Olivia Ball
ECE 516
8 April 2020

Milestone 1

Introduction

The goal of this assignment was to derive a 7-state model and a 5-state model for an unmanned vehicle (UV) that consists of two driving wheels and one caster wheel. The overall system dynamics can be represented through a series of first-order differential equations. Since *each* wheel is driven by a DC motor, we will need to observe the dynamics of both wheels (right and left). Throughout the assignment the right and left wheels will be designated with R and L subscripts, respectively.

The first-order differential equations provide us with a way to describe a system at a particular point in time based off of some initial t_0. The system will be described by a set of variables, called state variables (x_i), which are used to determine the state of the system at any fixed time. The *n*-states provide the number of state variables (n) needed in the state vector, x(t), which is represented as $x = [x_1\ x_2\ \cdots\ x_n]^T$. The system will also be comprised of a number, *r*, real-valued inputs represented as u_i(t). These inputs describe an $r \times 1$ input vector known as u(t). The output system will be comprised of a number, *m*, real-valued outputs represented as y_i(t). These outputs describe an $m \times 1$ output vector known as y(t).

The system will be observed over a time of interest, T, once the system has a well-defined state vector, input vector and output vector. Over a set time, these vectors provide unique elements at any given time to provide the state space (Σ), input space (U) and output space (Y). The following equation provides a state-space notation in terms of a set of first-order differential equations and a set of algebraic equations:

$$\dot{x} = f(x, u, t)$$
$$y(t) = h(x, u, t)$$

The state space representation is described by the two equations below: the state equations and the output equations. The state equations are on top and are made with the first-order differential equations where: *A(t)* is an $n \times n$ square matrix that describes the constant coefficients of the system and *B(t)* is an $n \times r$ matrix where the coefficients describe the weight of each input variable. The output equations are on the bottom and are made up of algebraic equations where: *C(t)* is an $n \times m$ matrix where the coefficients describe the state variables of interest and *D(t)* is a null matrix which can be neglected for a UV system. The *x* variable represents the state vector, which is of length *n*, the *u* variable represents the input vector, which is a length of *r,* and the *y* variable represents the output vector, of length *n*.

$$\dot{x} = A(t)x(t) + B(t)u(t)$$
$$y(t) = C(t)x(t) + D(t)u(t)$$

A numerical solution for the system dynamics can be achieved by implementing these state and output equations in Simulink. The specific UV system parameters will first be set in a MatLAB code that will not be changed. These values can then be used in the Simulink model to formulate each differential equation by using a variety of different blocks such as - integrator, summing junction, multiplier, function generator and gain. The Simulink model will provide a way to quickly test how different inputs affect the systems outputs. By using all of this information, the 7-state model and 5-state model can be derived and implemented using Simulink. The results of the Simulink simulations will be used to compare and show the results of both models.

Problem Formulation

The first step to solving for a 7-state model and 5-state model is to determine the inputs, state variables and outputs that will be most descriptive which is described in detail below:

The 7-state model will consist of 7 state variables: x_I, y_I, θ, ω_R, ω_L, i_R and i_L. These state variables were chosen due to their insight about the position within the Cartesian inertial coordinate (x_I, y_I, θ), the mechanical properties of the DC motor as they pertain to each wheel (ω_R, ω_L) and the electrical properties of the DC motor as they pertain to each wheel (i_R, i_L). The inputs of the system are voltages for each of the wheels: u_R and u_L. These inputs drive the DC motor which will allow us to find the armature winding current in each wheel (i_R, i_L) and the angular velocity in each wheel (ω_R, ω_L). The outputs will be the position in the Cartesian inertial coordinate (x_I, y_I) and the azimuth angle (θ). These outputs provide us with a way we can graph where the UV is and how it has moved over time.

The 5-state model will consist of 5 state variables: x_I, y_I, θ, ω_R and ω_L. Since we are considering the system dynamics over 10 seconds, we can neglect the UV electrical dynamics since they are in the order of milliseconds. This allows for the electrical dynamics to be set equal to zero. The inputs and outputs of this equation will remain the same as the 7-state model.

Once the state variables, input variables and output variables have been defined, we need to define differential equations in terms of each state variable. The differential position equations are provided for $\dot{x}_I, \dot{y}_I$ and $\dot{\theta}$ and do not need any modifications. The DC motor angular velocity differential equation describes the mechanical dynamics of the motor. Since we are modeling the system with the load conditions, the parameters are replaced with their equivalent terms ($J \rightarrow J_{eq}$, $B \rightarrow B_{eq}$ and $K_e \rightarrow K_{eq}$). These equivalent terms allow for the differential equation subscripts to be changed to the corresponding R and L terms for each wheel to provide two more differential equations $\dot{\omega}_R$ and $\dot{\omega}_L$. The DC motor winding current differential equation provides the electrical dynamics of the motor. In order to describe the differential equation in terms of each wheel, we need to provide a way to relate motor speed and wheel speed. The extra step needs to be taken since we are not using equivalent terms. The gear ratio can be used to provide us with an equivalent ratio. After making this substitution, all other subscripts can be changed to their corresponding R and L subscripts to give us two more differential equations $\dot{\iota}_R$ and $\dot{\iota}_L$. These are the 7 differential equations that will be modeled in Simulink.

