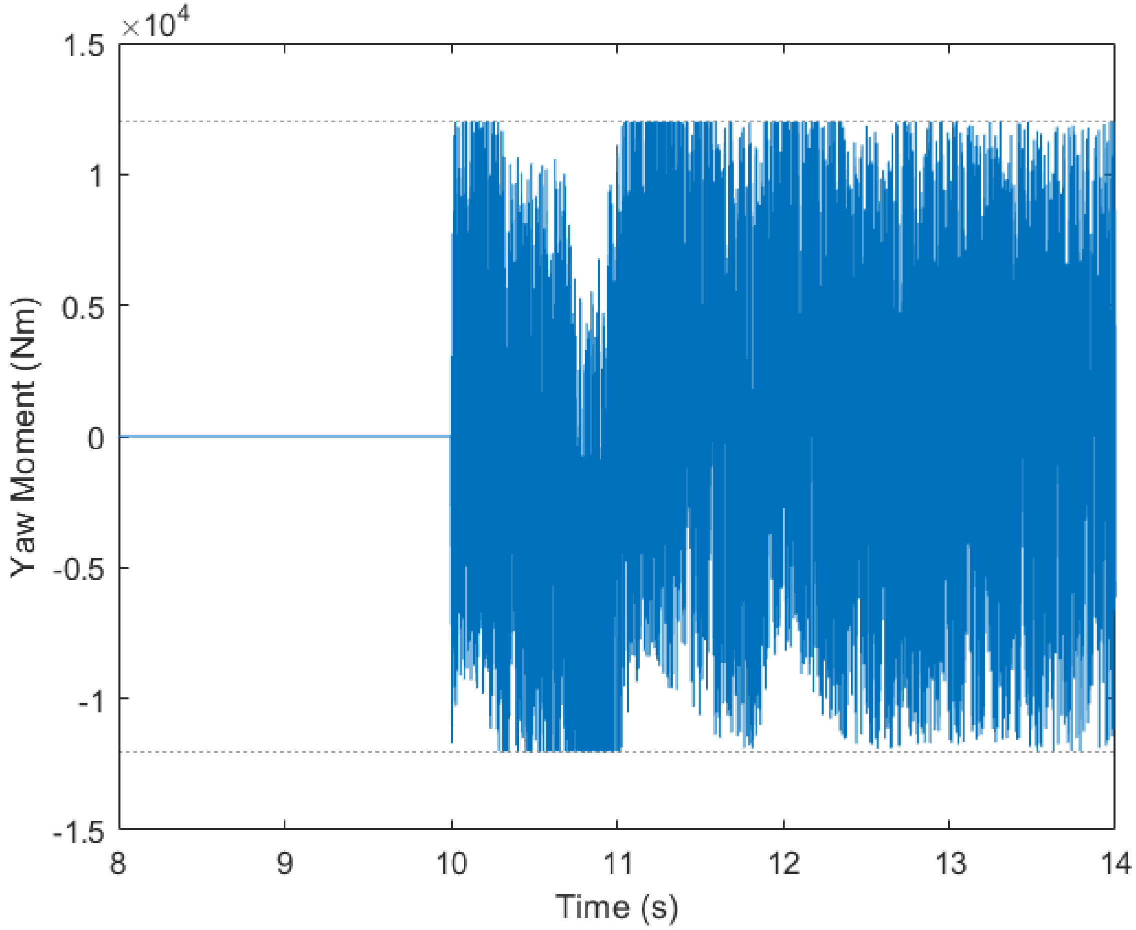

Figure 11: Yaw moment for PI controller (P = 2740, I = 0.0271)

7. Appendix

7.1 Simulink Diagrams

Pink is the input/output, orange is the steady state block, blue is the acceleration saturation and green is the non-linear response.

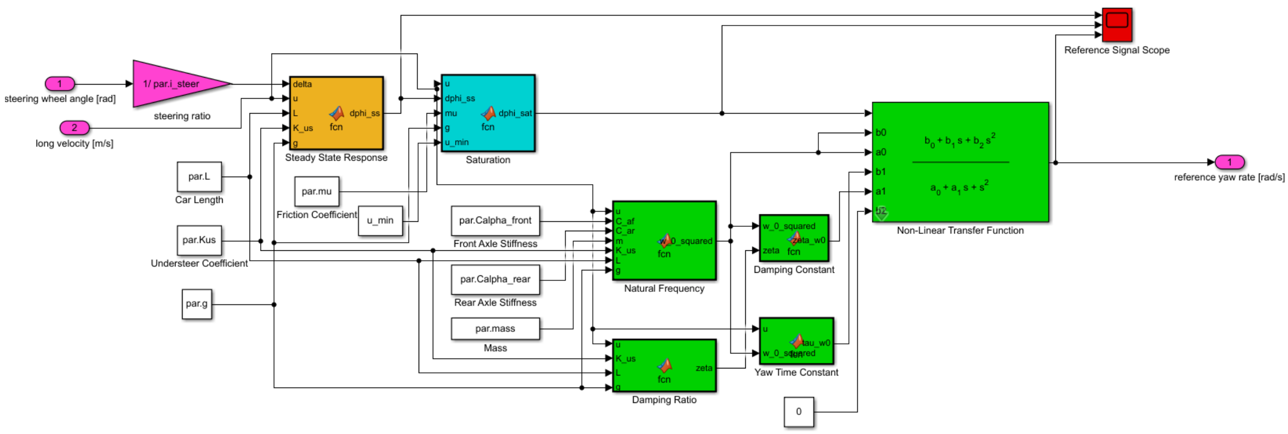

Figure 12: Reference Generator Diagram

Orange is the LQR controller, blue is the PID controller.

