Week 1

Monday, March 15, 2021 10:23 AM

RoboWeek 1

Chapter 13

Robo Week 1a: Intro to Mobile Robotics

Schedule

13.1 Technology Setup

We will be working with the NEATO BotVac as a way to physically validate concepts and calculations during this module. Programming and running experiments with the NEATO robots can be done with both physical robots and using the robot simulator. For the Spring 2021 version of QEA2, we will be using the robot simulator throughout the module. The setup steps on the Meeto Your Neato page must be completed before using the simulator. **NOTE:** The simulator is regularly updated and the workflow refined, so all students should complete these steps to ensure they have the most recent version.

13.2 Robo Ninja Warrior

Welcome to Module 3 entitled "Robo Ninja Warrior." In this module you'll be learning some of the fundamental ideas, concepts, and algorithms that lie at the heart of robotics. Along the way we'll be revisiting some mathematical and analytical concepts we touched upon earlier in the semester. Not only will we be applying these concepts in new contexts and to new purposes, but also extending them in important ways. The module is structured around a series of challenges in which you will be programming your robot to perform various tasks autonomously. As you and your robot face tougher and tougher challenges, you will need to carefully integrate a wider range of techniques in order to successfully complete the task at hand.

When we typically do this module, we use the Neato BotVac as our robot platform (you'll learn more about the platform soon, but briefly it is a really nice robot platform that has some very advanced sensors). Of course we are now separated from our beloved Neatos, but never fear! The QEA team has got you covered. In this module we'll be using a Neato simulator, and we are excited about some of the cool things we'll be able to do when working in simulation. You'll learn more about the Neato on the Meeto Your Neato page.

13.3 Demonstration of Robot Simulator

The robot simulator developed for QEA allows a user to command the wheel velocities of their virtual robot, interact with outputs from the bump and LIDAR sensors, and measure motion using the wheel encoders. The videos below demonstrate the virtual robot tackling two of the challenges you will encounter in this module.

1. **The Bridge of Doom:** In the bridge of doom challenge, you will need to program your robot to carefully traverse a bridge defined by a parametric curve. Program your robot incorrectly and it will plunge into a (virtual) lava pit! A great example of this challenge is here.
2. **The Gauntlet**: In the Gauntlet challenge, you will navigate a cluttered environment using the data form your robot's LIDAR sensor. A succesful navigation of the Gauntlet can be viewed here.

13.4 What is a Robot?

Before diving into the challenges, let's take a step back and look at some definitions of the word "robot". Merriam-Webster provides three definitions of the word.

- a machine that looks like a human being and performs various complex acts (such as walking or talking) of a human being
- a device that automatically performs complicated often repetitive tasks
- a mechanism guided by automatic controls

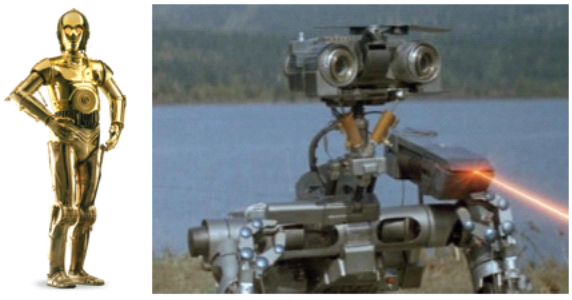


Figure 13.1: **Left:** C-3PO from the Star Wars franchise. **Right:** Johnny Five from the Short Circuit movies.

Exercise 13.1

Jot down a list of devices. For each device, determine which, if any, of the three definitions of the word "robot" apply.

These disparate definitions highlight the fact that depending on who you ask, the answer to the question "what is a robot?" will likely be very different. In a sense these three definitions proceed along a continuum of more restrictive to looser definitions, with the definition "a mechanism guided by automatic controls" being the loosest. Under this definition many things that you probably wouldn't intuitively call "robots" are just that. Take for example a thermostat. A thermostat is a mechanism that automatically regulates the heat in a building by comparing the measured temperature with a "desired" temperature. By the third definition, a thermostat is certainly a robot. At this point you may be thinking that if something as simple

Robot: Roomba, Kiva robot, slow dose
Not: laptop, toaster, car

as a thermostat is a robot, then definition 3 must be completely bogus. After all, robots are supposed to be complicated and hard! However, today we will see that robots can in fact be quite simple. Further, simple robots can do some pretty complicated things.

13.5 Sensory-Motor Loops

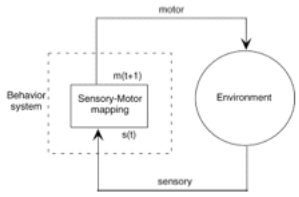


Figure 13.2: A schematic of a robot controlled by a sensory-motor loop.