Just like the state variables, reduce the 7-state differential equations down to the 5-state differential equations by eliminating the electrical state variables. To do this, rearrange the electrical differential equation to get the $i_{R,L}$ state variables in terms of the $\omega_{R,L}$ state variables. Once the relationship is achieved substitute the $i_{R,L}$ in the angular velocity differential equations. The system will be modeled with the same position differential equations and modified angular velocity differential equations ($\dot{x}_I, \dot{y}_I$, $\dot{\theta}$, $\dot{\omega}_R$, $\dot{\omega}_L$).

In order to create the 7-state model in Simulink, we will need a MatLAB code that contains all the constants that make up these equations and a combination of blocks as mentioned before. The system will have 3 subsections: input, DC Motor Dynamics and Robot Dynamics. The input subsection will be controlled in the MatLAB code and have specific voltage values for each wheel. The system will be driven by a step input. The DC Motor will utilize the inputs from the input subsection to make the $\dot{\iota}_R$, $\dot{\iota}_L$, $\dot{\omega}_R$ and $\dot{\omega}_L$ differential equations. The output of each integrator will be our state variables and provide us with the value of each element over the desired time. The ω_R and ω_L state variables will become the inputs for the robot subsystem. These will be used to create the differential equations for $\dot{x}_I, \dot{y}_I$ and $\dot{\theta}$ where the output of the integrator will be the corresponding state variables. The 5-state model in Simulink, will only require changes to the DC motor dynamics subsystem to account for the new differential equation mentioned above. The DC motor subsystem will only contain two outputs which are the state variables ω_R and ω_L which still serve as the inputs for the robot subsystem.

Mathematical Modeling

7-State Model: Described by differential equations 1-7

After using the state-space representation to solve for v (linear velocity) and α (linear acceleration) we can describe the position in the Polar Cartesian coordinates by the following differential equations:

(1) $\dot{x}_I = \frac{\rho}{2}(\omega_L + \omega_R) * cos\theta$

(2) $\dot{y}_I = \frac{\rho}{2}(\omega_L + \omega_R) * sin\theta$

(3) $\dot{\theta} = \frac{\rho}{W_r}(-\omega_L + \omega_R)$

The DC motor mechanical and electrical dynamics are described below after replacing K_{eq}, B_{eq} and J_{eq}, switching the subscripts to R and L for each wheel, and using the assumption $\omega_{motor} = N_{motor} * \omega_{wheel(R\ or\ L)}$ to provide the following differential equations:

(4) $\dot{\omega}_R = \frac{K_{eq}}{J_{eq}} i_R - \frac{B_{eq}}{J_{eq}} \omega_R$

(5) $\dot{\omega}_L = \frac{K_{eq}}{J_{eq}} i_L - \frac{B_{eq}}{J_{eq}} \omega_L$

(6) $\dot{i_R} = -\frac{R_a}{L_a} i_R - \frac{K_b}{L_a} N_{motor}\omega_R + \frac{1}{L_a} u_R$

(7) $\dot{i_L} = -\frac{R_a}{L_a} i_L - \frac{K_b}{L_a} N_{motor}\omega_L + \frac{1}{L_a} u_L$

5-State Model: Described by differential equations 1-3 and 8-9

The assumption that the electrical dynamics are zero provide us with the following equation with the goal of finding i_R and i_L in terms of ω_R and ω_L respectively:

$$\frac{1}{L_a} u_a = \frac{R_a}{L_a} i_a + \frac{K_b}{L_a} \omega_{motor} \rightarrow\ u_a = R_a i_a + K_b \omega_{motor}$$

$$i_a = \frac{u_a}{R_a} - \frac{K_b}{R_a} \omega_{motor}$$

Plug this into $i_{R,L}$ state variable from equations 4 and 5 above to get following generalized equation (a subscript):

$$\dot{\omega}_a = \frac{K_{eq}}{J_{eq}}\left(\frac{u_R}{R_a} - \frac{K_b}{R_a} N_{motor}\omega_a\right) - \frac{B_{eq}}{J_{eq}}\omega_a \rightarrow\ \dot{\omega}_a = \frac{K_{eq}}{J_{eq}R_a} u_a + (-\frac{B_{eq}}{J_{eq}} - \frac{K_{eq}K_b}{J_{eq}R_a} N_{motor})\omega_a$$

$$\dot{\omega}_a = \frac{K_{eq}}{J_{eq}R_a} u_a - (\frac{B_{eq}R_a + K_{eq}K_b N_{motor}}{J_{eq}R_a})\omega_a$$

Replace the general a subscript with respective R and L:

(8) $\dot{\omega}_R = \frac{K_{eq}}{J_{eq}R_a} u_R - \left(\frac{B_{eq}R_a + K_{eq}K_b N_{motor}}{J_{eq}R_a}\right)\omega_R$

(9) $\dot{\omega}_L = \frac{K_{eq}}{J_{eq}R_a} u_L - (\frac{B_{eq}R_a + K_{eq}K_b N_{motor}}{J_{eq}R_a})\omega_L$

