REGBOT BALANCE ASSIGNMENT - NOTE 2

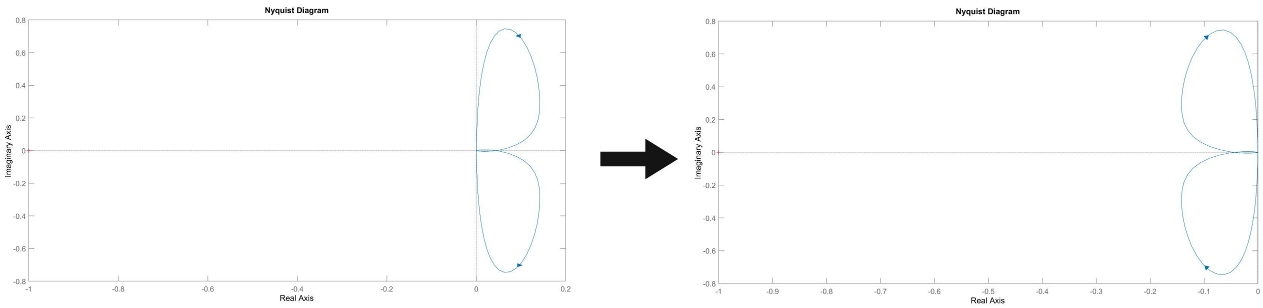
Members: Tan Rui Bin (S226445), Sim Qin Li (S226458), Georemon Russelraj (S226450)

Introduction

In this assignment, we aim to control the movements and positions of the REGBOT. A Simulink model has been set up to simulate the task and we shall calculate the working parameters and feed them through MATLAB. The results from MATLAB will be compared with the actual output by the REGBOT.

Balance Controller

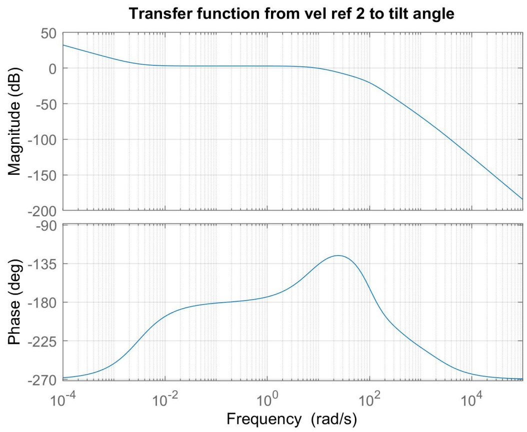
The first task involves getting the REGBOT to balance itself. With the wheel speed controller as provided in Simulink, we first want to linearise the transfer function from *vel_ref* to the output *tilt_angle*. This will allow us to obtain the open-loop Bode plot in order to decide on the type of controller required. However, the transfer function that we get is unstable, as depicted by the left Nyquist plot.



After obtaining the poles and zeros from the transfer function, we are able to determine that there is one RHP in the open-loop transfer function. Upon checking the Nyquist plot, we can observe that there is no counter-clockwise encirclement of the -1 point. We can also observe that the Nyquist plot is in the positive real axis domain. Therefore the value of K_{PS}, the Nyquist plot will only stretch further into the positive real axis if K value is positive. Therefore, K has to be negative to ensure a counterclockwise encirclement of the -1 point as the graph will now be reflected along the imaginary axis. This will ensure that the number of counterclockwise encirclements (Z) around the pole -1 is equal to the number of open-loop poles (P) in the RHP which shows that the closed loop system will be stable.

