EXPERIMENT 2: LOCOMOTION AND KINEMATICS

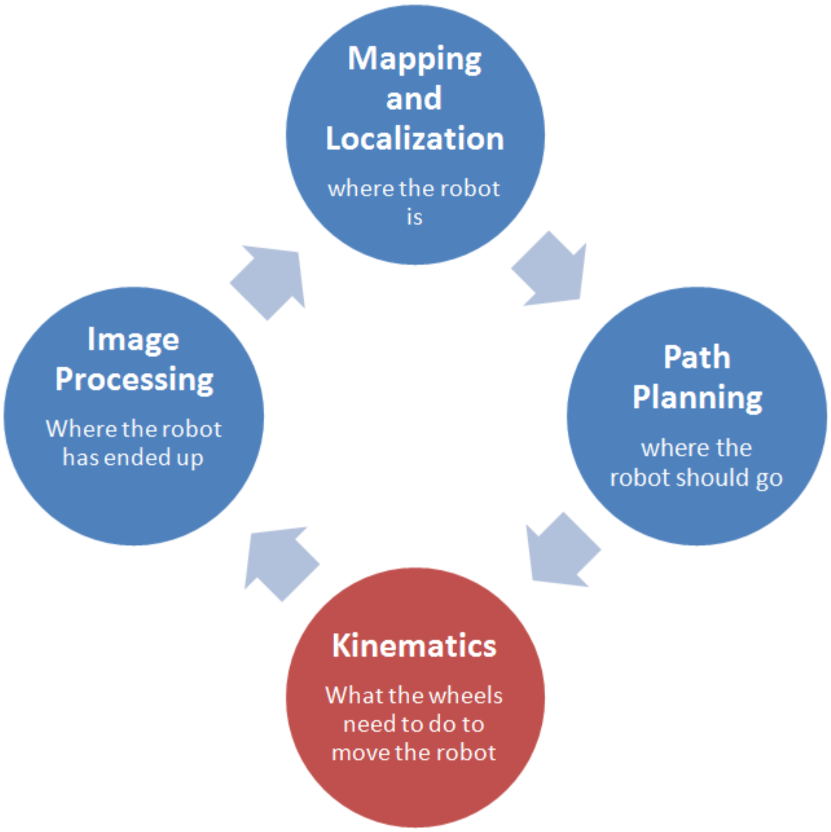
The purpose of this experiment is to study the basic motion behaviour of the Quanser QBot 2e Mobile Platform. The following topics will be studied in this experiment.

Topics Covered

- Differential Drive Kinematics
- Forward and Inverse Kinematics
- Odometric Localization and Dead Reckoning

Prerequisites

- The QBot 2e has been setup and tested. See the QBot 2e Quick Start Guide for details.
- You have access to the QBot 2e User Manual.
- You are familiar with the basics of MATLAB® and SIMULINK®.


Mapping and Localization
where the robot is
Path Planning
where the robot should go
Kinematics
What the wheels need to do to move the robot
Image Processing
Where the robot has ended up

Kinematics is used to determine the appropriate actuator commands for the robot

DIFFERENTIAL DRIVE KINEMATICS

The Quanser QBot 2e Mobile Platform uses a drive mechanism known as differential drive. It consists of two central drive wheels mounted on a common axis that bisects the robot. Castors at the front and back of the robot stabilize the platform without compromising movement. Each drive wheel can be independently driven forward and backward, to actuate different motion from the robotic base. This approach to mobile robot wheel geometry is very common due to it's simplicity and maneuverability.

Differential drive kinematics is the mathematical relationship that maps the independent motion of the wheels to the overall movement of the robot chassis. This fundamental topic is the foundation of all mobile robot control, in that it is chiefly responsible for the predictable mobility of the robot. In this laboratory you will investigate the differential drive kinematics of the Quanser QBot 2e Mobile Platform.

Topics Covered

- Differential drive mechanism of the QBot 2e
- Derive the kinematics model of the QBot 2e differential drive system

1 Background

The QBot 2e is driven by a set of two coaxial wheels. These wheels are actuated using high-performance DC motors with encoders and drop sensors. To determine the relationship between the independent motion of the two wheels and the motion of the overall robot, we begin by modeling the motion of the robot about a common point.