Simulation

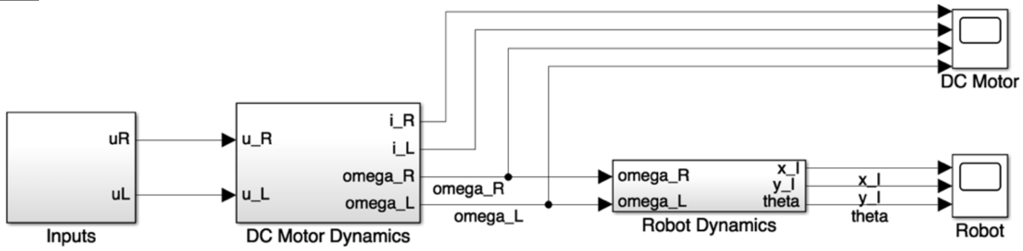

Figure 1: Overall simulation for UV dynamics 7-state model

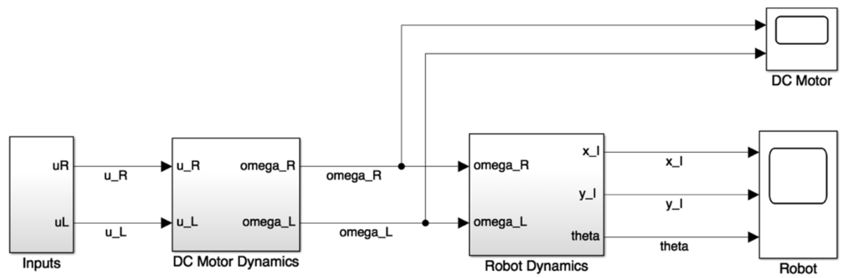

Figure 2: Overall simulation for UV dynamics 5-state model

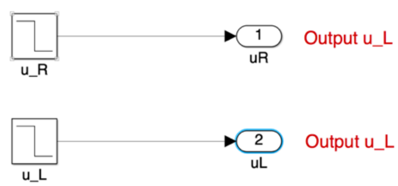

Figure 3: Inputs Subsystem for 7-state model and 5-state model

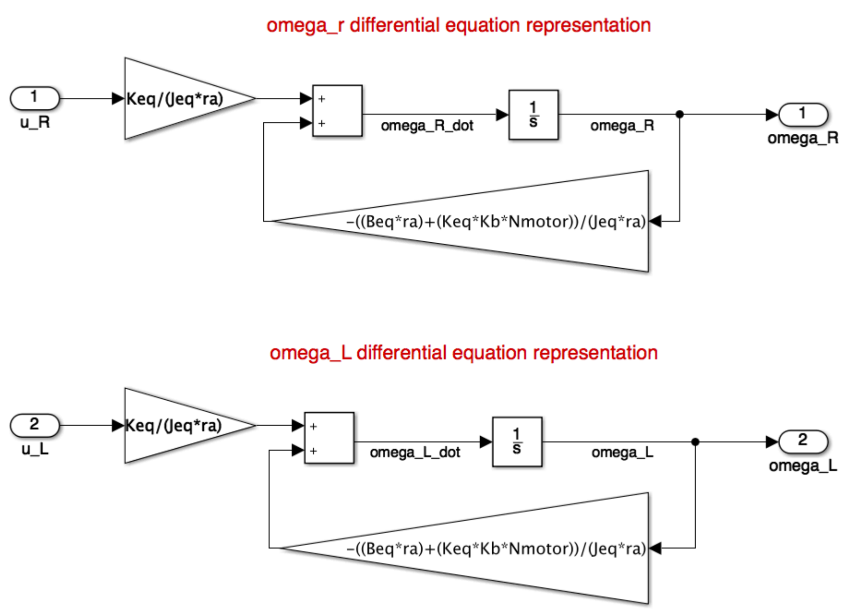

Figure 4: DC Motor Dynamics Subsystem for 5-state model