Once we have included the post-integrator controller ($C_{PI,post}$ and *stabilised*), we will linearise the system from *vel_ref2* to the output *tilt_angle* to get an updated transfer function and Bode plot.

$$G\_tilt = \frac{5.893e05\ s^4 + 7.71e08\ s^3 + 1.435e11\ s^2 + 2.566e12\ s + 7.683e09}{s^7 + 2418\ s^6 + 1.317e06\ s^5 + 1.977e08\ s^4 + 1.312e10\ s^3 + 1.796e11\ s^2 - 9.658e11\ s - 1.525e13}$$

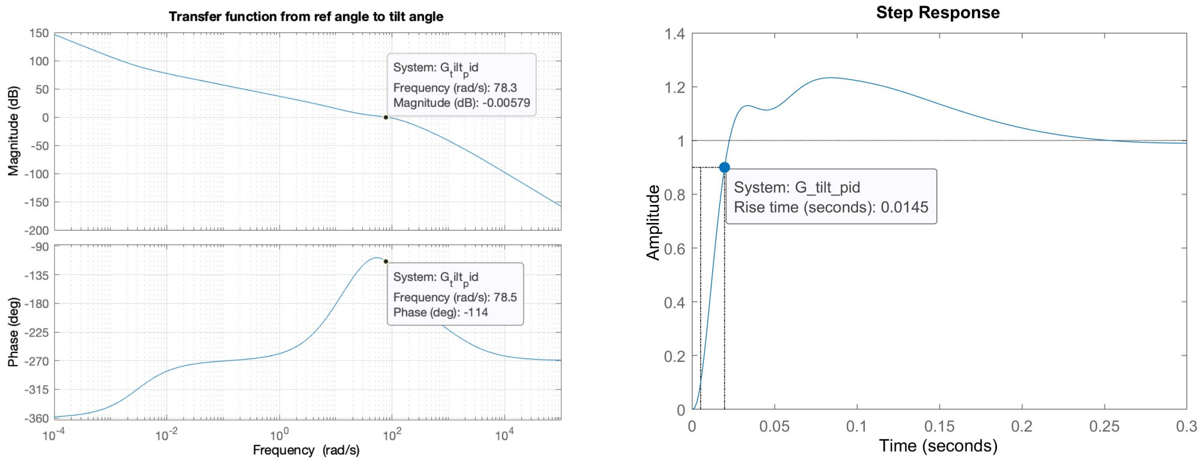


Open-loop bode plot with post-integrator

From the above Bode plot of the open-loop transfer function, it can be observed that there is a positive gain margin and a positive phase margin which indicates stability in a system.

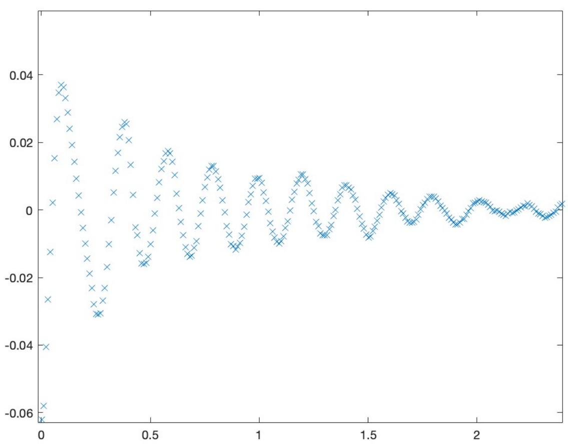
After including the PI-Lead controller, we linearised the transfer function from *ref_angle* to *tilt_angle* and checked for the new open-loop Bode plot shown below. We see that the magnitude graph is monotonically decreasing, and at the crossover frequency, the phase margin is about 66 degrees.

We then closed the loop and plotted the step response for the tilt controller. Despite the overshoot, the system has a fast response with settling time within 0.2 s. We are also able to observe that there is no RHP in the closed loop transfer function which indicates that the system is stable.