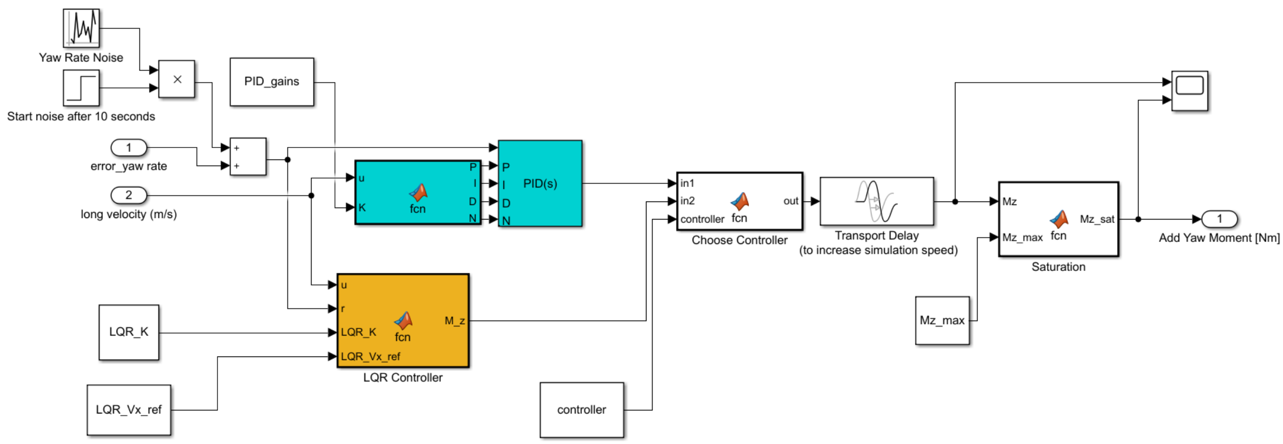

Figure 13: High Level Controller Diagram

7.2 Vehicle Dynamics Controller

7.2.1 PID Controller

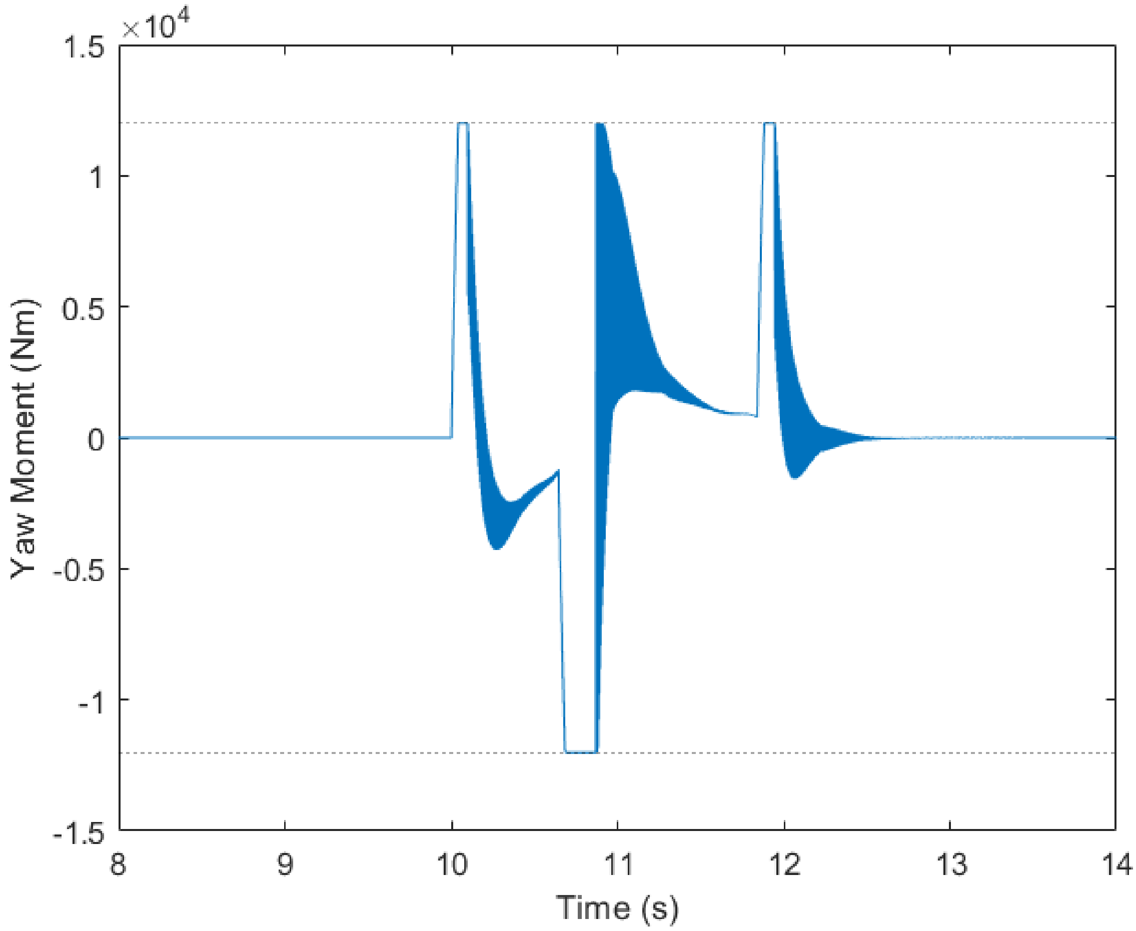

Figure 14: Yaw Moment Oscillations at high controller gains

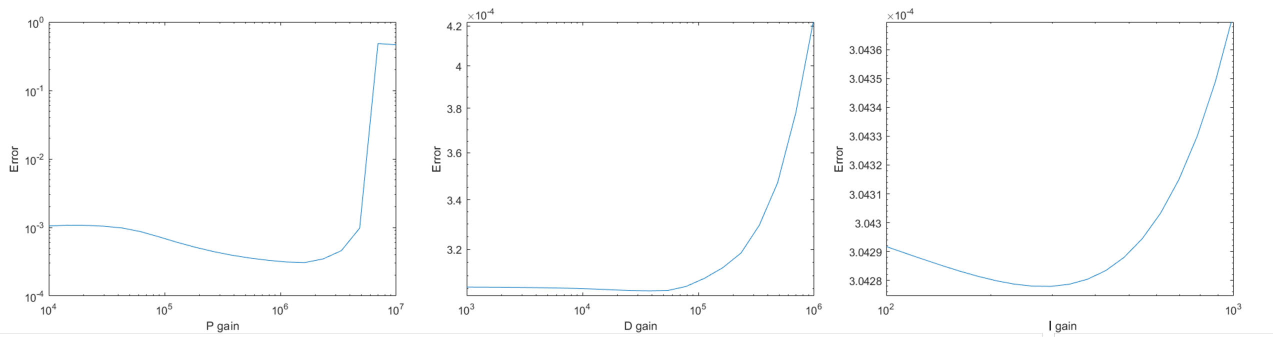

Figure 15: PID Gain Optimisation at 60 km/h

