MEMORANDUM

To: Charlie Refvem, Department of Mechanical Engineering, Cal Poly SLO
crefvem@calpoly.edu

From: Antonio Ventimiglia
ventimig@calpoly.edu

Date: October 19th, 2025

RE: **Homework 0x03: System Modeling**

Developing a dynamic model of the Romi robot, which is still under development, provides critical insight into its expected behaviors and informs effective control strategies. Figure 1 presents an illustration of the Romi platform.

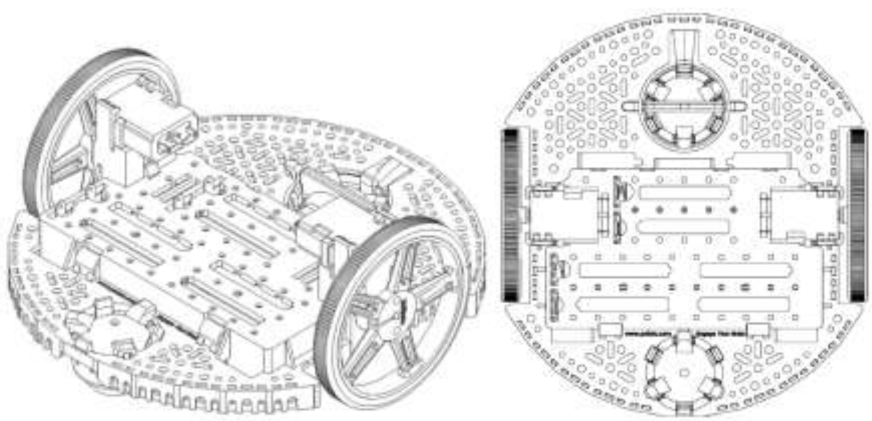

Figure 1. Illustration of Romi Robot

Kinematics

It was chosen that the local x-axis for the Romi will go along its long axis, pointing towards the "front" of the bot. Figure 2 shows the respective velocity vectors associated with the components of the Romi bot with this given local coordinate system. Note the two crucial dimensions provided for the Romi: The radius of the wheel from its axis of rotation to the contact surface, r, and the track width of the wheels, w.

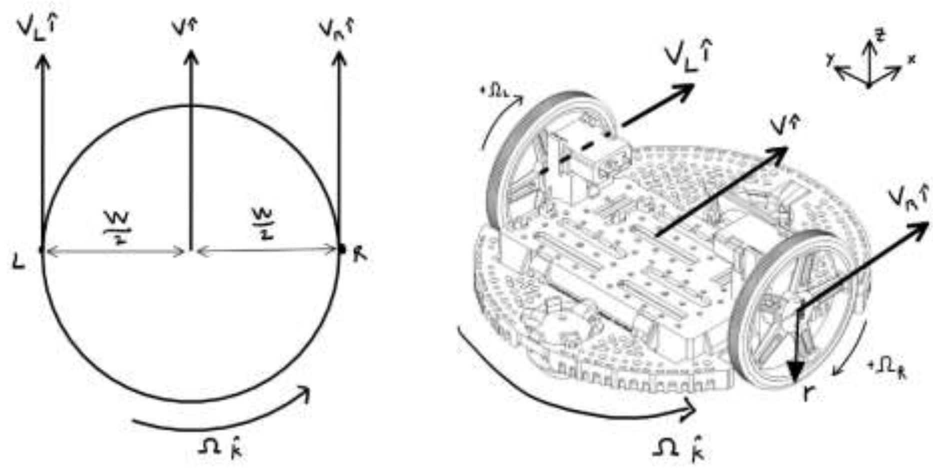

Figure 2. Illustration of Romi kinematics in its local coordinate system

Applying the equations for general plane motion to Figure 2, we can derive the velocity of the right wheel with respect to the left wheel in their local coordinate systems (Equation 1a):

$$V_R\,\hat{\imath} = V_L\,\hat{\imath} + \left(\Omega\,\hat{k}\right) \times (-\mathrm{w}\,\hat{\jmath}) \tag{1a}$$

where V_R and V_L are the translational velocities of the right and left wheels' centers (respectively), Ω is the Romi's yaw rate, and w is the track width. Equation 1a can be arranged into Equation 1b as follows.