We ended our last section on a somewhat cryptic note. If we seek to design simple robots, how on earth can they do complex things? The answer to this question lies in the fact that robots are not isolated machines, but instead interact with a complex and ever-changing world. A simple model that captures this idea is the sensory-motor loop (see Figure 13.2). The model situates the robot within an environment that it interacts with through two pathways.

The first is a **motor pathway** by which the robot executes actions which affect its environment. In our thermostat example these actions would be turning the heating system on or off. In a more conventional example of a robotic arm in a manufacturing plant, this could be operating the motors that control the joints of the arm.

The second is a **sensory pathway** by which the robot perceives its environment. In our thermostat example this could be a temperature sensor such as a thermocouple. In the case of a robotic arm in a manufacturing plant, this could be a potentiometer that measures the angle of each of the arm's joints or pressure sensors that measure contacts between the robot arm and other objects.

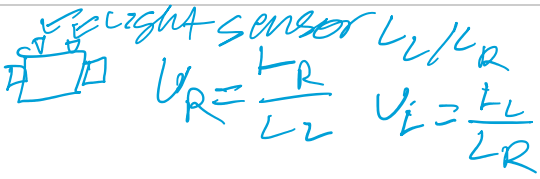
The "brain" of the robot, if you will, is defined by the box labeled "behavior system". In the general case you could imagine that the robot's brain might integrate multiple pieces of sensory information over time to form representations of the world around it. Take for example a robot mapping a building. The robot could build a progressively more detailed map by moving around in the building and collecting sonar readings (which provide an estimate of distance to objects in the world) over time. Putting aside this more complex form, let's restrict ourselves to robots with fairly simple behavior systems. **What about a robot that has no memory at all?** Such a robot would have to make all of its decisions based on its current sensory information.

Exercise 13.2

Design robots using the model in Figure 13.2. Restrict yourselves to robots that have no memory (i.e. ones that act at any moment in time directly based on their sensory input). To help get your creative juices flowing, it may help to make lists of sensors and actuators. You can then create interesting ideas by seeing what would happen if you paired a particular sensor with a particular actuator in a particular context. Don't worry too much about trying to design useful robots (whimsical is good), the goal here is to be creative and to think through the mental simulations necessary to understand how your robot would behave. Here are some suggestions for sensors and actuators.

Sensors: vibration sensor, microphone, camera, thermal camera, wheel rotation sensor, pressure sensor, light intensity sensor, laser range sensor, bump detector, temperature sensor, breathalyzer, etc. (Wikipedia has a good list).

Actuators (which is just a more general term for something that causes an action, e.g., a motor): DC motors, combustion engines, stepper motors, solenoids, speakers, lasers, LEDs, etc.



13.6 Grey Walter's Tortoises

Two very early examples of electric robots that worked using the principle of sensory-motor mappings were Grey Walter's robotic "Tortoises" Elmer and Elsie (see left panel of Figure 13.3). This YouTube video probably tells the story better than we possibly could.

As Grey Walter said himself, the robots behave as if they had a very simple two-cell nervous system that specifies the sensory-motor mapping (or behavior system). Despite this striking simplicity, the robots are capable of complex behavior such as obstacle avoidance and phototaxis (navigating towards the light that marks the charging kennel). This is an example of what we've been alluding to several times in this document: simple sensory-motor mappings can lead to complex behavior when put into a complex environment.

13.7 Braitenberg Vehicles

The pioneering work of Grey Walter was extended by a number of others. One particularly interesting line of research was conducted by Valentino Braitenberg. Valentino Braitenberg was interested in how vehicles controlled by very simple sensory-motor loops could execute behaviors that when viewed by humans would cause them to ascribe emotion and feelings of intelligence and intentionality to these vehicles. The name typically used to refer to these hypothetical robots is "Braitenberg Vehicles". While Braitenberg never actually built these vehicles (he was more interested in how these simple vehicles might inform various philosophical issues, particularly in the area of philosophy of mind), others have followed up and actually built these vehicles. Here is a video from a group at MIT that built several of Braitenberg's vehicles.