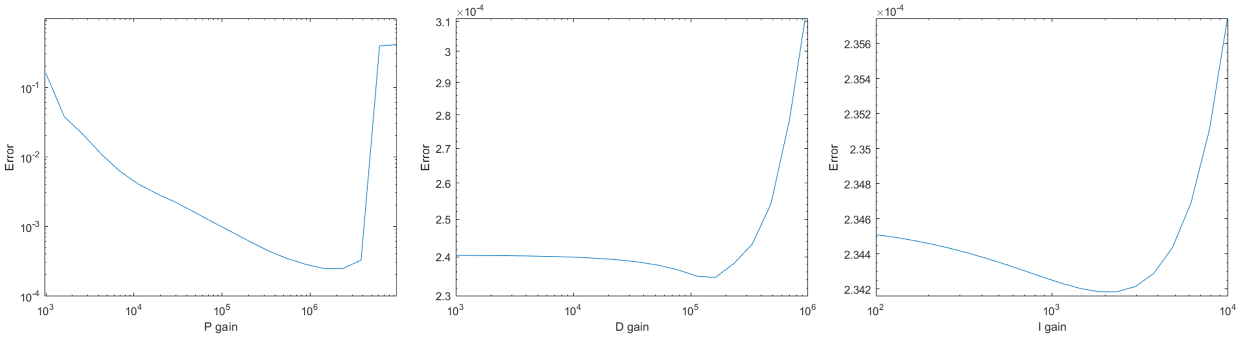

Figure 16: PID Gain Optimisation at 100 km/h

7.2.2 LQR Controller

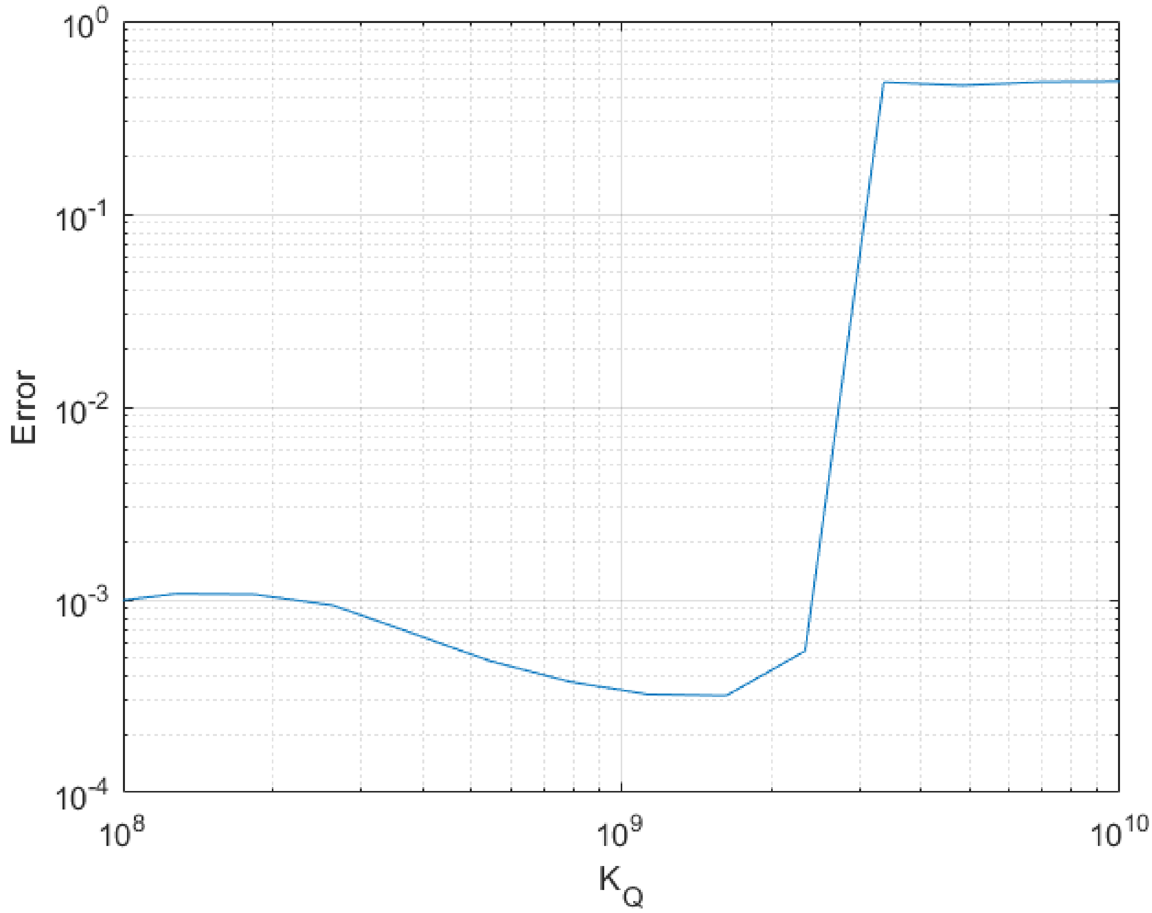

Figure 17: Q Gain Optimisation at 60 km/h

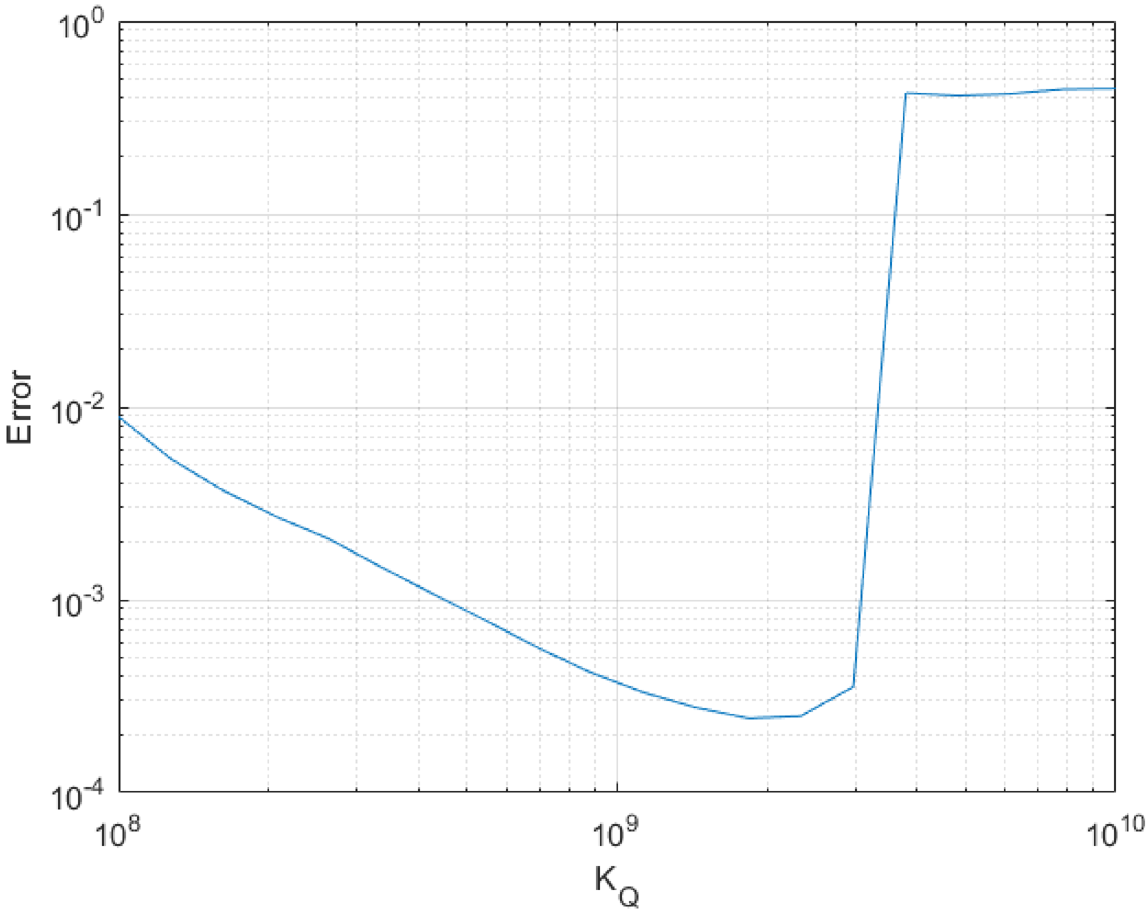

Figure 18: Q Gain Optimisation at 100 km/h

7.3 No Noise