$$(\mathrm{V_R} - \mathrm{V_L})\,\hat{\imath} = \mathrm{w}\Omega\,\hat{\imath}$$

$$\Omega = \frac{V_R - V_L}{w} \tag{1b}$$

The translational velocity of each wheel can be expressed as a function of each wheel's angular velocity, assuming there is a no-slip condition at the contact point between the wheel and the ground. This assumption will be discussed later. This relation is as follows:

$$V_{center} = \Omega_{wheel} * r \tag{2}$$

where V_{center} is the translational velocity of the center of the wheel, Ω_{wheel} is the angular velocity of the wheel, and r is the radius of the wheel. This sign convention is consistent for both wheels since positive rotation is defined as the direction that moves the Romi forward. Substituting Equation 2 into Equation 1b for both wheels, we find Equation 3

$$\Omega = \frac{r}{w}(\Omega_R - \Omega_L) \tag{3}$$

which finds the yaw rate of the Romi as a function of each wheel's rotational speed. It is important to note that the radii for both wheels are the same.

Similarly, the velocity of the center of the Romi, v, is found using the general plane motion equation with respect to the left wheel (Equation 4a)

$$v\,\hat{\imath} = V_L\,\hat{\imath} + \left(\Omega\,\hat{k}\right) \times \left(-\frac{\mathrm{w}}{2}\,\hat{\jmath}\right) \tag{4a}$$

$$v\,\hat{\imath} = \left(V_L + \frac{w}{2}\Omega\right)\,\hat{\imath} \tag{4b}$$

If we plug in Equation 3 into Equation 4b, we get Equation 5, which gives the velocity of the Romi as a function of each wheel's angular velocity

$$v = \frac{r}{2}(\Omega_R + \Omega_L). \tag{5}$$

Coordinate Systems

The Romi's local positioning can be expressed in global coordinates using ψ_R, which is the heading angle of the Romi from the global "x" axis. This is illustrated in Figure 3.

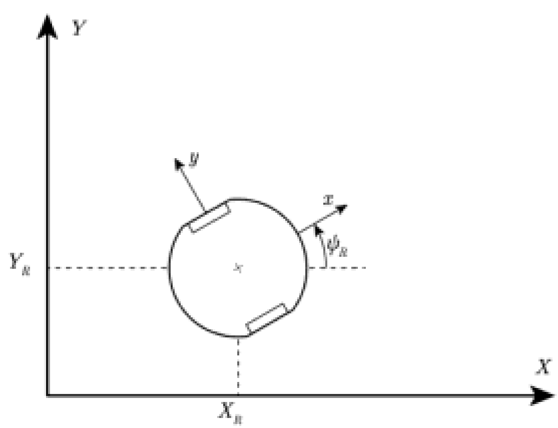

Figure 3. Romi's global and local coordinate systems.

From visual analysis of Figure 3, the local coordinates can be decomposed into their global coordinate components:

$$\begin{bmatrix} \hat{\imath} \\ \hat{\jmath} \end{bmatrix} = \begin{bmatrix} cos\psi & sin\psi \\ -sin\psi & cos\psi \end{bmatrix} \begin{bmatrix} \hat{I} \\ \hat{J} \end{bmatrix} \tag{6}$$

This means any vector given into the local coordinates can be transformed into the local coordinate system for the Romi.

Since the Romi is assumed only to drive forward or backward, the global velocities, $\dot{X}$ and $\dot{Y}$, can be described with Equation 7

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} vcos\psi \\ vsin\psi \end{bmatrix} \tag{7}$$

As a sanity check, the transformation from Equation 6 is applied to Equation 7

$$\begin{bmatrix} \hat{\imath} \\ \hat{\jmath} \end{bmatrix} = \begin{bmatrix} \cos\psi & sin\psi \\ -sin\psi & cos\psi \end{bmatrix} \begin{bmatrix} vcos\psi \\ vsin\psi \end{bmatrix}$$

$$\begin{bmatrix} \hat{\imath} \\ \hat{\jmath} \end{bmatrix} = \begin{bmatrix} vcos^2\psi + vsin^2\psi \\ -vsin\psi cos\psi + sin\psi cos\psi \end{bmatrix}$$

which is the same as

$$\begin{bmatrix} \hat{\imath} \\ \hat{\jmath} \end{bmatrix} = \begin{bmatrix} v \\ 0 \end{bmatrix}$$

This result is desired, since it is a restatement of our assumption that the Romi only travels along its heading axis.