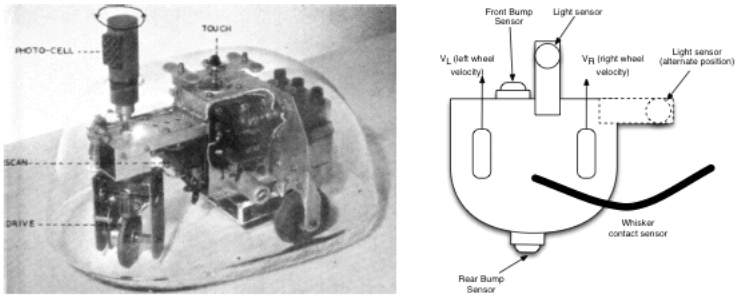


Figure 13.3: **Left:** Gray Walter's Tortoise Elsie. **Right:** A schematic of the vehicle from the real-life Braitenberg vehicles video. The robot is a differential drive vehicle with two wheels. We use V_L and V_R to refer to the velocities of the left and right wheels (positive is forward by convention). Also labeled are the other sensors on the robot, including the light sensor that reads higher values when exposed to more light, a whisker sensor that reads 0 when it is not in contact with something and 1 if it is, and two bump sensors that read 1 when they hit something and 0 otherwise.

A schematic of the vehicle (or robot) in the video is shown in the right panel of Figure 13.3. The robot has two motors that each control one of the robot's wheels. We can use the symbols V_L and V_R to refer to the velocities of each of the wheels (positive indicating a forward velocity). Using the framework from the previous section, these are our actuators. The robot also has a number of sensors. A light sensor outputs a continuous value which reads out larger values when in the presence of bright light and smaller values in the presence of low light (this is basically a one-pixel camera!). Additionally the robot has two bump sensors that have binary outputs. That is, they output 1 when they strike an object and 0 otherwise. Finally, the robot has a whisker sensor that is also binary and outputs a 1 when it contacts something and 0 otherwise.

Before getting on to the task of figuring out how one might program this robot, we need to understand a bit about how the drive systems of these robots work. The configuration shown in Figure 13.3 is known as *differential drive*. We will be thinking much more systematically about differential drive in the first robot challenge, but for now let's work to understand it from a qualitative perspective.

Exercise 13.3

To build a qualitative understanding of differential drive it helps to understand a few limiting cases. Good ones to start with are ones that involve the wheels moving at equal speeds in either the forward (positive) or reverse (negative) direction. In these cases the robot will either move forward in a straight line or backwards in a straight line. In these cases the speed of the robot is directly proportional to the speed of its wheels. Now, let's consider cases where the velocities of the two wheels are unequal. To help you with your intuition it might help to imagine the right wheel pulling either forwards or backwards on the right side of the robot and the left wheel pulling either forwards or backwards on the left side of the robot. Here is a potential list of limiting cases to consider. Make predictions about what would happen in these cases. It may help to sketch a couple of key frames (poses of the robot) over time.

1. What if V_L is positive and $V_R = -V_L$? spin CW
2. What if V_R is positive and $V_L = -V_R$? spin CCW
3. What if $V_L = 0$ and V_R is positive? pivot CCW about L
4. What if $V_R = 0$ and V_L is positive? pivot CW about R
5. What if $V_L = 0$ and V_R is negative? pivot CW about L
6. What if $V_R = 0$ and V_L is negative? pivot CCW about R
7. What if V_R is positive and $V_L = \frac{1}{2}V_R$? veer left
8. What if V_L is positive and $V_R = \frac{1}{2}V_L$? veer right

Solution 13.1

Here are a few examples to get you started.

- A thermostat (2) and (3)
- R2D2 (3) (maybe also (2)?)
- A toaster oven (2) (most toasters don't really have a true control system)
- C3PO (1) and (3)
- Spot from Boston Dynamics (3) (maybe (2)?)

Solution 13.2

Sorry, no solution here! You all will come up with cooler things than we ever could.

Solution 13.3

1. The robot will rotate in place in the clockwise direction if you were looking down at the robot from above (i.e., spin to the right).
2. The robot will rotate in place in the counterclockwise direction (i.e., spin to the left).
3. The robot will rotate in the counterclockwise direction with the center of rotation at its left wheel.
4. The robot will rotate in the clockwise direction with the center of rotation at its right wheel.
5. The robot will rotate in the clockwise direction with the center of rotation at its left wheel.
6. The robot will rotate in the counterclockwise direction with the center of rotation at its right wheel.
7. The robot will rotate in the counterclockwise direction while also translating forward.
8. The robot will rotate in the clockwise direction while also translating forward.

Chapter 14

Robo Week 1b: Making a Robot Move

Schedule

14.1 Sense, Think, Act

One way to think about robots is to loosely break down their behaviors and subsystems into *sense*, *think*, and *act*. In the broadest terms, *sensing* involves acquiring information about the robot's surroundings, *thinking* means interpreting sensor input and making decisions about what to do next, and *acting* involves executing on those decisions.