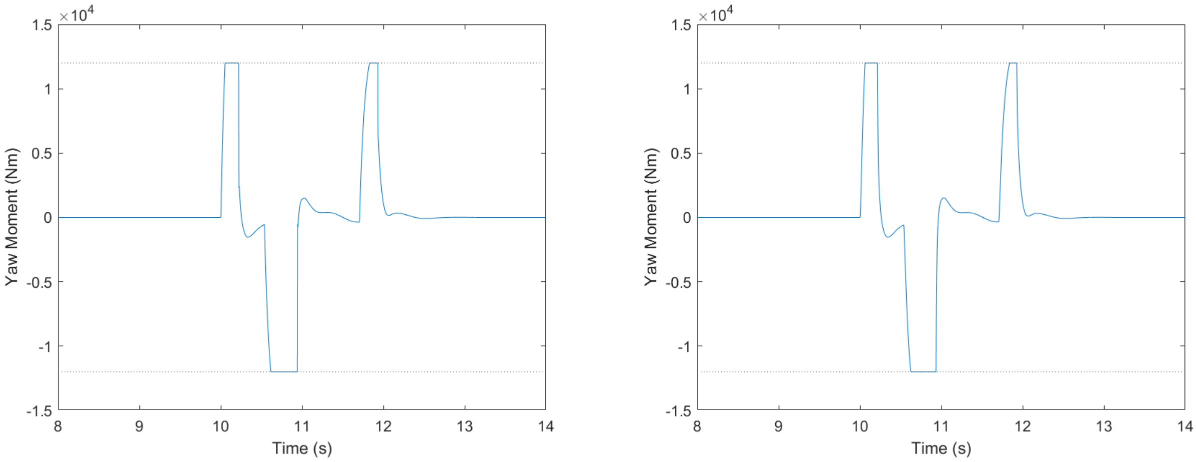

Figure 19: Yaw moment at 60 km/h for PID (L) and LQR (R) controllers

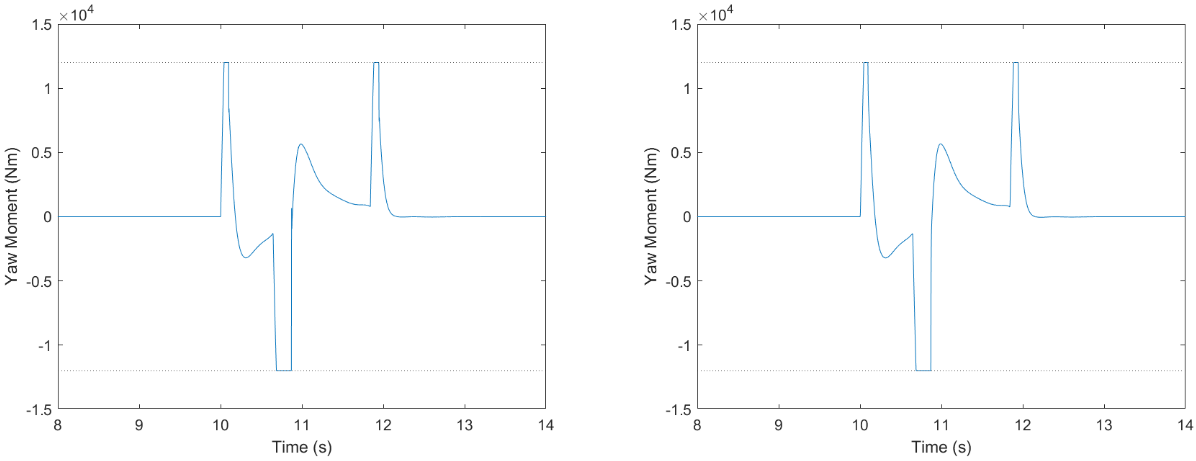

Figure 20: Yaw moment at 100 km/h for PID (L) and LQR (R) controllers

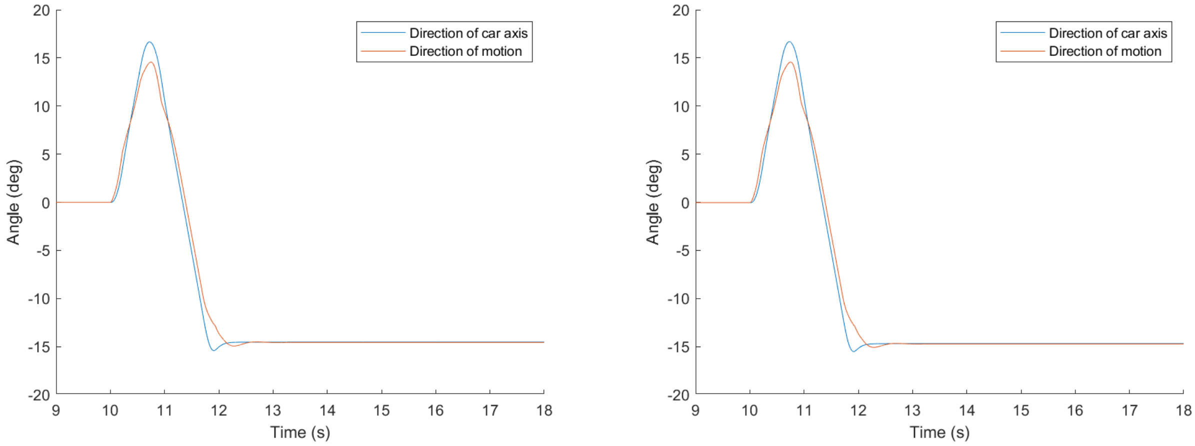

Figure 21: Yaw at 60 km/h for PID (L) and LQR (R) controllers