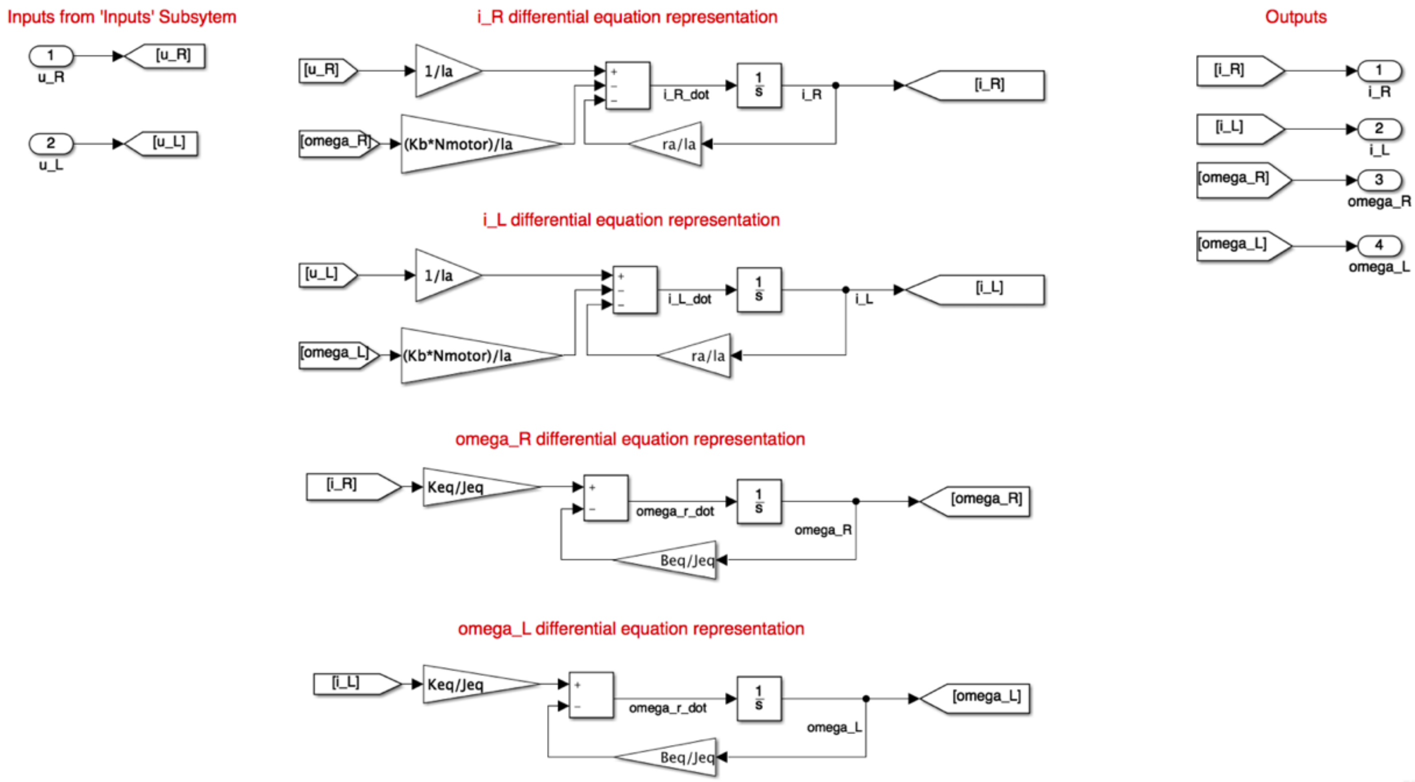

Figure 5: DC Motor Dynamics Subsystem for 7-state model

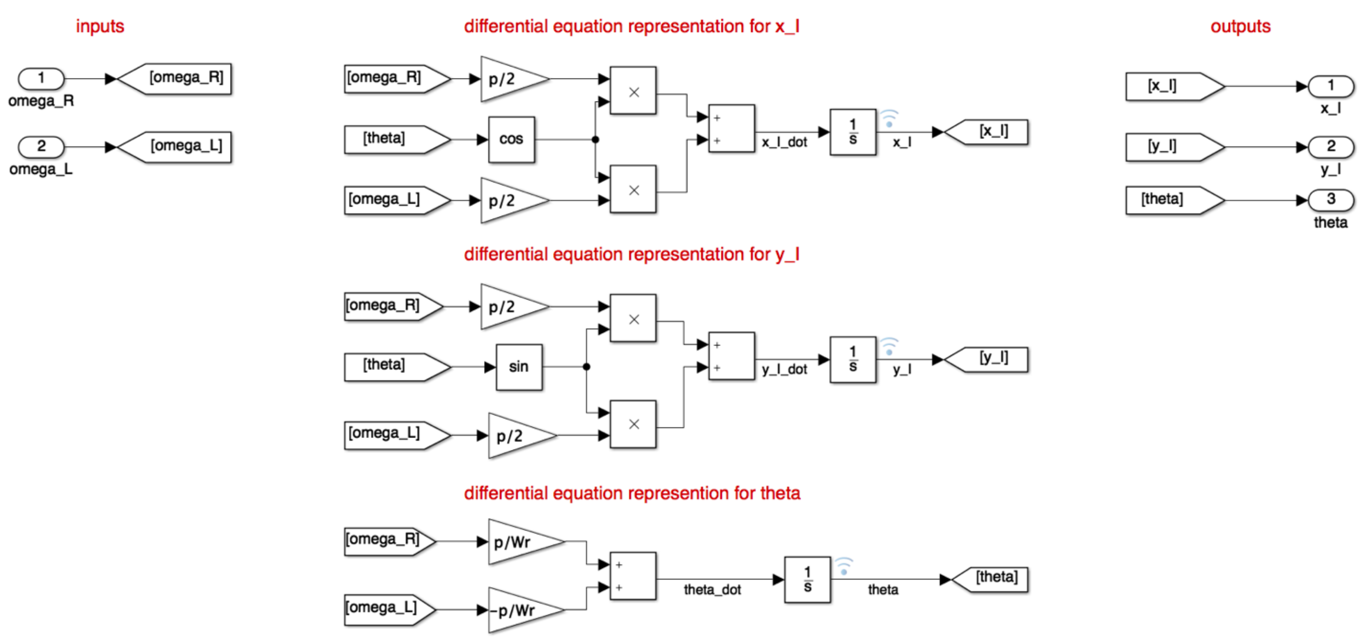

Figure 6: Robot Dynamics Subsystem for 7-State Model and 5-State Model

Outputs: 7-State Model

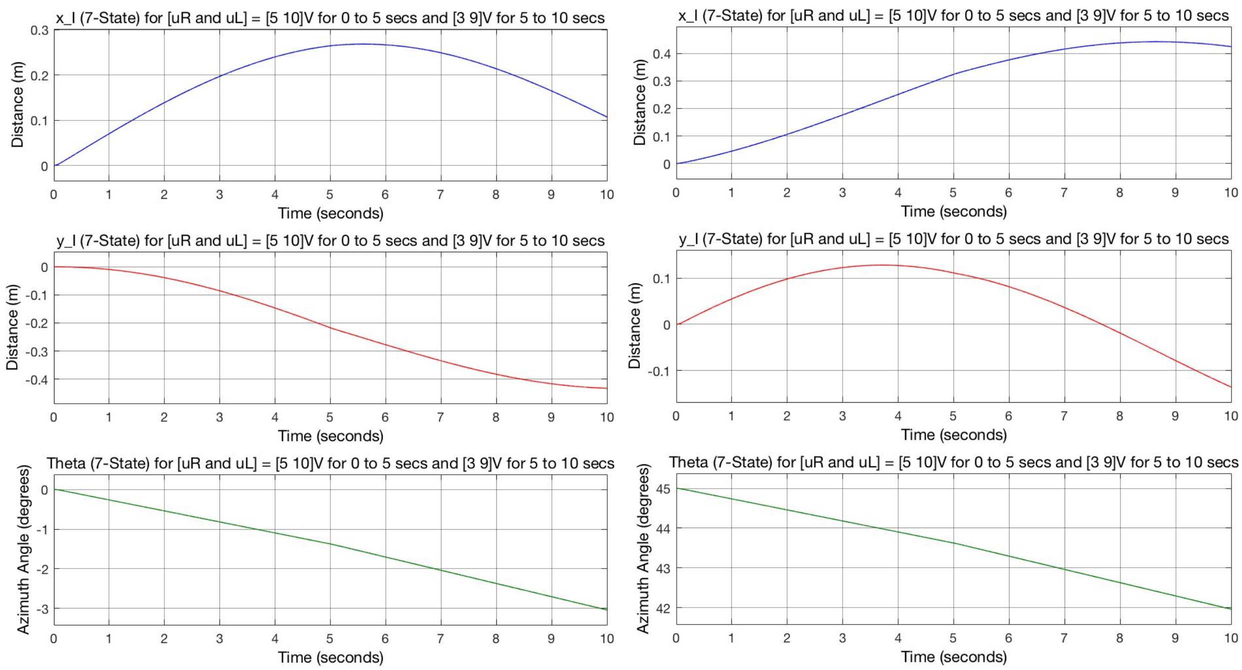

Figure 7: EV dynamics of a system with inputs, $u_L > u_R$, as stated in each title each