Let the radius of the wheels be denoted by r, and the wheel rotational speed be denoted by ω_L and ω_R for the left and right wheel respectively. The linear speed of the two wheels along the ground is then given by the following equations:

$$v_L = \omega_L r \tag{1.1}$$

$$v_R = \omega_R r \tag{1.2}$$

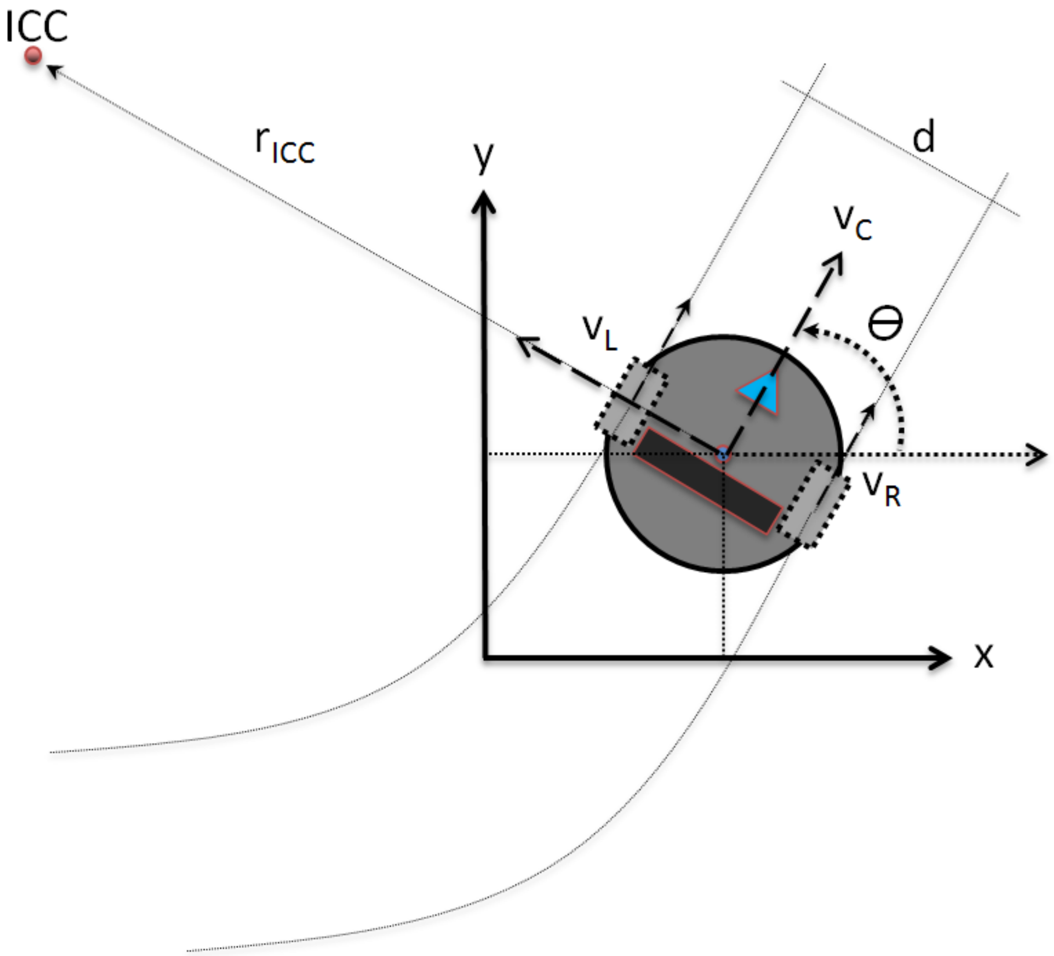


Figure 1.1: Quanser QBot 2e Mobile Platform Reference Frame Definitions

Assuming there is no wheel slippage, the QBot 2e can move along the horizontal plane in straight or curved trajectory, as well as spin on a spot, by varying the relative speed between the left and right wheels.

Since we are assuming that the wheels are not subject to slip, the motion of the wheels are constrained to move along their forward and backward directions. This, together with the inherent constraint that is imposed by the robot chassis coupling the two wheels together, means that all robot chassis rotations must be about a point that lies along the common wheel axis. For example, if only one of the two wheels rotates, the robot would rotate (pivot) about the non-moving wheel. On the other hand, if both wheels rotate at the same speed, the robot rotates about a point infinitely far from the robot. This center of rotation is known as the *Instantaneous Center of Curvature* (ICC).