In the robo module we will be working on all of these areas as we learn new conceptual material:

- **Sense:** We will use data from the Robot's LIDAR to identify walls and obstacles.
- **Think:** We will use our knowledge of functional covers and the location of obstacles to determine a safe path for the robot.
- **Act:** We will determine the appropriate wheel velocities needed for out robot to drive a planned path, and send the appropriate commands to the motors to achieve our objective.

14.2 Programming a Robot on a (Miro) Whiteboard

Today, you will be programming a vehicle to perform the behaviors you saw in the video of the real life Braitenberg vehicles (the one with wary, obsessive, etc.). Feel free to re-watch the real-life Braitenberg vehicles video. However, instead of programming the robot using a computer, you will be programming it on everyone's favorite piece of software: the Zoom! How can you program using a whiteboard (we imagine you conveniently might ask)?!? Remember, we are thinking of our robot's brain as a sensory-motor mapping. Translated into the language of mathematics this simply means that our robot program is specified by a function from sensors to motors!

There are many ways to represent a function. One way is as an equation. For instance, if we use the symbol ℓ to represent the reading of the robot's light sensor, then a robot that moves faster and faster as it see more and more light might have $V_L(\ell) = \ell^2$ and $V_R(\ell) = \ell^2$. Another way to represent a function is to define it graphically. You could draw a function that has ℓ on the x-axis and V_L on the y-axis. You could then sketch the relationship between those two quantities. Doing this graphically you could either have a quantitatively accurate sketch or a sketch that simply characterizes the function's qualitative behavior. For

sensors that have binary values, like the bump sensors, you can exhaustively enumerate all conditions. For instance, here is the program of a robot that drives forward at a $0.5\frac{m}{s}$ until it rams into something

$$V_L(bump_F) = \begin{cases} 0.5\frac{m}{s}, & bump_F = 0 \\ 0\frac{m}{s}, & bump_F = 1 \end{cases} \tag{14.1}$$

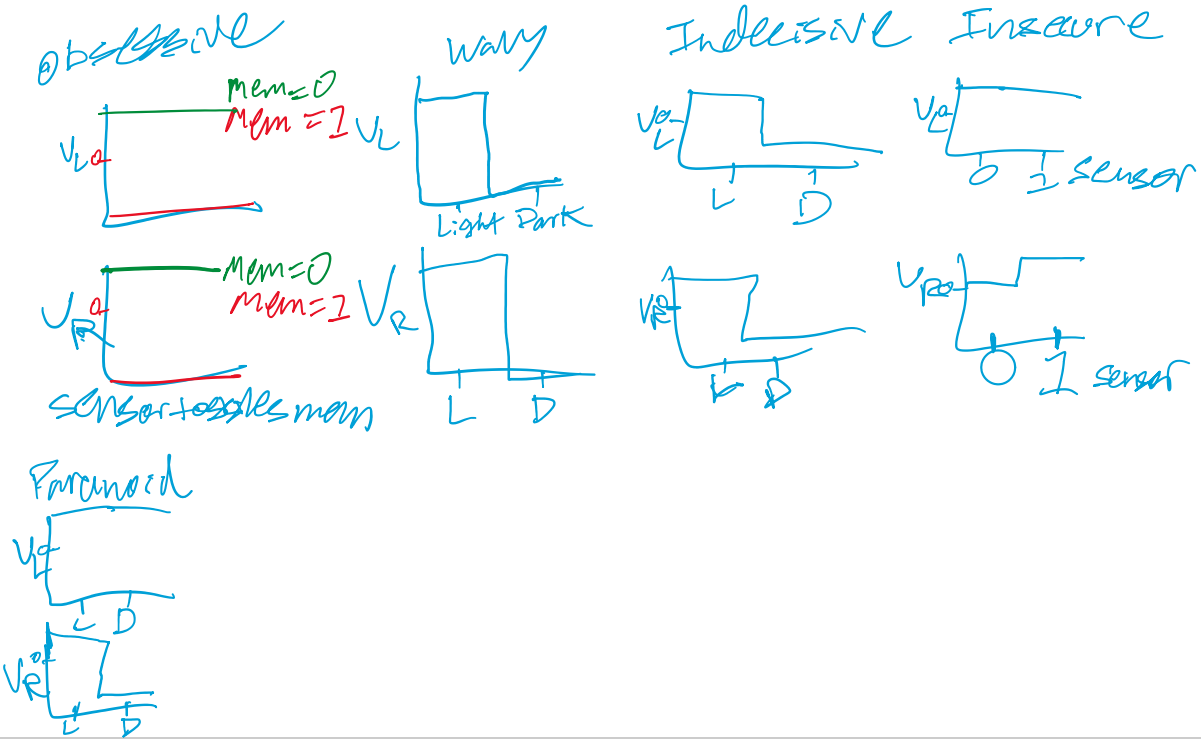
$$V_R(bump_F) = \begin{cases} 0.5\frac{m}{s}, & bump_F = 0 \\ 0\frac{m}{s}, & bump_F = 1 \end{cases} \tag{14.2}$$

where $bump_F$ is the value of the forward bump sensor (1 when in contact with something, 0 otherwise).