Open-loop bode plot with PI-Lead controller (left) and closed loop step response (right)

$$G\_tilt\_pid = \frac{1.263e07\ s^7 + 1.726e10\ s^6 + 4.054e12\ s^5 + 2.513e14\ s^4 + 6.339e15\ s^3 + 6.908e16\ s^2 + 2.625e17\ s + 7.854e14}{s^{10} + 2666\ s^9 + 1.916e06\ s^8 + 5.242e08\ s^7 + 6.214e10\ s^6 + 3.432e12\ s^5 + 4.357e13\ s^4 - 2.547e14\ s^3 - 3.781e15\ s^2}$$



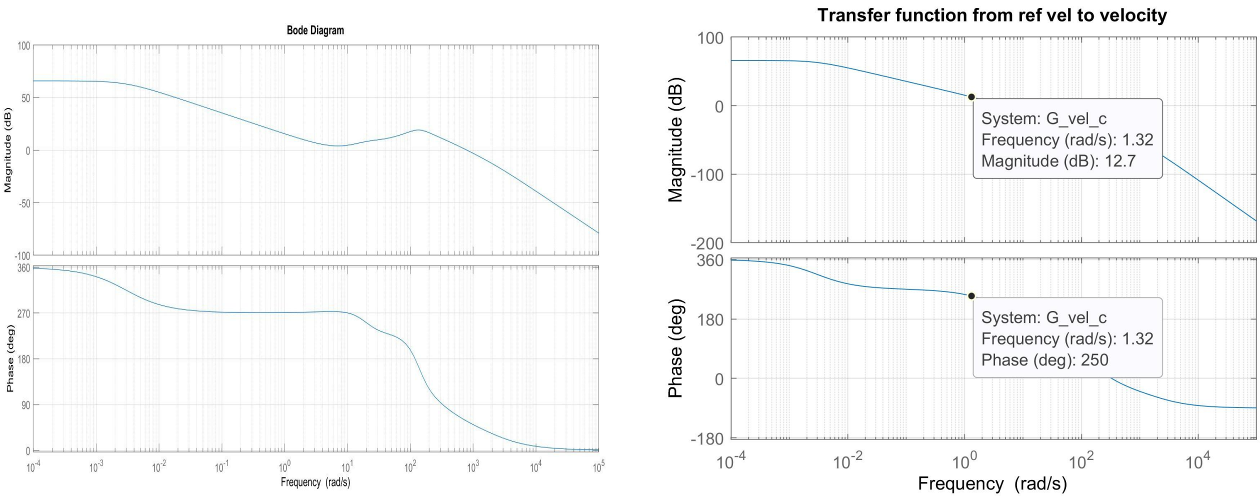
We then loaded the calculated parameters into the REGBOT and ran the following mission with only the tilt controller enabled: `bal=1 : time = 4`

From the initial run, we found that the system was not as stable as the Simulink model. After some experimentation with the controller variables such as K_p, tau_i and tau_d, the REGBOT managed to attain the desired tilt though encountering significant oscillations, as shown in the step response on the left. This could be due to various differences between the Simulink model and the REGBOT, as well as additional undesired disturbances.

Velocity Controller

For the velocity controller, we linearise the system from *ref_angle* to *velocity* to obtain the transfer function and open-loop Bode plot as shown below.