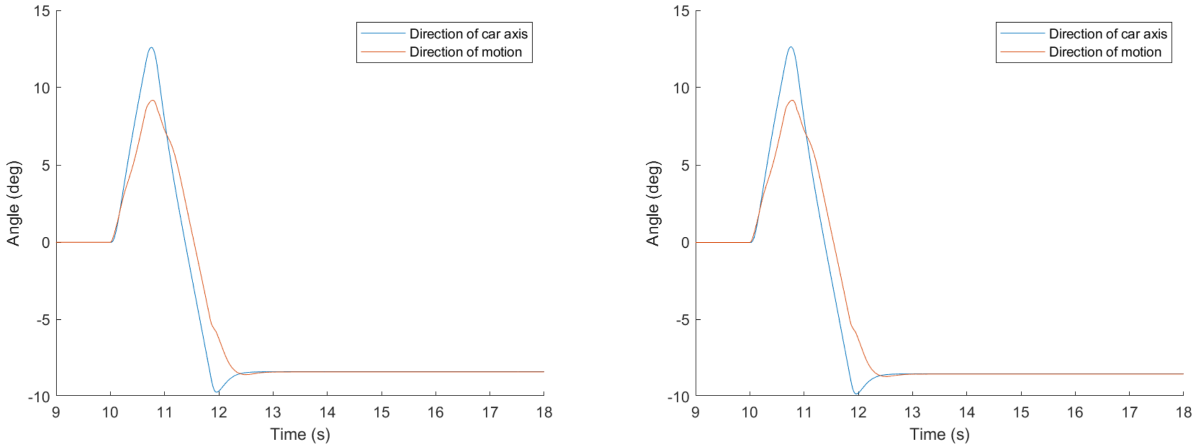

Figure 22: Yaw at 100 km/h for PID (L) and LQR (R) controllers

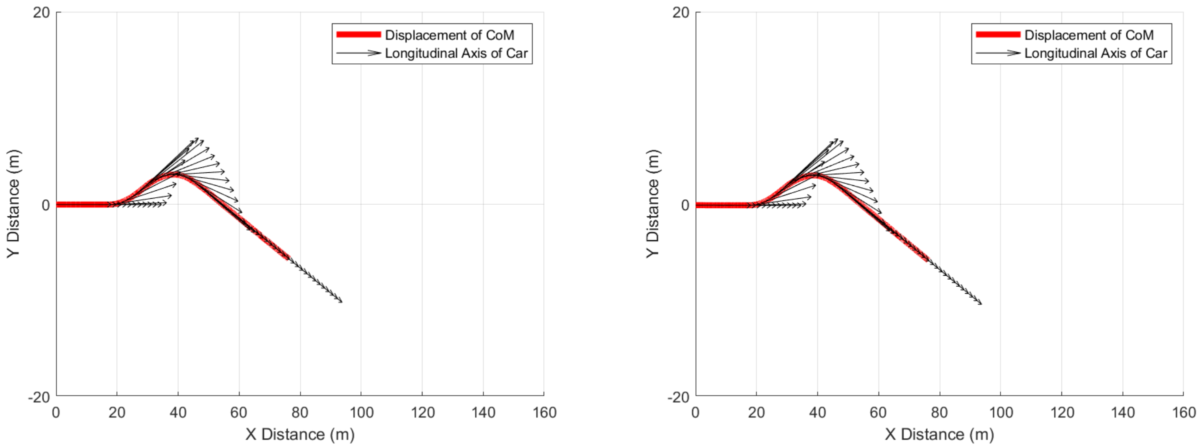

Figure 23: Planar View at 60 km/h for PID (L) and LQR (R) controllers

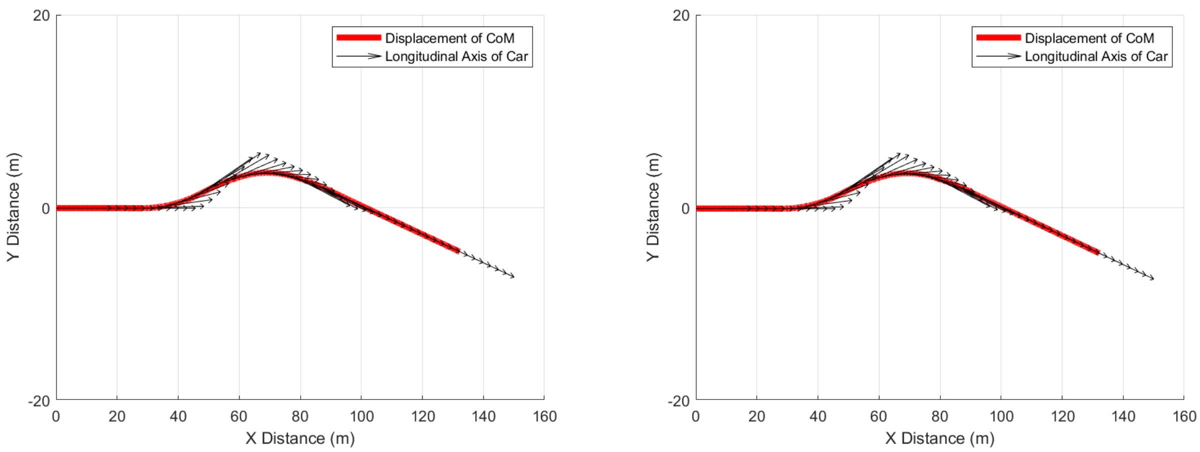