1.1 Kinematic Model

Let r_{ICC} be the distance measured from the center of the robot chassis, which is halfway between the left and right wheels, to the ICC. If d is the distance between the left and right wheels, θ is the heading angle of the robot, and v_C is the (forward/backward) speed of the robot chassis center, the motion of the QBot 2e chassis can be summarized in the following equations:

$$v_C = \dot{\theta}\, r_{ICC} \tag{1.3}$$

$$v_L = \dot{\theta}\left(r_{ICC} - \frac{d}{2}\right) \tag{1.4}$$

$$v_R = \dot{\theta}\left(r_{ICC} + \frac{d}{2}\right) \tag{1.5}$$

Notice that v_C, v_L and v_R are all defined along the same axis, which lies in the forward/backward direction of the chassis. Given the wheel speed, v_L and v_R, the robot speed, v_C, the angular rate, $\omega_C = \dot{\theta}$, and the distance from ICC, r_{ICC}, we can relate the motion of the wheels to the motion of the robot using the following kinematic model for the differential drive system:

$$v_C = \frac{v_R + v_L}{2} \tag{1.6}$$

$$\omega_C = \dot{\theta} = \frac{v_R - v_L}{d} \tag{1.7}$$

$$r_{ICC} = \frac{d\,(v_R + v_L)}{2\,(v_R - v_L)} \tag{1.8}$$

2 In-Lab Exercise

2.1 Wheel Command Scenarios

The Simulink model for this exercise is `QBot2e_Diff_Drive_Kinematics.mdl` the snapshot of which shown in Figure 2.1.

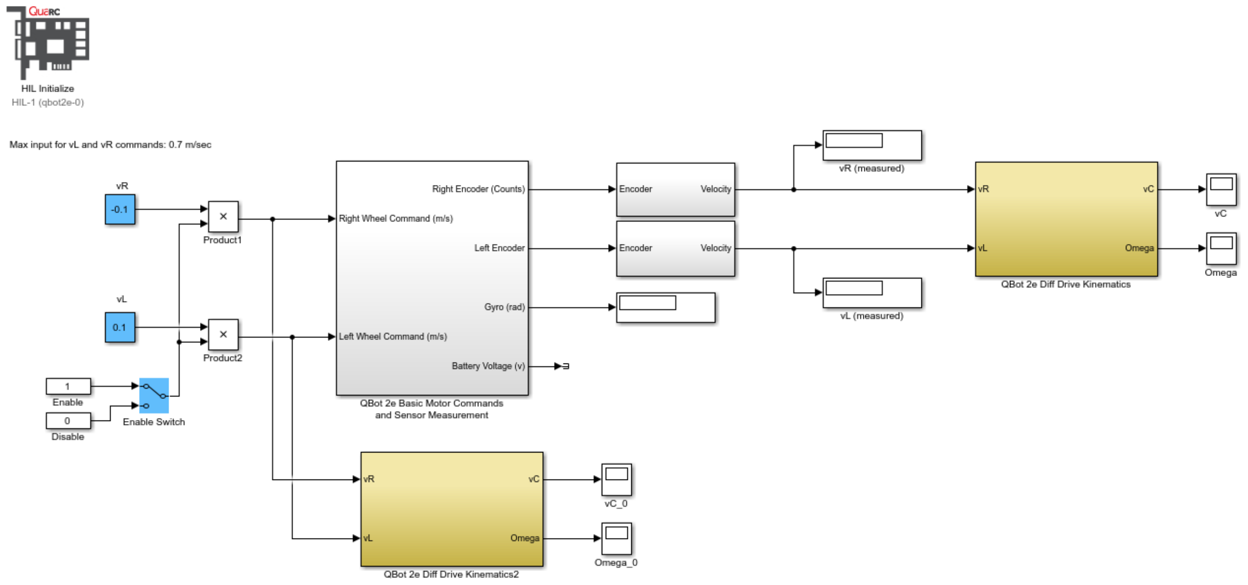


Figure 2.1: Snapshot of the controller model `QBot2e_Diff_Drive_Kinematics.mdl`

The *QBot 2e Diff Drive Kinematics* block, shown in yellow, receives the left and right wheel velocities as input and computes the forward and angular velocities. Compile and run the model, then follow the procedure outlined below. Make observations on the motion of the robot, and answer the associated questions.

