December 28, 2022

Differential Swerve State Space Controller Design

Henry LeCompte
FRC Team 2383, The Ninjineers

Contents

1 Introduction

Swerve is a type of drivetrain that is characterized by having at least two wheels that can each rotate and spin independently of each other (here rotate is define as a rotation around the robots up axis and spin is defined as a rotation around the modules y axis). This allows the robot to move in any direction and turn in place. Differential Swerve is a type of swerve drivetrain in which the rotation of the wheel is controlled by the difference of the motor velocities and the spin if the wheel is controlled by the average of the motor velocities. This allows for the combination of the torque produced by both motors when driving and turning. This is the main advantage over conventional (coaxial) swerve drivetrains in which rotation and spin are controlled by separate motors. The main disadvantage of differential swerve is that it is significantly more complex to control.

2 Designing the State Space Model

We need to design a continuos, time-invariant state space model of the system. This will allow us to estimate the response of the system to create a feedforward and feedback controller. The following is the general form of a state space model.

$$\dot{x} = Ax + Bu \tag{1}$$

$$y = Cx + Du \tag{2}$$

2.1 Inputs, Outputs, and States

The States will be in the form of

$$x = \begin{bmatrix} v_t \\ v_b \\ \theta \end{bmatrix} \tag{3}$$

Where θ is the angle of the wheel andv_t and v_b are the top and bottom motor velocities respectively.

The velocity of the wheel is not included as a state because it is a linear combination of the motor velocities so it can be calculated from the current state.

The inputs will be in the form of

$$u = \begin{bmatrix} V_t \\ V_b \end{bmatrix} \tag{4}$$

Where V_t is the top motor voltage and V_b is the bottom motor voltage.

Our output matrix is the same as the state matrix as we can measure all states.

$$y = \begin{bmatrix} v_t \\ v_b \\ \theta \end{bmatrix} \tag{5}$$

Where v_t is the top motor velocity, v_b is the bottom motor velocity, and θ is the angle of the wheel.

2.2 DC Motor Model

We know that a permanent magnet DC motor follows the general equation of

$$V = K_v \dot{x} + K_a \ddot{x} \tag{6}$$

And we can rewrite this as

$$\begin{aligned} V &= K_v \dot{x} + K_a \ddot{x} \\ V - K_a \ddot{x} &= K_v \dot{x} \\ -K_a \ddot{x} &= K_v \dot{x} - V \\ \ddot{x} &= \frac{-K_v \dot{x} + V}{K_a} \\ \ddot{x} &= \frac{-K_v \dot{x}}{K_a} + \frac{V}{K_a} \end{aligned}$$

We can also substitute v as $\dot{x}$ to create

$$\dot{v} = \frac{-K_v v}{K_a} + \frac{V}{K_a} \tag{7}$$

This equation can then be written in state space form as

$$\dot{x} = \begin{bmatrix} \frac{-K_v}{K_a} \end{bmatrix} x + \begin{bmatrix} \frac{1}{K_a} \end{bmatrix} u \tag{8}$$

$$y = 1x + 0u \tag{9}$$

2.3 A and B Matrices

Now that we know how to calculate the angular velocity of a motor based on its constants and the input voltage we can start to create the formulas needed to compute the different velocity components of the system

$$\dot{v_t} = \frac{-K_{v_{drive}}}{K_{a_{drive}}} v_t + \frac{1}{K_{a_{drive}}} V_t \tag{10}$$

$$\dot{v_b} = \frac{-K_{v_{drive}}}{K_{a_{drive}}} v_b + \frac{1}{K_{a_{drive}}} V_b \tag{11}$$

$$\dot{\theta} = \frac{v_t + v_b}{2} * K_{turn\_ratio} \tag{12}$$

We can now rewrite these equations as a system that has coefficients for every state and input

$$\dot{v_t} = \frac{-K_{v_{drive}}}{K_{a_{drive}}} v_t + 0v_b + 0\theta + \frac{1}{K_{a_{drive}}} V_t + 0V_b \tag{13}$$

$$\dot{v_b} = 0v_t + \frac{-K_{v_{drive}}}{K_{a_{drive}}} v_b + 0\theta + 0V_t + \frac{1}{K_{a_{drive}}} V_b \tag{14}$$

$$\dot{v_w} = -\frac{K_{v_{drive}}}{2K_{a_{drive}}} v_t + \frac{K_{v_{drive}}}{2K_{a_{drive}}} v_b + 0\theta + \frac{1}{2K_{a_{drive}}} V_t - \frac{1}{2K_{a_{drive}}} V_b \tag{15}$$

$$\dot{\theta} = \frac{K_{turn\_ratio}}{2} v_t + \frac{K_{turn\_ratio}}{2} v_b + 0\theta + 0V_t + 0V_b \tag{16}$$

Now that we have these linear equations we can turn them into the A and B matrix

$$A = \begin{bmatrix} \frac{-K_{v_{drive}}}{K_{a_{drive}}} & 0 & 0 \\ 0 & \frac{-K_{v_{drive}}}{K_{a_{drive}}} & 0 \\ \frac{K_{turn\_ratio}}{2} & \frac{K_{turn\_ratio}}{2} & 0 \end{bmatrix} \tag{17}$$

$$B = \begin{bmatrix} \frac{1}{K_{a_{drive}}} & 0 \\ 0 & \frac{1}{K_{a_{drive}}} \\ 0 & 0 \end{bmatrix} \tag{18}$$

2.4 C and D matrices

With these, we just need the C and D matrices. Because the output matrix is the same as the state matrix C is just the identity matrix.