Figure 24: Planar View at 100 km/h for PID (L) and LQR (R) controllers

7.4 Noise

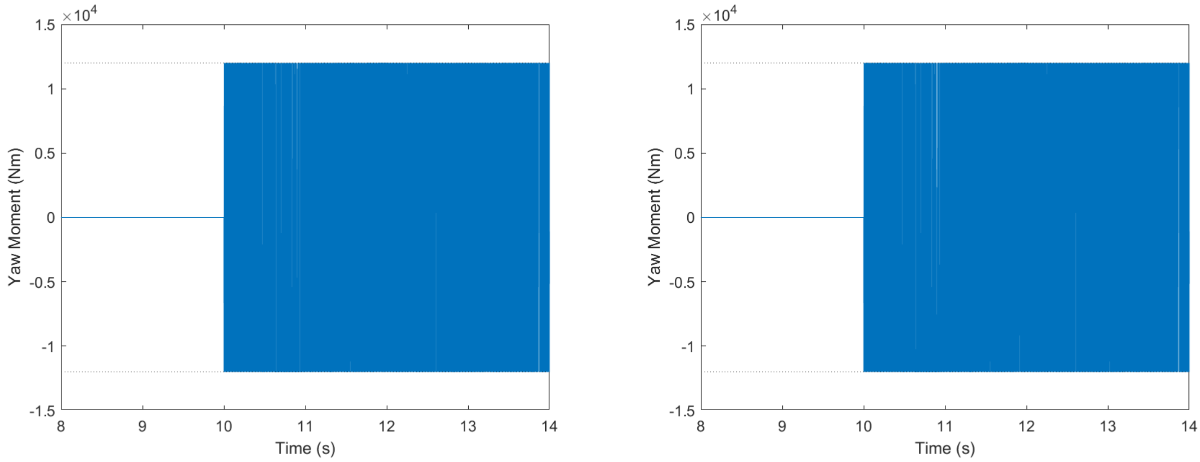

Figure 25: Yaw moment at 60 km/h for PID (L) and LQR (R) controllers

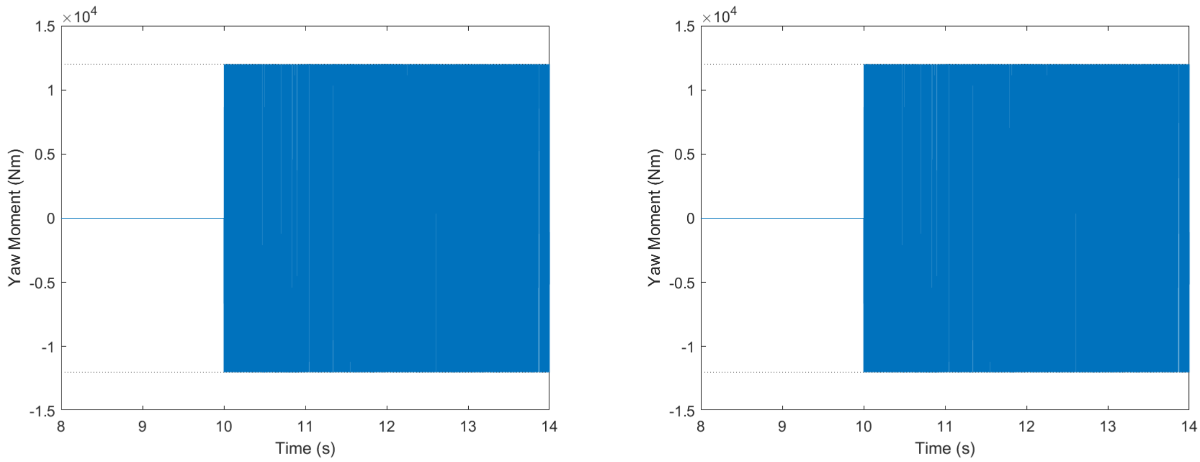

Figure 26: Yaw moment at 100 km/h for PID (L) and LQR (R) controllers

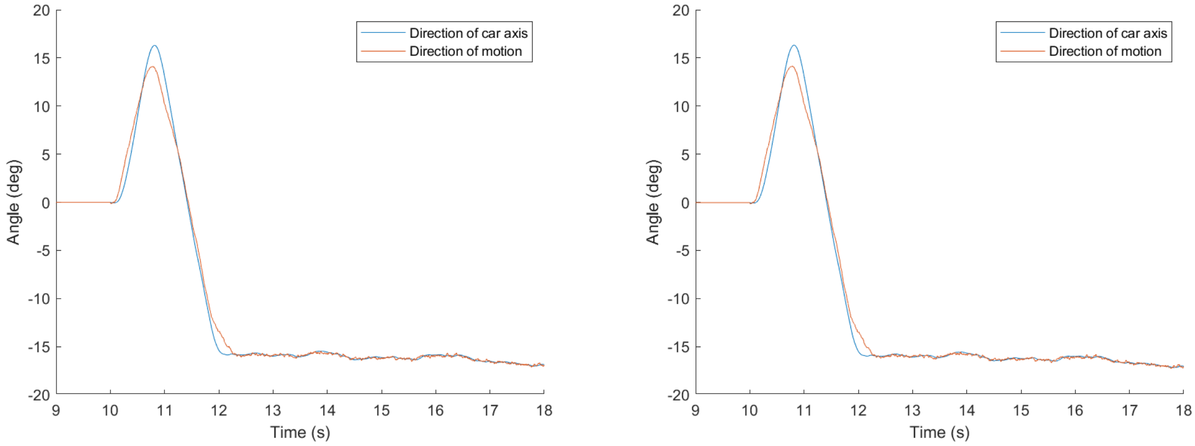