1. Wait 5 seconds until the QBot 2e has fully initialized, and then enable the movement of the robot using the manual switch shown in Figure 2.1.
2. Set the left and right wheel velocity set points, highlighted with blue, to 0.1 m/s. Run the model and observe the linear and angular velocities. What are the values of r_{ICC} and ω_C when the left and right wheels are moving at the same speed (i.e. $v_L = v_R$)? What do your results indicate about the relationship between r_{ICC} and ω_C?
3. Change the right wheel velocity to -0.1 m/s and keep the left wheel velocity set point at 0.1 m/s. What is the value of r_{ICC} when the left and right wheels are moving at the same speed but in the opposite direction (i.e. $v_L = -v_R$)? What do these results indicate? Does the relationship identified earlier hold?
4. What does it mean when r_{ICC} is negative?
5. What does it mean when ω_C is negative?

FORWARD AND INVERSE KINEMATICS

The objective of this exercise is to investigate the forward and inverse kinematics of the Quanser QBot 2e Mobile Platform. Forward kinematics is used to determine the linear and angular velocity of the robot in the world coordinate frame given robot's wheel speeds. Inverse kinematics on the other hand, is used to determine the wheel commands needed for the robot to follow a specific path at a specific speed. Inverse kinematics is an essential tool for mobile robotics as it bridges the gap between a navigation and path planning module, and actual robot locomotion.

Topics Covered

- Forward kinematics model of the QBot 2e
- Inverse kinematics model of the QBot 2e

1 Background

A typical kinematics model that computes the robot chassis speed, v_C, and turning rate, ω_C, from the wheel speed, v_R and v_L, with wheel separation distance, d, for a robot with differential drive system like the QBot 2e is given by:

$$v_C = \frac{1}{2}(v_R + v_L) \tag{1.1}$$

$$\omega_C = \dot{\theta} = \frac{1}{d}(v_R - v_L) \tag{1.2}$$

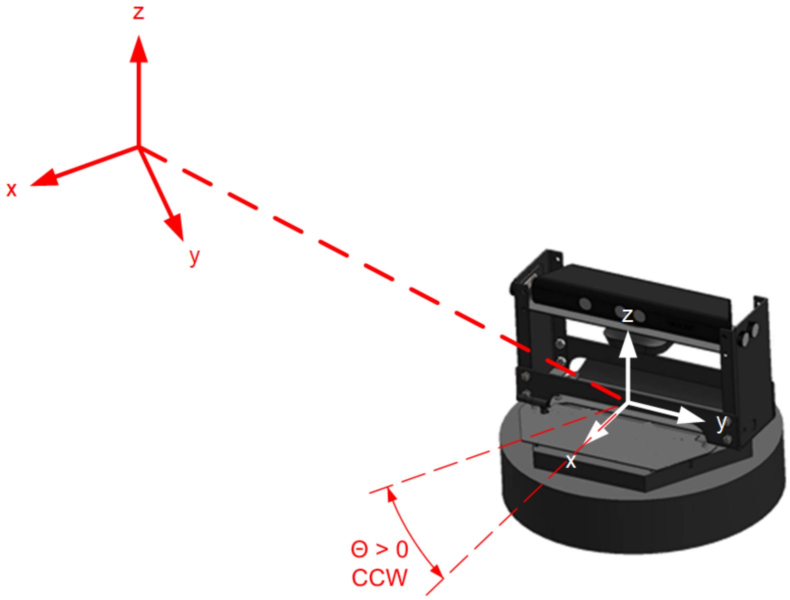


Figure 1.1: Kinematics is used to determine the appropriate actuator commands for the robot

Implicit in the derivation of the above kinematics model is the use of a local frame of reference. In other words, the chassis speed, v_C, is expressed in the forward/backward (heading) direction of the robot chassis and not the global frame that would be used in a map of the environment. Since the robot chassis heading changes when the angular rate is non-zero, $\omega_C = \dot{\theta}$, we need to apply a transformation to the differential drive kinematics model in order to compute the robot chassis motion with respect to the global reference frame. For a robot with a heading, θ, the transformation required is the following rotation matrix:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.3}$$

This transformation maps motion expressed with respect to the robot chassis local frame to the corresponding motion in the global frame.