graph (Left) initial state values are zero for all state variables (Right) initial state values set as $x_i = 0$, $y_I = 0$, $\theta=45$, $i_L=0.4$, , $i_L=0.4$, $\omega_R=0$ and $\omega_L = 0$.

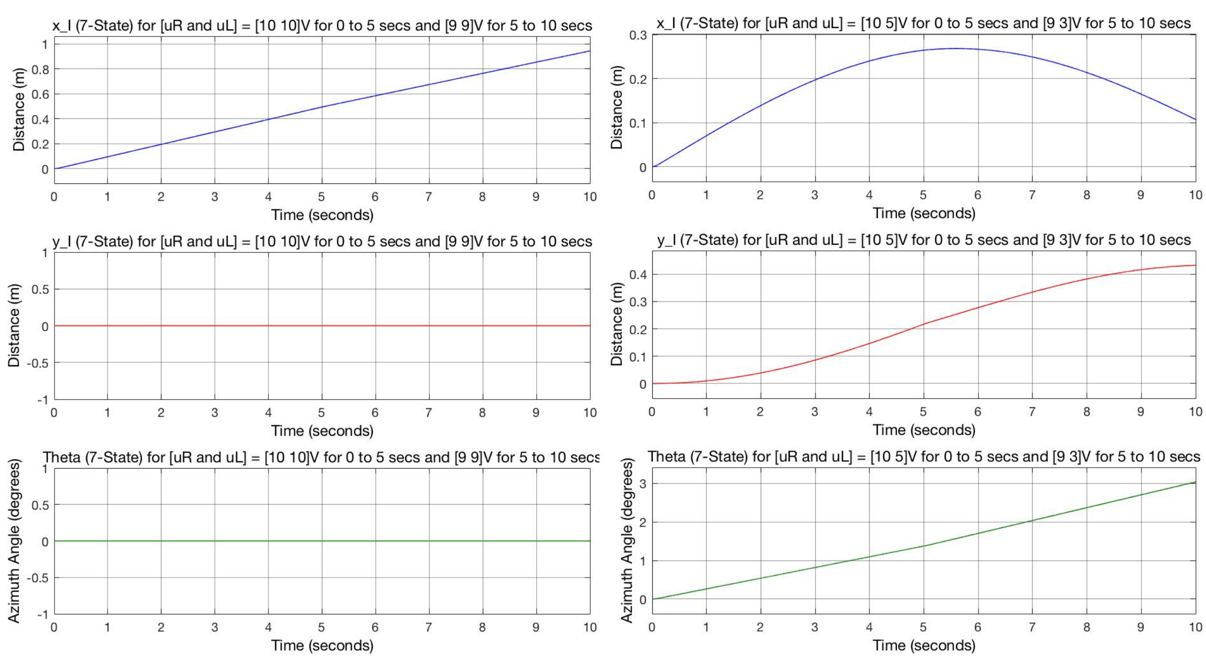

Figure 8: Both graphs have all initial state values equal to zero for all state variables but have varying input variables (Left) $u_L=u_R$ (Right) $u_L<u_R$