Figure 27: Yaw at 60 km/h for PID (L) and LQR (R) controllers

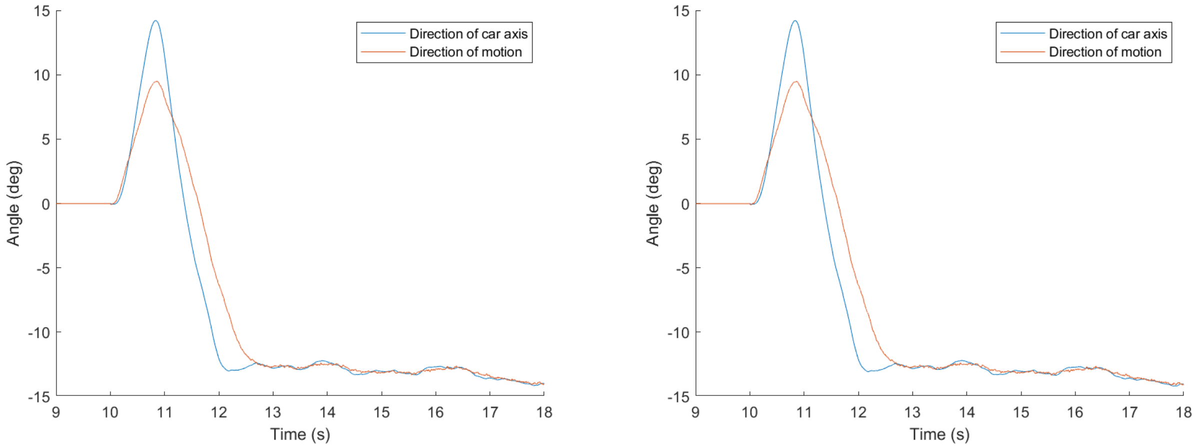

Figure 28: Yaw at 100 km/h for PID (L) and LQR (R) controllers

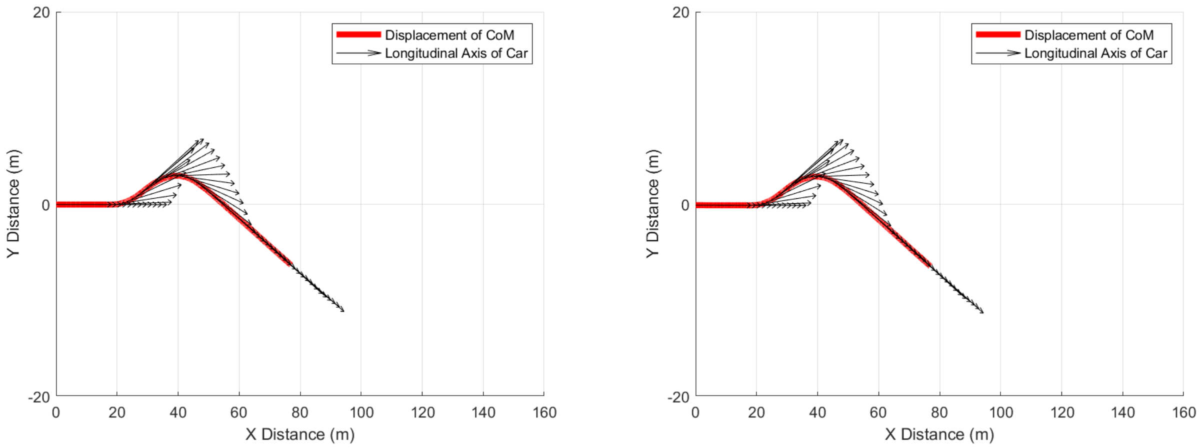

Figure 29: Planar View at 60 km/h for PID (L) and LQR (R) controllers