The corresponding inverse mapping is given as follows:

$$R^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.4}$$

1.1 Forward Kinematic Model

We define a state vector, S, as the position, x and y, and the heading, θ, of the robot chassis. Its definition and rate of change are given as follows:

$$S = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \qquad \dot{S} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

The x and y axes lie in the “ground” plane that the robot primarily travels in. The heading, θ, is measured about the vertical z axis, which is defined as positive pointing upwards. The heading is zero, ($\theta = 0$), when the robot chassis' forward direction aligns with the global x axis. The rate of change of the states can be expressed in terms of the robot chassis speed, v_C, and angular rate, ω_C, as follows:

$$\dot{S} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = R \begin{bmatrix} v_C \\ 0 \\ \omega_C \end{bmatrix} = \begin{bmatrix} v_C \cos\theta \\ v_C \sin\theta \\ \omega_C \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(v_R + v_L)\cos\theta \\ \frac{1}{2}(v_R + v_L)\sin\theta \\ \frac{1}{d}(v_R - v_L) \end{bmatrix} \tag{1.5}$$

Equation 1.5 represents the forward kinematics model for the QBot 2e that computes the linear speed, ($\dot{x}$ and $\dot{y}$), and turning rate, (ω_C), of the robot chassis given its heading, (θ), and wheel speed, (v_R and v_L).

Similarly, the position of the Instantaneous Center of Curvature (ICC) in space, (x_{ICC} and y_{ICC}), expressed with respect to the global reference frame can be obtained as follows:

$$\begin{bmatrix} x_{ICC} \\ y_{ICC} \\ 0 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} + R \begin{bmatrix} 0 \\ r_{ICC} \\ 0 \end{bmatrix} = \begin{bmatrix} x - r_{ICC}\sin\theta \\ y + r_{ICC}\cos\theta \\ 0 \end{bmatrix} = \begin{bmatrix} x - \frac{d}{2}\frac{(v_R+v_L)}{(v_R-v_L)}\sin\theta \\ y + \frac{d}{2}\frac{(v_R+v_L)}{(v_R-v_L)}\cos\theta \\ 0 \end{bmatrix} \tag{1.6}$$

This can be useful for path planning or obstacle avoidance algorithms.

1.2 Inverse Kinematic Model

As mentioned in the Background section, if you want the robot to follow a certain path or speed, you need to send appropriate wheel commands to the robot. The inverse kinematics model computes the required wheel speed to obtain a desired robot chassis speed v_C, and angular rate ω_C. It is obtained by solving Equation 1.1 and Equation 1.2 together for the wheel speed v_R and v_L and is given as follows:

$$\begin{bmatrix} v_R \\ v_L \end{bmatrix} = \begin{bmatrix} v_C + \frac{1}{2}d\,\omega_C \\ v_C - \frac{1}{2}d\,\omega_C \end{bmatrix} \tag{1.7}$$

2 In-Lab Exercise

2.1 Forward Kinematics

The controller model for this exercise, shown in Figure 2.1, is called `QBot2e_Forward_Kinematics.mdl`.