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{19}$$

$$D = 0 \tag{20}$$

2.5 Full model

And now will all of these matrices we can write the full continuos model as

$$\dot{\begin{bmatrix} v_t \\ v_b \\ \theta \end{bmatrix}} = \begin{bmatrix} \frac{-K_{v_{drive}}}{K_{a_{drive}}} & 0 & 0 \\ 0 & \frac{-K_{v_{drive}}}{K_{a_{drive}}} & 0 \\ \frac{K_{turn\_ratio}}{2} & \frac{K_{turn\_ratio}}{2} & 0 \end{bmatrix} \begin{bmatrix} v_t \\ v_b \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{1}{K_{a_{drive}}} & 0 \\ 0 & \frac{1}{K_{a_{drive}}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_t \\ V_b \end{bmatrix} \tag{21}$$

$$\begin{bmatrix} v_t \\ v_b \\ \theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_t \\ v_b \\ \theta \end{bmatrix} + 0 \begin{bmatrix} V_t \\ V_b \end{bmatrix} \tag{22}$$

Now we can also add the following constants to the model (These will be different for each robot)

$$K_{v_{drive}} = 0.023 \tag{23}$$

$$K_{a_{drive}} = 0.001 \tag{24}$$

$$K_{turn\_ratio} = \frac{1}{28} \tag{25}$$

3 Validating the State Space Model

Now that we have a state space model we need to validate that it is controllable, observable and stable.

3.1 Controllability

Controllability is a property of the system that allows any initial state to be driven to any desired state using control inputs in a finite time. The Controllability matrix for continuos time-invariant systems is defined as

$$R = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \tag{26}$$

And the system is said to be controllable if the rank of the controllability matrix is equal to the number of states (n).

$$rank(R) = n \tag{27}$$

We can now calculate the rank of the controllability matrix for our system and find that it is indeed 3. This means that out system is indeed controllable and we can move on to the next step.

3.2 Observability

A system is said to be observable if, for every possible state, the current state can be estimated using only the information from outputs. The observability matrix for continuos time-invariant systems is defined as

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{28}$$

And the system is said to be observable if the rank of the observability matrix is equal to the number of states (n).

$$rank(O) = n \tag{29}$$

Calculating this for our current system we find that the rank is indeed 3. This means that our system is observable and we can move on to the next step.

3.3 Stability

A system is said to be stable if the eigenvalues of the A matrix are all negative. Calculating the eigenvalues of the A matrix we can find that the real parts of the eigenvalues are 0 and -23. This means that our system is marginally stable but we can fix this with the LQR controller in the next section.

3.4 Step Response

To validate that our system is stable we can also look at the step response of the system. The step response is the response of the system to a step input. This can be calculated with the python package control. We can graph the step response values for the system and find that the system is indeed stable. This graph can be seen in figure 1.

3.5 Forced Response

We also want to make sure that the wheel is turning correctly and is still stable with different inputs. To do this we can create a graph of forced inputs. This graph uses 5 time steps of 12 volts on each motor. 495 more time steps of 12 and -12 volts and 500 more time steps of no voltage. This graph can be seen in figure 2.

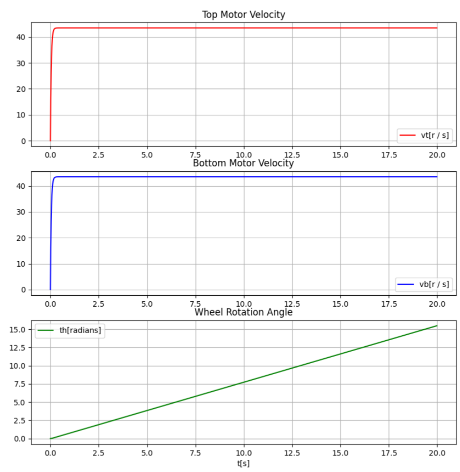

Figure 1: Step response of the system

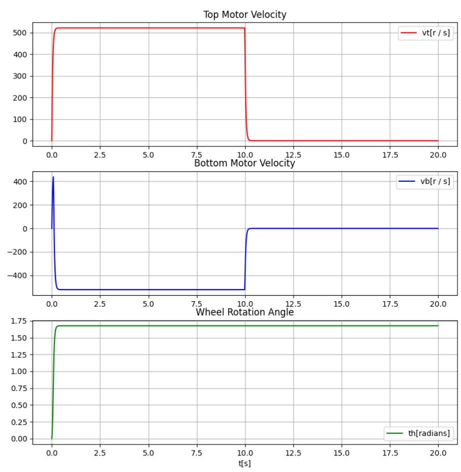

Figure 2: Forced response of the system