Outputs: 5-State Model

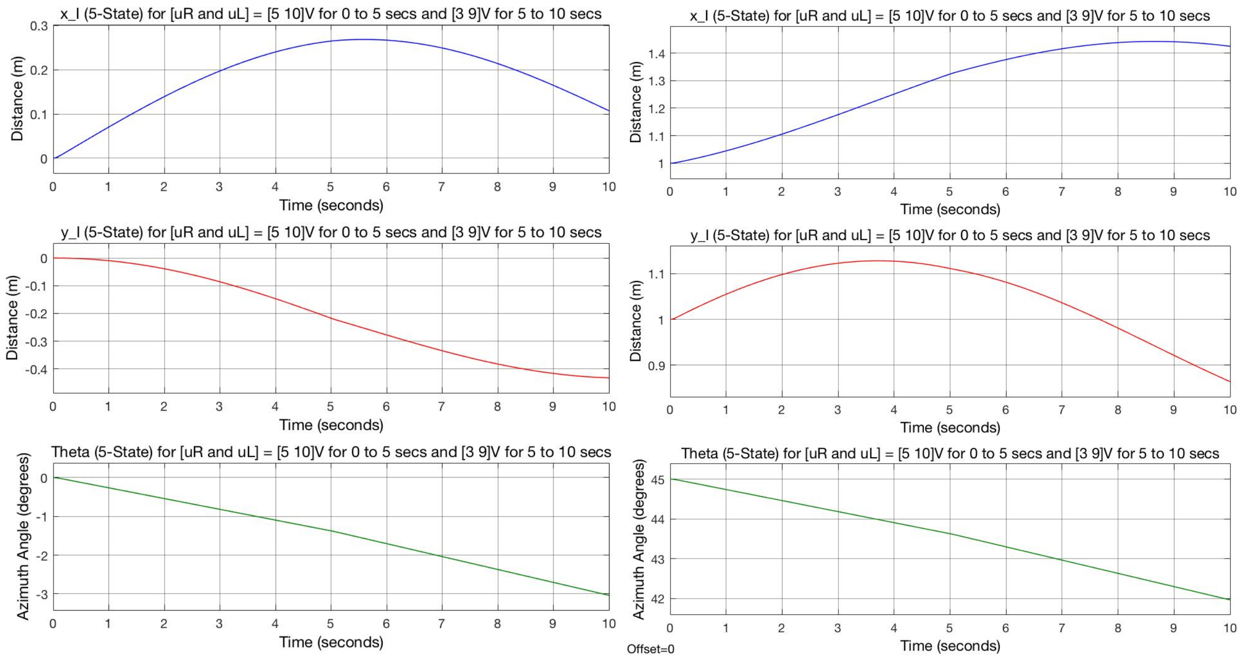

Figure 9: : EV dynamics of a system with inputs, $u_L > u_R$, as stated in each title each graph (Left) initial state values are zero for all state variables (Right) $x_i = 1$, $y_I = 1$, $\theta=45$, $\omega_R=0$ and $\omega_L = 0$

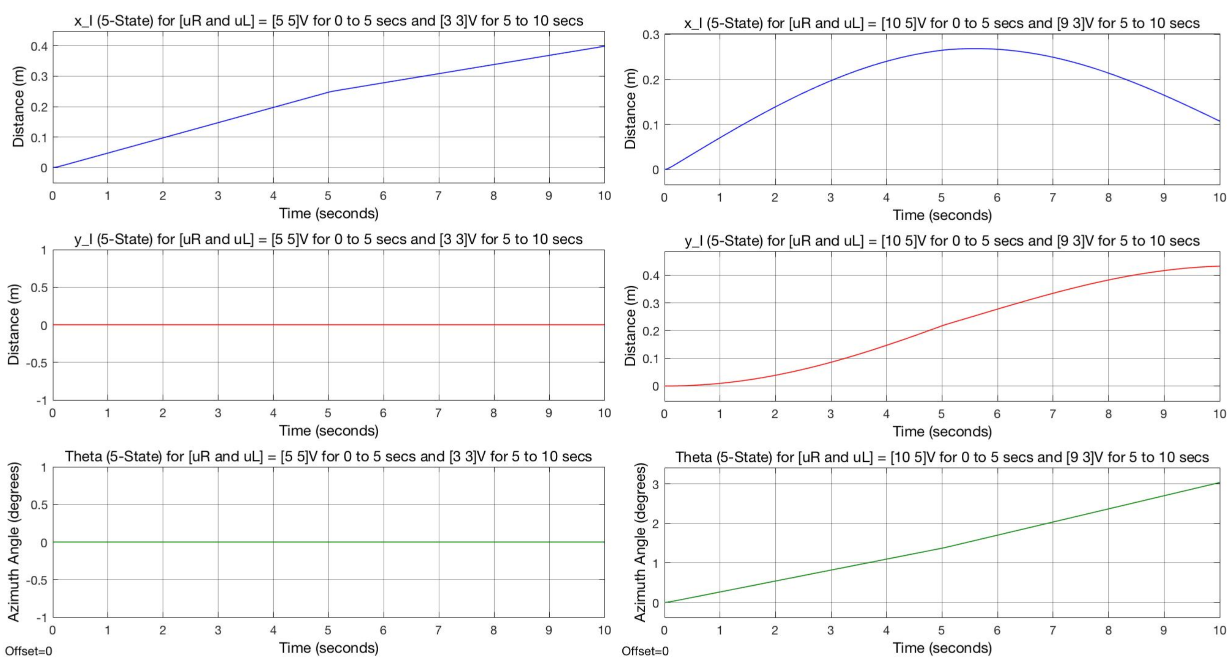

Figure 10: Both graphs have all initial state values equal to zero for all state variables but have varying input variables (Left) $u_L = u_R$ (Right) $u_L < u_R$