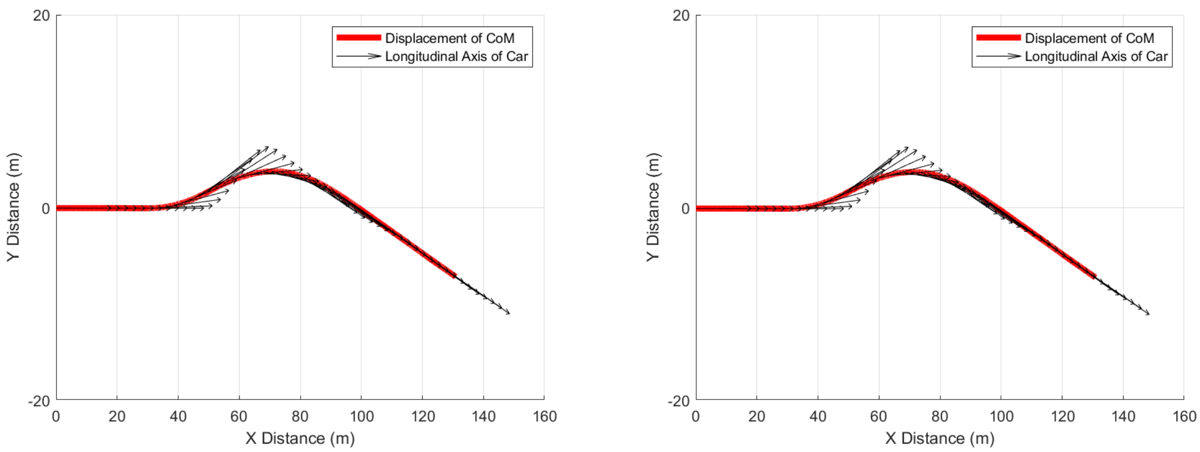

Figure 30: Planar View at 100 km/h for PID (L) and LQR (R) controllers

References

Kobayashi, I., Kuroda, J., Uchino, D., Ogawa, K., Ikeda, K., Kato, T., . . . Kato, H. (2023). Research on yaw moment control system for race cars using drive and brake torques. *Vehicles*, *5*(2), 515–534.