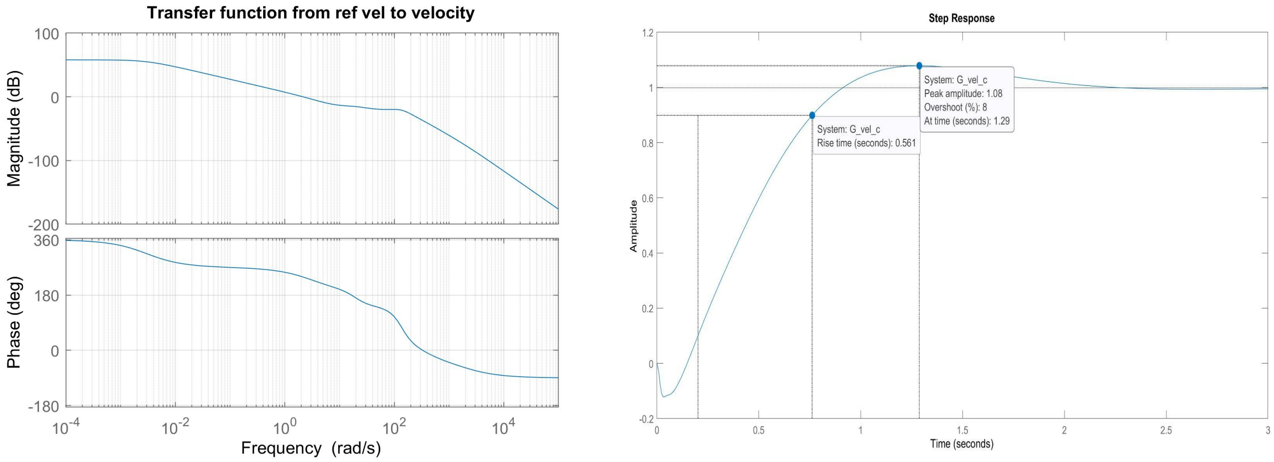
$$G\_vel = \frac{-1.129e06\ s^8 - 1.443e09\ s^7 - 3.345e11\ s^6 - 2.058e13\ s^5 - 4.969e14\ s^4 - 4.177e15\ s^3 + 1.617e16\ s^2 + 4.107e17\ s + 1.559e18}{s^{10} + 2666\ s^9 + 1.916e06\ s^8 + 5.368e08\ s^7 + 7.94e10\ s^6 + 7.485e12\ s^5 + 2.949e14\ s^4 + 6.085e15\ s^3 + 6.53e16\ s^2 + 2.625e17\ s + 7.854e14}$$



Open loop bode plot for velocity before (left) and after (right) first-order low-pass filter

A first-order low-pass filter was added to reduce the gain beyond a particular frequency chosen. The filter's time constant and break frequency determine the frequency at which the magnitude starts to decrease at a rate of 20dB per decade. To achieve a more optimal cut-off frequency, a K_p value of 0.40 (slightly higher than the calculated 0.232 for 70 degrees phase margin) is used to prevent an overly sluggish response.

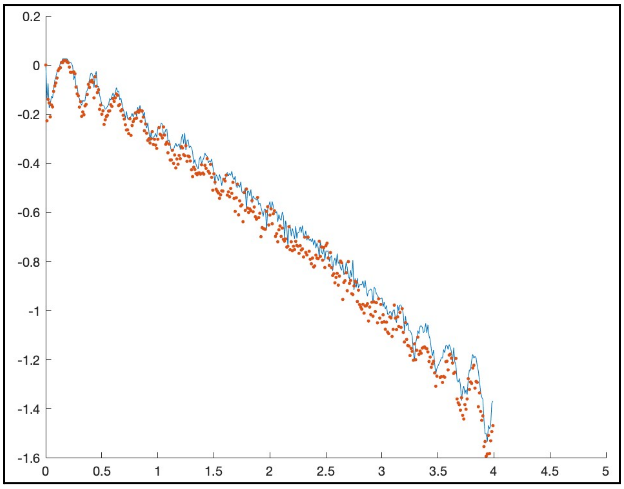
As obtained from the open loop Bode plot below (with the P-controller and first-order filter), the phase margin is approximately 70 degrees as expected and the magnitude shows a monotonic decrease which suggests that the gain of the system dies out towards higher frequency. Upon closing the feedback loop, the step response shows that the system is relatively responsive with a rise time of 0.561second and approximately 2 seconds to reach steady state. Also the amplitude at steady state is 1, which suggests no steady state error and therefore presents a suitable outcome for our controller.



Open loop bode plot with P-controller and first-order filter (left) and step response for velocity controller (right)

G_vel_c =

$$\frac{-1.505e06\, s^8 - 1.924e09\, s^7 - 4.46e11\, s^6 - 2.744e13\, s^5 - 6.625e14\, s^4 - 5.57e15\, s^3 + 2.156e16\, s^2 + 5.476e17\, s + 2.079e18}{s^{11} + 2670\, s^{10} + 1.925e06\, s^9 + 5.432e08\, s^8 + 8.119e10\, s^7 + 7.75e12\, s^6 + 3.199e14\, s^5 + 7.068e15\, s^4 + 8.559e16\, s^3 + 4.801e17\, s^2 + 8.756e17\, s + 2.618e15}$$



The controller parameters were then transferred to the REGBOT GUI under the velocity controller. We ran the following mission:

```
bal=1, vel=0: time = 2
vel=0.5: time = 2
```

Unfortunately, the REGBOT did not move as specified and had a constant acceleration, causing the velocity to continuously increase. We were unable to achieve the desired outcome despite adjusting the control parameters.