The last important note is that since the global and local coordinates share the same "k" unit vector, the rate of change of the heading angle is equal to the yaw rate of the Romi

$$\dot{\psi} = \Omega \tag{8}$$

Motor Dynamics

The motor's dynamics are to be estimated using a first-order system, meaning they will be estimated using Equation 9

$$\tau_m \dot{\Omega}_m + \Omega_m = K_m V_m \tag{9}$$

where τ_m is the motor's time constant, Ω_m and $\dot{\Omega}_m$ are the motor's angular velocity and rate of change of angular velocity (respectively), K_m is the motor's steady state gain, and V_m is the voltage applied across the motor. It will prove useful in our state space model to reorder Equation 9 to isolate the rate of change of the motor's angular velocity (Equation 10)

$$\dot{\Omega}_m = \frac{1}{\tau_m}\left(K_m V_{,m} - \Omega_m\right) \tag{10}$$

State Space Model

To correctly encapsulate the system dynamics, six state variables were chosen: global position coordinates, X_R and Y_R, heading angle, ψ_R, arc length or "odometer" reading for the Romi, s, and angular velocity of the left and right wheel, Ω_L and Ω_R (respectively). The inputs to the system would be the voltage applied to the left and right motors, u_L and u_R (respectively). An output matrix was also constructed to be the global position coordinates, heading angle, arc length, Romi velocity, and yaw rate. The matrices in Equation 11 show the result of this state space representation.

$$\underline{x} = \begin{bmatrix} X_R \\ Y_R \\ \psi_R \\ s \\ \Omega_L \\ \Omega_R \end{bmatrix}, \qquad u = \begin{bmatrix} u_L \\ u_R \end{bmatrix}, \qquad \underline{y} = \begin{bmatrix} X_R \\ Y_R \\ \psi_R \\ s \\ v \\ \Omega \end{bmatrix} \tag{11}$$

To finish the state space representation, the time derivative for the state variables are required. The change of global position with respect to time have been derived in Equation 7, the change in heading angle with respect to time can be obtained from combining Equation 3 with Equation 8, the rate of change for the arc length is fundamentally the velocity of the Romi (which is Equation 5), and the rate of change of each angular velocity is found by applying Equation 10 to each motor. This results in both the state-space time derivative matrix (Equation 12) and output matrix (Equation 13).

$$\dot{\underline{x}} = \begin{bmatrix} \dot{X}_R \\ \dot{Y}_R \\ \dot{\psi}_R \\ \dot{s} \\ \dot{\Omega}_L \\ \dot{\Omega}_R \end{bmatrix} = \begin{bmatrix} \frac{r}{2} cos\psi_R(\Omega_R + \Omega_L) \\ \frac{r}{2} sin\psi_R(\Omega_R + \Omega_L) \\ \frac{r}{w}(\Omega_R - \Omega_L) \\ \frac{r}{2}(\Omega_R + \Omega_L) \\ \frac{1}{\tau_L}(K_L u_L - \Omega_L) \\ \frac{1}{\tau_R}(K_R u_R - \Omega_R) \end{bmatrix} \tag{12}$$

$$y = \begin{bmatrix} X_R \\ Y_R \\ \psi_R \\ s \\ \frac{r}{2}(\Omega_R + \Omega_L) \\ \frac{r}{w}(\Omega_R - \Omega_L) \end{bmatrix} \tag{13}$$

Discussion Questions

1. Explore the validity of the no-slip assumption and any other assumptions required for the kinematic model in favor of a full dynamic model. Consider things like how the nonlinear aspect of the motors may or may not easily integrate with the model, how the mass and friction properties of the system may influence the model, and any other effects that may not match the idealized nature of the kinematic model. Is the kinematic model good enough for our purposes?
   a. There are multiple assumptions required for the current kinematic model. First, the no-slip condition is required to relate the speed of Romi's side with that side's respective wheel speed. It is predicted that slipping is not unlikely given the combination of high-speed regimes and unpredictable surfaces over which the Romi rolls. Additionally, the motor is assumed to be first order, but it might exhibit non-linear behavior. This would be difficult to model internally within the Romi since obtaining information for the model would require gathering information on the masses, friction properties, and the effect of the path on loading. It is hypothesized that this complex modeling would be too difficult to integrate. For the most part, the current kinematic model is good enough to estimate

the state of the Romi for the short distances it is intended to be used, despite the fact that a fuller dynamic model would yield higher accuracy.