Discussion

The differential equations used to create the 7-state space model, can be used to form the following spate space representation. The *state space* will consist of all elements of the state vector (x_I, y_I, θ, ω_R, ω_L, i_R and i_L) over our desired time, the *input space* will consist of all elements of the input vector (u_R and u_L) over desired time and the *output space* will consist of all elements of the output vector (x_I, y_I and θ) over the desired time:

$$\begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta} \\ \dot{\omega}_R \\ \dot{\omega}_L \\ \dot{i}_R \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \frac{\rho}{2}\cos(\theta) & \frac{\rho}{2}\cos(\theta) & 0 & 0 \\ 0 & 0 & 0 & \frac{\rho}{2}\sin(\theta) & \frac{\rho}{2}\sin(\theta) & 0 & 0 \\ 0 & 0 & 0 & -\frac{\rho}{W_r} & 0 & \frac{K_{eq}}{J_{eq}} & -\frac{\rho}{W_r} \\ 0 & 0 & 0 & -\frac{B_{eq}}{J_{eq}} & 0 & \frac{K_{eq}}{J_{eq}} & 0 \\ 0 & 0 & 0 & 0 & -\frac{B_{eq}}{J_{eq}} & 0 & \frac{K_{eq}}{J_{eq}} \\ 0 & 0 & 0 & -\frac{K_b}{L_a}N_{motor} & 0 & -\frac{R_a}{L_a} & 0 \\ 0 & 0 & 0 & 0 & -\frac{K_b}{L_a}N_{motor} & 0 & -\frac{R_a}{L_a} \end{bmatrix} \begin{bmatrix} x_I \\ y_I \\ \theta \\ \omega_R \\ \omega_L \\ i_R \\ i_L \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ \frac{1}{L_a} \end{bmatrix} \begin{bmatrix} u_R \\ u_L \end{bmatrix}$$

$$[y] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_I \\ y_I \\ \theta \\ \omega_R \\ \omega_L \\ i_R \\ i_L \end{bmatrix}$$

The differential equations used to create the 5-state space model, can be used to form the following spate space representation. The *state space* will consist of all elements of the state vector (x_I, y_I, θ, ω_R and ω_L) over the desired time, the *input space* will consist of all elements of the input vector (u_R and u_L) over the desired time and the *output space* will consist of all elements of the output vector (x_I, y_I and θ) over the desired time:

$$\begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{\theta} \\ \dot{\omega}_R \\ \dot{\omega}_L \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \frac{\rho}{2}\cos(\theta) & \frac{\rho}{2}\cos(\theta) \\ 0 & 0 & 0 & \frac{\rho}{2}\sin(\theta) & \frac{\rho}{2}\sin(\theta) \\ 0 & 0 & 0 & \frac{\rho}{W_r} & -\frac{\rho}{W_r} \\ 0 & 0 & 0 & \frac{B_{eq}R_a + K_{eq}K_bN_{motor}}{J_{eq}R_a} & 0 \\ 0 & 0 & 0 & 0 & \frac{B_{eq}R_a + K_{eq}K_bN_{motor}}{J_{eq}R_a} \end{bmatrix} \begin{bmatrix} x_I \\ y_I \\ \theta \\ \omega_R \\ \omega_L \end{bmatrix} + \begin{bmatrix} \frac{K_{eq}}{J_{eq}R_a} \\ \frac{K_{eq}}{J_{eq}R_a} \end{bmatrix} \begin{bmatrix} u_R \\ u_L \end{bmatrix}$$

$$[y] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_I \\ y_I \\ \theta \\ \omega_R \\ \omega_L \end{bmatrix}$$

The results from implementing our 5-state model and 7-state model in Simulink showed that in many different cases we got the same output for both models. The outputs to each model were the same with varying inputs and initial state variable conditions. By obtaining these results, it confirms that the electrical dynamics of the system were negligible and setting the electrical dynamics equal to zero is valid.

In order to understand the results, we must first understand the initial position of the UV. When the initial state values are x=0 as stated, the position is located at $x_I = 0$ and $y_I = 0$ within the Cartesian inertial coordinates and θ =0 meaning the azimuth angle (orientation) of the robot is zero degrees. The other state values being zero let you know that it is not moving yet. By using the orientation provided to us in the assignment we know that, the center of the car is at the origin, the left wheel is in the positive y_I direction and the right wheel is in the negative y_I direction. These wheels are both the same distance $\frac{W_r}{2}$ away from the origin where W_r is the distance between the two wheels.

The initial condition orientation allows for us to understand how our inputs affect direction of the UV in a clear way. The three different input options are described below with respect to the 5 state model figures since the outputs are the same:

The inputs have larger left wheel values than right wheel values ($u_R < u_L$) which can been seen in figure 9: $[5\ 10]^T$ V for $0 \leq t \leq 5$ seconds & $[3\ 9]^T$ V for $5 < t \leq 10$ seconds. The first value in the vector corresponds to the input for the right wheel (u_R) and the second value in the vector corresponds to the left wheel (u_L). Since the left wheel has a higher input than the right wheel, the expectation is that the UV will start to move in the negative y_I direction since the left wheel will overpower the right wheel and start to twist in a negative θ direction. Additionally, the UV will initially have some movement in the positive x_I direction. The results show exactly what we would expect and show that the UV dynamics over 10 seconds: will only go so far in the positive x_I (0.2684 m) then slowly start back to the origin, goes in the negative y_I (-0.4326 m) and appears to start turning back to x_I and has a negative θ which corresponds to a left turn.