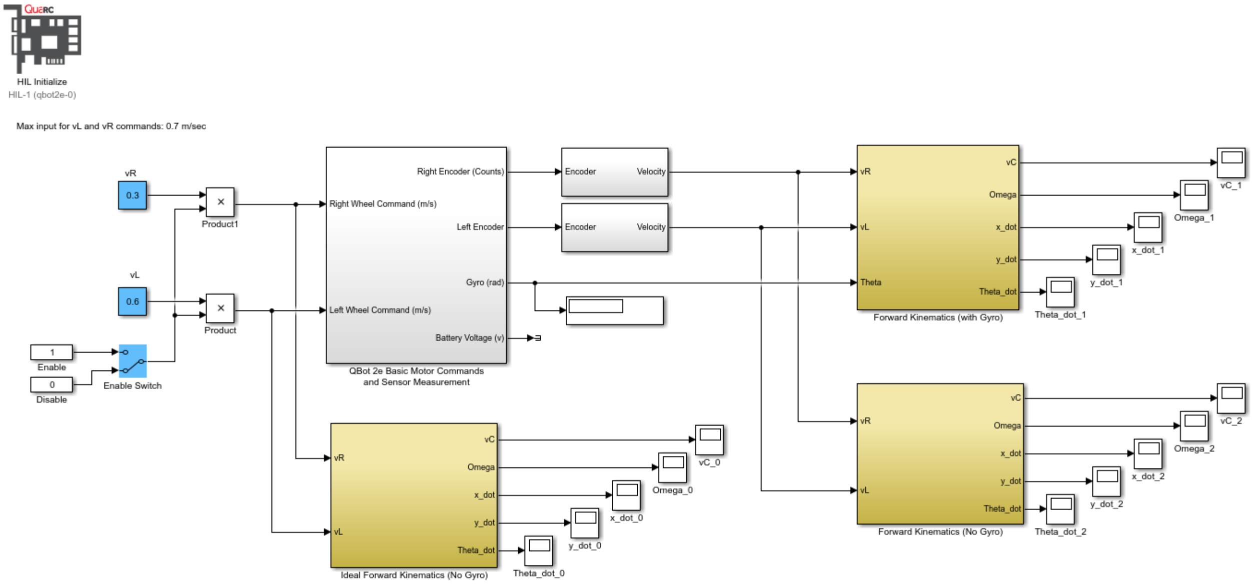


Figure 2.1: Snapshot of the controller model `QBot2e_Forward_Kinematics.mdl`

The forward kinematic blocks, shown in yellow, receive the left and right wheel velocities as inputs and compute the linear velocities in the world coordinate frame. These blocks are used on both commanded wheel velocities giving us the *ideal* robot speed, as well as on the measured wheel velocities, resulting in the actual measured robot velocity.

Compile and run the model, then follow the procedure outlined below. Make observations on the motion of the robot, and answer the associated questions.

1. Wait until the QBot 2e has fully initialized (you should hear the QBot 2e start-up chime), then enable the movement of the robot using the manual switch shown in Figure 2.1.
2. Set the left and right wheel velocity set points, highlighted with blue, to 0.6 m/s and 0.3 m/s accordingly. Run the model and observe the ideal and measured linear and angular velocities in the world coordinate frame ($\dot{x}$, $\dot{y}$ and $\dot{\theta}$).
3. What is the shape of the robot trajectory when the right wheel is commanded to travel at twice the speed of the left wheel (i.e. $v_R = 2v_L$)? Comment on the effect of changing the value of v_L on the robot chassis trajectory.
4. Compute the required constant wheel speeds v_R and v_L to generate a trajectory with a constant turning rate of $\omega_C = 0.1\, rad/s$ and a constant turning radius of $r_{ICC} = 1\, m$. Implement the wheel speed command on the robot chassis, observe and explain the resulting chassis trajectory. Compare the desired turning rate and radius to the measured turning rate and radius.

2.2 Inverse Kinematics

The QUARC model for this exercise is called `QBot2e_Inverse_Kinematics.mdl` and is shown in Figure 2.2. In this model, the Inverse Kinematics block for the QBot 2e is shown in yellow and the input commands for v_C and ω_C are highlighted in blue.

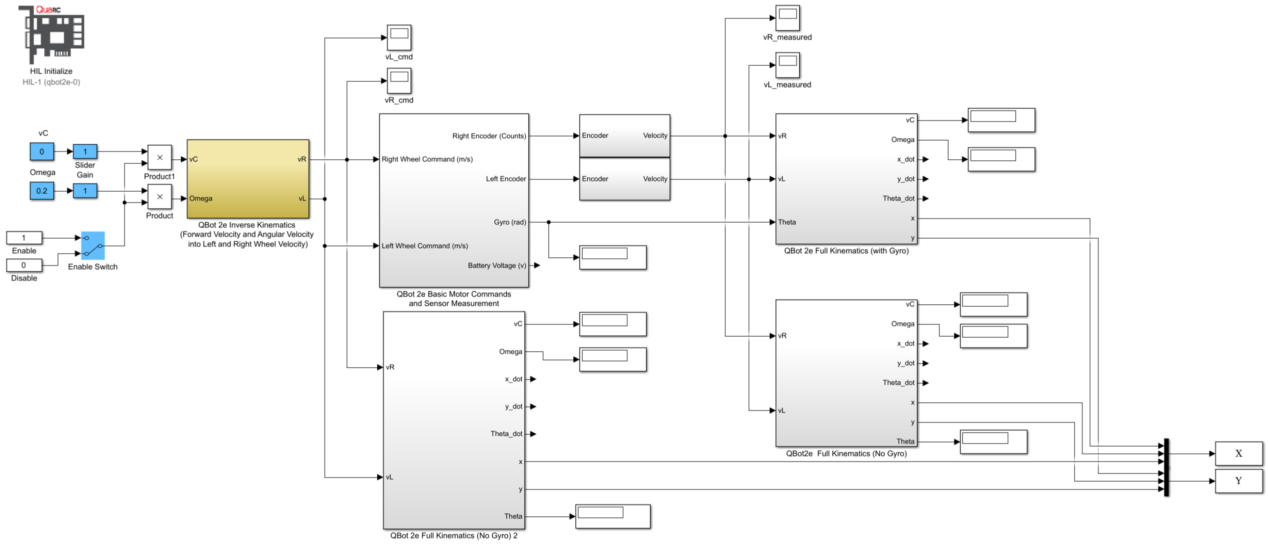


Figure 2.2: Snapshot of the controller model `QBot2e_Inverse_Kinematics.mdl`

Open the supplied Inverse Kinematics controller model, compile it and go through the following steps. For each step observe the motion of the robot chassis and answer the associated questions.