2. In Lab you will run (or have already ran) step response tests to experimentally determine the parameters associated with Romi's motors. Is it valid to assume that the parameter values while driving forward are the same as while turning? Compare pivoting in place and driving in a straight line, will both of these cases have the same effective inertia loading the motors?
    a. The loading on the motors will be dependent on the movement being executed. Both wheels rotating such that Romi drives forward, would have external loading from the translational inertia of Romi, while them rotating in the opposite direction to induce spinning would experience an external load from the Romi's rotational inertia; two quantities that are different. This means the type of loading on each motor is dependent on the type of motion occurring. This change in mechanical loading would result in different time constants for each motion.

3. One of the problems with models like this is that significant deviation or drift will occur between the simulation and the behavior of the real vehicle, especially when accumulated over a long simulation. The concept of dead reckoning suggests that we can predict the position of a system simply by integrating its velocity or acceleration over time. Naturally, small deviations, like those caused by minor wheel slip, add up over time. A small error in velocity, once integrated over time to produce position, can lead to wildly inaccurate estimations of position. How can we correct for this drift error? What other information would help us keep our simulated model tracking with reality? *Hint:* consider what kind of sensors may be added to the robot to measure the value of your selected state variables.
    a) The inertia measurement unit (IMU) can be used to sense the physical translation and rotation experienced by the Romi and then utilized as a comparison to see if the model is drifting too far from that sensor's output. The appropriate correction algorithm would have to be created, however.

4. While this is not a controls course, you will need to implement some kind of controller on your robot to complete the term project. For this question, investigate ways in which you may incorporate feedback into your system to make the robot move in ways that will be advantageous for the term project. Address nuances such as: 1) What sensor inputs will be useful for feedback. 2) How you might split up a complicated path into segments such as straight lines, circular arcs, in-place turns, etc.
    a) In order to execute a closed-loop control of the motor, there needs to be a measurement that can be compared to the input. This means, at a minimum, the encoder needs to provide the measured velocity of the wheels. But if a cascaded control system is desired, monitoring the position using the IMU could be used as well.
    b) Since there are different ideal outputs for each type of movement (straight vs. circular vs. arcs), it is theorized that there should be some internal structuring where the closed-loop controller takes in specific measurements depending on what the desired output is. Since rotating and translation (for example) would require different functions to determine how far the actual value has drifted from the desired value, the motor controller will have to know which type of movement is being done.

5. Is it possible to determine the orientation and location of the robot after it has followed a specific path simply by knowing the angle swept out by each wheel? If not, would it be possible to know

the orientation and location with a time history of the angle swept out by each wheel? If you know the robot's orientation and location, can you figure out the angle swept out by each wheel? Investigate what information is required to track the robot's absolute position in space, or its position relative to a particular maneuver, like a circular arc. What kind of data will you need to track in your firmware to make use of this information?

1. The distance swept out by each wheel should be obtainable from the yaw rate and forward velocity of the Romi. This means that knowing how much each wheel has rotated obtains this information. In a scenario where slipping is present, the IMU should correct the "misreading" caused by the encoders (as discussed in the previous question). Knowing this information, there are two options: track the orientation with a time-history or just track the relative change between the start and end of maneuvers. I believe both to be viable.

References:

- Refvem, C. (2025). *Homework 0x03 - System Modeling* [Unpublished lab instructions]. Department of Mechanical Engineering, Cal Poly. Retrieved from https://canvas.calpoly.edu/courses/161863/assignments/1368599?module_item_id=4790183
- Refvem, C. (2025). *Lecture 10 – Controls* [Unpublished lecture notes]. Department of Mechanical Engineering, Cal Poly.
- Refvem, C. (2025). *Lecture 11 – Romi Dynamics* [Unpublished lecture notes]. Department of Mechanical Engineering, Cal Poly.