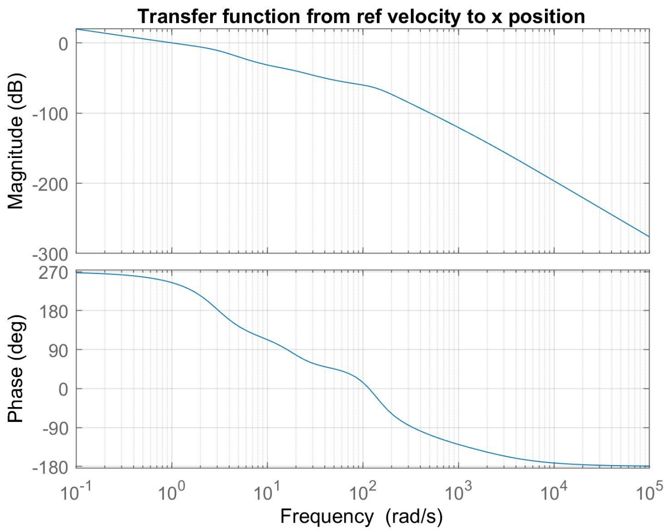
The graph shows the left and right wheel velocities in the two different colours.

Position Controller

With the balance and the velocity managed, we now aim to control the REGBOT's position. A transfer function is obtained from *ref_vel* to *x_pos*, along with the open-loop Bode plot.

G_pos =

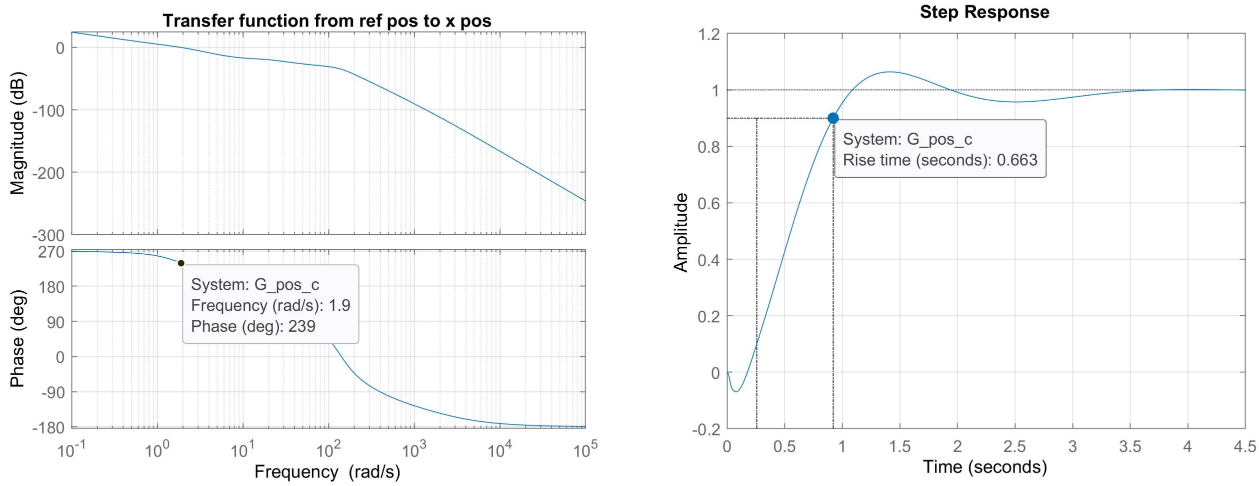
$$\frac{-8.654e05\ s^8 - 1.106e09\ s^7 - 2.564e11\ s^6 - 1.578e13\ s^5 - 3.809e14\ s^4 - 3.203e15\ s^3 + 1.24e16\ s^2 + 3.148e17\ s + 1.195e18}{s^{12} + 2670\ s^{11} + 1.925e06\ s^{10} + 5.432e08\ s^9 + 8.102e10\ s^8 + 7.563e12\ s^7 + 3.059e14\ s^6 + 6.703e15\ s^5 + 8.233e16\ s^4 + 4.91e17\ s^3 + 1.185e18\ s^2 + 1.198e18\ s}$$