1. Wait until the QBot 2e has fully initialized (you should hear the QBot 2e start-up chime), then enable the movement of the robot using the manual switch shown in Figure 2.2.
2. Set the desired forward speed $v_C = 0.1$ m/s and the desired turning rate ω_C = 0.1 rad/s.
3. Run the model.
4. Record and observe the corresponding wheel speed commands v_R and v_L and the measured values. Compare them with the wheel speeds computed in the Forward Kinematics exercise.
5. Set the desired forward speed $v_C = 0$ m/s and the desired turning rate $\omega_C = 0.2$ rad/s and run the model again. Observe the behaviour of the robot as well as the measured v_L and v_R signals.

ODOMETRIC LOCALIZATION AND DEAD RECKONING

The objective of this exercise is to explore the concept of Odometric Localization as applied to the Quanser QBot 2e Mobile Platform.

Topics Covered

- Equations of motion for odometry
- Accumulated errors

1 Background

Odometric Localization, also known as Dead Reckoning, is the estimation of a robot's position and orientation (pose) based on the measured or estimated motion of the robot. In the case of the Quanser QBot 2e Mobile Platform, the procedure for odometric localization involves estimating the wheel speeds, ($v_R(t)$ and $v_L(t)$), based on encoder data or the integrated gyro. The forward kinematics model is then applied to estimate the robot chassis' linear speed, $v_C(t)$, and angular rate, $\omega_C(t)$. The data is then integrated over time starting from a known initial location, ($x(0)$ and $y(0)$), and heading, $\theta(0)$, to obtain an estimate of the robot chassis' pose.

This approach to localization is the most basic methodology used in mobile robotics, but is still routinely applied in industrial robotics applications that do not require high-fidelity location estimation, or as a redundant backup system for validation and error detection.

1.1 Equations of Motion

Given the robot chassis state vector, $S(t)$, and its rate of change, $\dot{S}(t)$, expressed in the global inertial frame, the robot pose at time, t, can be computed as follows:

$$S(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \int_0^t \dot{S}(t)\, dt \tag{1.1}$$

For the QBot 2e, given the wheel separation, d, the heading, θ, and the wheel speed, v_R and v_L, the equations of motion for odometric localization are given by:

$$S(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \int_0^t \begin{bmatrix} \frac{1}{2}(v_R(t) + v_L(t)) \cos\theta(t) \\ \frac{1}{2}(v_R(t) + v_L(t)) \sin\theta(t) \\ \frac{1}{d}(v_R(t) - v_L(t)) \end{bmatrix} dt \tag{1.2}$$

In MATLAB/Simulink, it is easy to use the built-in integration blocks to solve the odometric calculations. However, for low-level languages, we need to employ other integration methods. For example, for a small time step, δt, the above QBot 2e equations of motion can be approximated as a first order Taylor series expansion:

$$S(t+\delta t) = S(t) + \dot{S}(t)\delta t = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{2}(v_R(t) + v_L(t)) \cos\theta(t) \\ \frac{1}{2}(v_R(t) + v_L(t)) \sin\theta(t) \\ \frac{1}{d}(v_R(t) - v_L(t)) \end{bmatrix} \delta t \tag{1.3}$$

2 In-Lab Exercise

2.1 Trajectory Errors

The controller model for this exercise, shown in Figure Figure 2.1, is called `QBot2e_Odometric_Localization.mdl`.

