STUDENT C- JORDAN- HEADING ANGLE CONTROL

The “Roomba” style robot is very interesting since it is extremely mobile. While cars and bicycles can only move along paths defined by the steering radius, a roomba has the ability to rotate in place, giving it much more freedom in its path planning. Thinking about the use case, this makes a lot of sense, since roombas are typically deployed in small home environments, where there is not sufficient room to turn with such a large radius.

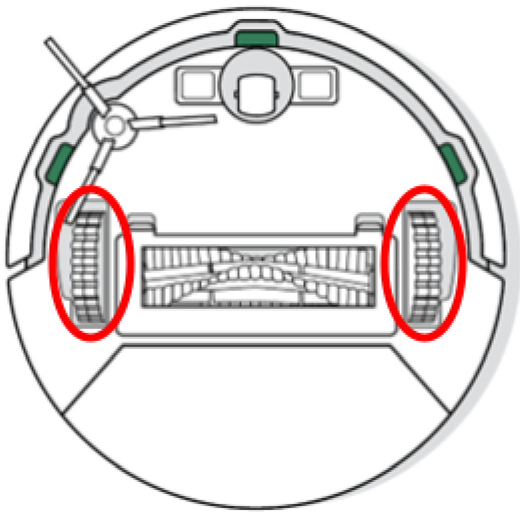


Figure 1
Source: Adapted from [4]

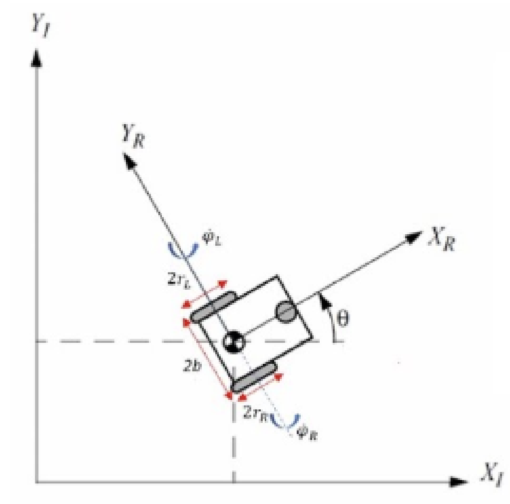


Figure 2
Source: Adapted from [3]

The magic behind this spin-in-place mechanism is the differential drive system. As seen in Figure 1, the robot has 2 driven wheels. By rotating them in opposite directions, the roomba is able to spin in place, adjusting its heading angle directly. By rotating the wheels in the same direction at different speeds, the robot can robot and translate simultaneously, opening up the door for smooth paths that are more optimal. We can model the roomba with the following coordinate frames and variables shown in Figure 2.

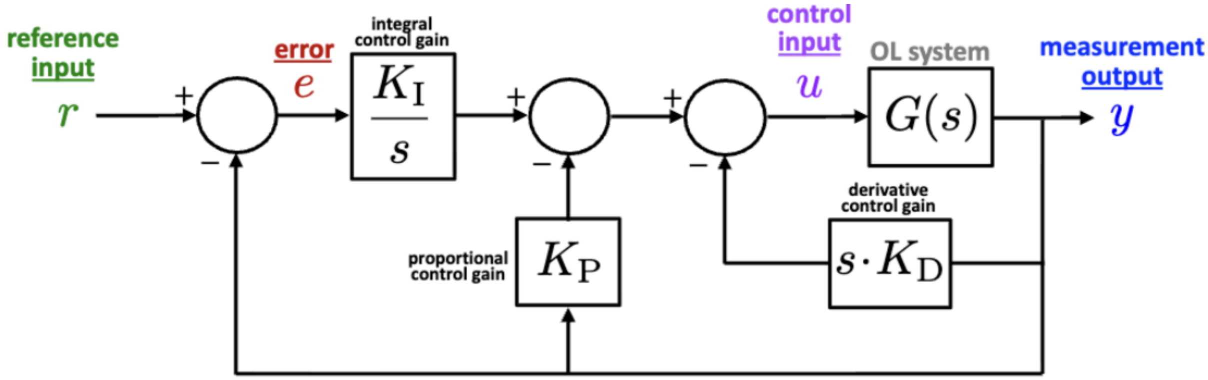
For our analysis of the heading angle controller, the most important equation is

$$\frac{d\theta}{dt} = \frac{r}{2b}(\omega_R - \omega_L)$$

where r is the radius of the wheels, 2b is the base diameter of the roomba, and ω_R and ω_L are the angular velocities of the right and left wheels respectively. For convenience, we will create a variable u(t)= $\omega_R - \omega_L$ for the control input. Taking the Laplace transform of this equation and solving for the transfer function we get

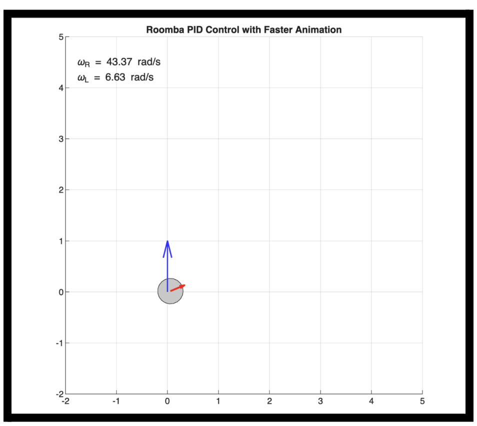
$$G(s) = \frac{\Theta}{U} = \frac{r}{2b} \cdot \frac{1}{s}$$

, which fully encapsulates the dynamics of this system. Next, we need to address the controller. The roomba uses a PID controller to decrease the response time of the system and ensure there is zero steady state error. The block diagram might look something like this.



The only small difference is that this block diagram is completely isolated from the localization step discussed earlier in the report. It assumes that the measurement output is completely accurate, which in reality, it is not. In a more accurate version of this block diagram, the measured output would lead into the localization block diagram. The output of that block diagram, a more correct estimate of the pose of the robot, would then be fed back into this diagram to compute the error.

To better understand this system, take a look at this matlab animation.



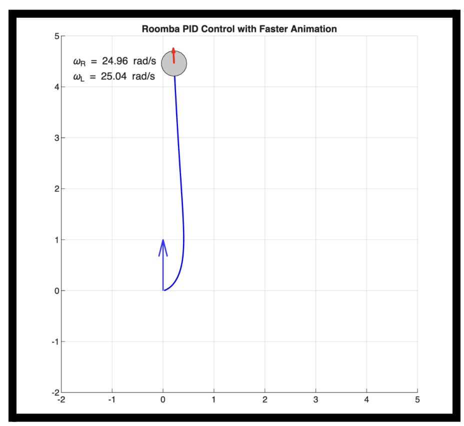


Figure 3 Figure 4

In Figure 3, the roomba is very far away from its target angle. As a result, the right wheel velocity is much higher, which will cause the robot to rotate to the left. In Figure 4, the roomba is pretty much aligned with the desired heading angle, so the velocities are very similar. This is the proportional part of the controller in action! Additionally, take note of how the robot first overshoots the target angle and then comes back– that proves that this is a higher order system. The I and D parts of the controller each add a pole. Lastly, all of this happens while the roomba is in motion, so no time is wasted. So cool!