Open-loop bode plot of position without controller

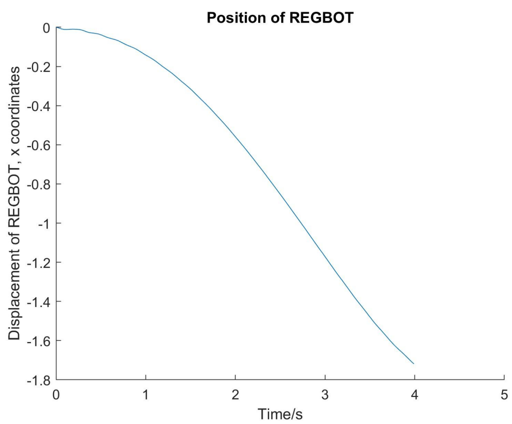
The phase margin is 70 degrees which seems quite optimal but it is also observed that the crossover frequency is quite low at 1 rad/s which may lead to a slower response. However, as required by the task, the system should be able to move with a max speed of more than 0.7 m/s which suggests that a faster response is preferred.

In order to continue maintaining the optimal phase margin of 60 to 70 degrees while improving the responsiveness of the system, we have included a P-Lead controller with alpha selected to be 0.03 and K_p to be 1.7. The moderately low alpha value will have a more aggressive contribution to the phase while a K_p greater than 1 will cause the magnitude curve to shift upwards. The crossover frequency becomes larger, resulting in a larger bandwidth for faster response.



Open loop bode plot with P-Lead controller (left), closed-loop step response of system (right)

As seen from the open loop bode plot with the P-Lead controller, the phase margin is approximately 60 degrees as expected, with a larger crossover frequency and a monotonic decrease in the magnitude plot which serves as a good basis for closing the loop. The step response for the close-loop system shows a sufficiently fast response with a rise time of 0.663s yet not having an excessively large overshoot. The amplitude at steady state is 1 which suggests no steady state error and therefore the system is able to reach the simulated desired position and stop there.



For the position controller implementation on the REGBOT, we have adopted the mission to move to a given position by stating the distance to be travelled:

```
bal=1, vel=0: time = 2
dist = 2;
bal=1, vel=0: time = 2
```

During implementation, the REGBOT was able to start off at a balanced state and then move a certain distance. However, when it was supposed to come to a stop and balance on the spot, the REGBOT was observed to decelerate abruptly which caused it to lose balance.

Another issue faced was that REGBOT was beginning to steer while moving which could be possibly due to the loosening of the wheel from the motor shaft. From this response, we could see that the system was not as robust in terms of maintaining its stability as compared to the Simulink simulation where it could stop relatively quickly with minimal overshoot and undershoot. This can be attributed to varying reasons such as disturbances and external forces such as friction depending on the surface used, which can lead to unconsidered inputs and errors that the simulation did not face, and hence unexpected performance of the REGBOT.

Conclusion

In conclusion, the team has adopted various controllers and filters into the Matlab/Simulink model to control each portion of the system respectively. The end product for the simulation shows a sufficiently responsive system with no steady state error albeit not fully reflected in the real-world REGBOT system due to aforementioned reasons. Nevertheless, this modelled system serves as a basis for future improvement for the REGBOT controller to achieve its desired performance.