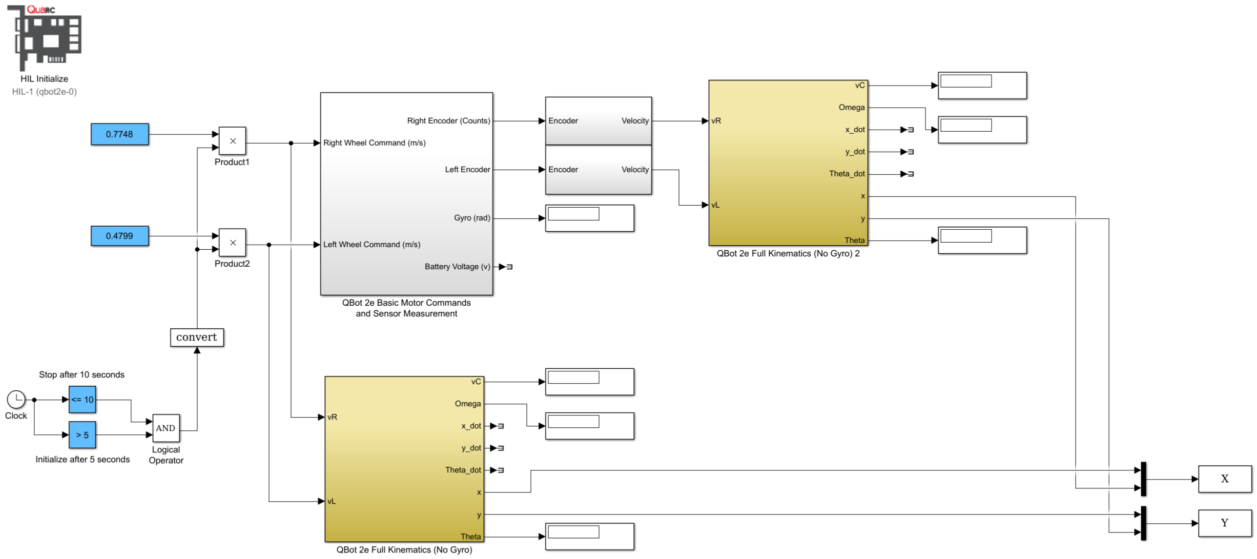


Figure 2.1: Snapshot of the controller model `QBot2e_Odometric_Localization.mdl`

The supplied model includes an open-loop trajectory controller for the QBot 2e. The specified path consists of the following segments:

- Rotate in a large circle with a radius of 0.5 m
- Stop when the robot has returned to the initial position

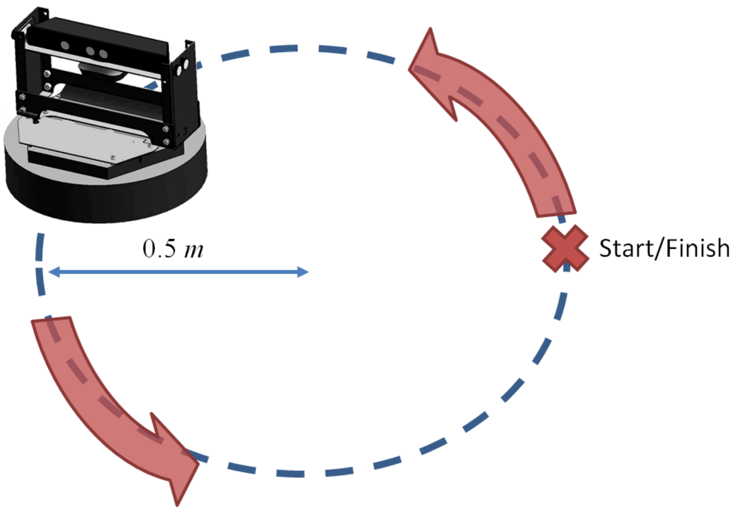


Figure 2.2: Path following controller desired path

At the end of the specified path, the QBot 2e is expected to have returned to approximately the starting point. Compile and run the model, then follow the procedure outlined below. Make observations on the motion of the robot, and answer the associated questions.

Note: It is helpful to mark the starting location and heading prior to running the supplied controller model.

1. Once the QBot 2e has fully initialized (upon hearing the QBot 2e start-up chime), it will begin to move.
2. Measure and record the final position and heading of the robot chassis with respect to the starting position and heading.

   Note: A MATLAB function called `plotXY.m` has been provided to generate an appropriate plot for the path analysis.

   - What is the error between the desired end-point of the robot, and the actual final x-y position? Recall, the x axis is in the forward/backward direction of the robot and the y axis is in the left/right direction.
   - What is the heading error?
3. Determine the theoretical right wheel and left wheel commands, and time required, for the QBot to travel in a straight line 5 meters in length. Enter your values into the appropriate fields in the controller model, indicated in blue and shown in Figure 2.2.
4. Repeat steps 1 and 2, and record your new results. Modify the wheel commands and time values until the robot is able to reach the correct position.
5. Identify and discuss the different factors that contribute to the path/trajectory tracking error. Specifically, note the error reported by the controller model and the error based on direct physical measurements.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com