The inputs have the same inputs for the right and left wheels ($u_R = u_L$) which can been seen in figure 10: $[5\ 5]^T$ V for $0 \leq t \leq 5$ seconds & $[3\ 3]^T$ V for $5 < t \leq 10$ seconds. The first value in the vector corresponds to the input for the right wheel (u_R) and the second value in the vector corresponds to the left wheel (u_L). Since the input to both wheels is the same, we are assuming the UV is pointed in the positive x_I direction the system should just go in a positive x_I direction. At the end of 10 seconds, the UV travels 0.40 meters in the positive x_I direction. The system dynamics confirm our assumptions and show a value of zero for y_I and θ. When the UV is only traveling in one direction, we are able to see the affects the input value has on distance traveled. In figure 8, we use the input $[10\ 10]^T$ V for $0 \leq t \leq 5$ seconds & $[9\ 9]^T$ V for $5 < t \leq 10$ seconds. The input to the system is doubled for the first 5 seconds and more than doubled for the next 5 seconds. The UV travels over double the distance and moves in the positive x_I direction 0.94 meters.

The inputs have larger right wheel values than left wheel values ($u_R > u_L$) which can be seen in figure 10: $[10\ 5]^T$ V for $0 \leq t \leq 5$ seconds & $[9\ 3]^T$ V for $5 < t \leq 10$ seconds. The first value in the vector corresponds to the input for the right wheel (u_R) and the second value in the vector corresponds to the left wheel (u_L). Since we used the same values to model when the right wheel has a higher input than the left wheel, the expectation is that system dynamics have the same initial movement in the positive x_I direction. However, since the right wheel is overpowering the left the UV will start to move in the positive y_I direction and start to twist in a positive θ direction. The results show exactly what we would expect and show that the UV dynamics over 10 seconds: will only go so far in the positive x_I (0.2684 m) then slowly start back to the origin, goes in the positive y_I (0.4326 m) and appears to start turning back to x_I and has a positive θ which corresponds to a right turn. The y_I and θ values are exact mirrors while x_I is exactly the same.

The system dynamics were also evaluated when the initial state spaces were not zero which are shown in figures 8 and 10. The only difference the state variables for these two graphs are that x_i & y_I are zero in figure 8 and x_i & y_I are one in figure 10. The results confirmed that changing either of these initial state variables only changes the starting location. The angular velocities for both states, ω_R and ω_L, remained at zero so we could observe differences between the models. In order to determine the max winding current input to the system, was estimated using the max voltage $i_{max} = \frac{v}{R} = \frac{10\ V}{7.4\Omega} = 1.35\ A$. In order to make sure that the initial winding current input was set within the limits both wheels were set at 0.4. Even when the winding input currents are not zero for the 7-state model, the overall models are still identical to one another.

Since both state models were ran over the same simulation time, so in order to compare the two in terms of performance I utilized the performance report provided by Simulink to estimate the number of FLOPS. The number of FLOPS represents the number of floating-point operations per second. A floating-point operation is any mathematical operation (such as the integrator, gain, multiplication and addition/subtraction blocks) where the number is a floating-point number. Using this information. The performance report stated that the 5-state model had a total recording time of 3.41 seconds with a total of 181 blocks methods. Since these block methods represent mathematical operations, I will estimate that the 5-state system has about 53 FLOPS. The performance report for the 7-state mode had a total recording time of 73.94 seconds with a total of 233 block methods. By making the same assumptions, we can determine that the 7-state model had about 3.15 FLOPS.

When a system is in good conditions, it has to have the correct state variables and be able to be modeled in Simulink. Both of the state models meet this requirement, but I would use the 5-state model when the UV is in good conditions. This is due to the fact that both of the models produce the same results for the given UV system but the performance of the 5-state system is significantly faster. Also since we have determined through our results that information about the winding current can be neglected the 5-state model provides us with all the information that we need. The 5-state model provides the smallest set of variables which can be used to determine the state.

If the DC motor is burned out in the middle of the run (t=3 seconds) which causes the inductance and resistance values to drop, I would use the 5-state model. The state variables that include these terms are i_L and i_R which are accounted for in the 5-state model. When you make changes to these values in the MatLAB code, you still get the same output for both models. When the inductance and resistance values are decreased to 0.001 H and 3 ohms, respectively, this causes an increase in the i_L and i_R outputs. Although we can see these in the 7-state model they are still accounted for in the 5-state model. Since the angular velocity is dependent upon the winding current, the increase in the winding current will lead to an increased speed. Since the speed is the input for our position outputs, we will have a variation in the position of our UV. The increase in speed will cause an increase of theta and these variations on theta will cause very different position results for x_I and y_I since they are both functions of theta.

In most situations, I would use the 5-state model. The overall model is simpler than the 7-state model therefore it is easier to implement in Simulink. Due to its simplicity, the model also has a better performance than the other model. Additionally, we are able obtain the same exact output results for speed and position results by this model.

If there was a situation where I wanted to visualize the winding current for the left and right wheels, I would use the 7-state model. The DC motor dynamics subsystem provides these state variables which can be visualized with a scope. Additionally, the 7-state model can be good for showing how inductance specifically is alters these values. The final state space representations presented below show that the 7-state space representation is the only one that includes inductance, L_a.

References

[1] Brogan, W. L. (1974). *Modern Control Theory, Third Edition.* Quantum Publishers.

[2] Chow, M.-Y. (n.d.). ECE 516 System Control Engineering- T02 State Space Descriptions.

[3] El-Badawy, A., & Mowafy, A. (2012). Globally stable integrated kinematic and dynamic control for trajectory tracking of a mobile robot provide with caster wheel. *Journal of Al Azhar University Engineering Sector, 7*(24), 1531-1544.

[4] Rowell, D. (2002). *State-Space Representation of LTI Systems.* MIT.

[5] Shojaei, K., Shakri, A., Tarakameh, A., & Tabibian, B. (2011). Adaptive trajectory tracking control of a differential drive wheeled mobile robot. *Robotica, 29*(03), 391-402.