Exercise 14.1

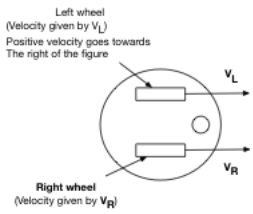
Work through generating robot programs to realize the behaviors in the video of the real life Braitenberg vehicles (obsessive, wary, indecisive, paranoid, insecure). Before jumping in, read the questions below and make sure to look at the template on the next page.

1. In order to validate your proposed program, it helps to do a quick whiteboard simulation. Sketch out a few key instants in time, what the robot's sensors would read, and what the wheel velocities would be. After the first couple you may be able to easily see whether or not your "program" is correct without actually sketching the key frames. **If this seems a little vague, check out the template on the next page, which we have made to help scaffold this activity. If it is still vague, check out the sample solution for the "wary" behavior.**
2. At least one of the behaviors cannot be reproduced without some primitive form of memory (although perhaps if you are very creative it can work). Which behaviors are these? How can you tell?



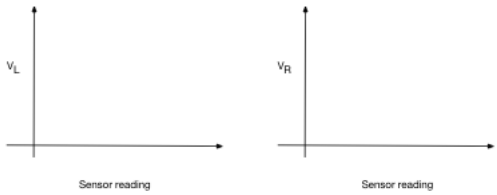
This template is designed for annotating in Zoom

Step 1: Label your motors and sensors



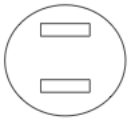
Step 2: Define your Sensory-Motor Mapping

You could do this graphically using the provided axes, or as a function (as shown in the document)

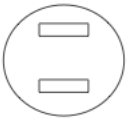


Step 3: Sketch some key frames by showing where the robot is in its environment, its sensor readings, and its resultant velocity

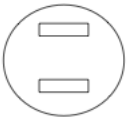
Time 1



Time 2



Time 3



Exercise 14.2

Next, Imagine your robot can remember a small amount of information. Specifically, your robot has access to a single flag that starts out with the value 0. Its value can be toggled from 0 to 1 or from 1 to 0 when a particular event occurs. For instance, if the light sensor reads a certain value, the robot might toggle its flag. The value of the flag can then inform the behavior of the robot. Given this new capability, implement any behaviors in the video that you couldn't before.

If you get done early and you're feeling excited about this, consider adding an additional light sensor to your robot. Now that you have two light sensors, you can get the robot to do a richer set of behaviors. Sketch the configuration of sensors on your new Braitenberg vehicle (equipped with two light sensors). Try to reproduce behaviors such as light seeking and light avoiding. For more ideas see the Wikipedia page on Braitenberg vehicles.

14.3 The Motion of Rigid Bodies

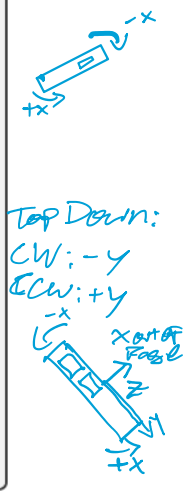
Our first challenge in Robo Ninja Warrior is going to involve the motion of rigid bodies (a rigid body just means an object that doesn't change shape as it moves around). You're probably all familiar with the idea of the velocity of a body from your previous coursework in physics, but you probably haven't thought too much about the concept of the angular velocity of a rigid body. The angular velocity of rigid bodies will turn out to be important for programming our robot to move along precisely defined paths. In this section, you'll be starting to get a feel for the concept.

We are going to explore the concept of angular velocity using an "app" for your phone. Please download and install the "phyphox" app to your phone, and open the "Gyroscope sensor."

Exercise 14.3

Qualitative: For each of the questions below, you should "plot" the data on the board, and interpret the data qualitatively in terms of the coordinate system of the phone.

1. Place your phone on the table and spin it, in place, counterclockwise. Note that you should be spinning it about an axis that is orthogonal to the face of the phone. Which axis is this on the data graph? What happens if you spin it clockwise? Z, CCW = −, CW = +
2. Now spin the phone about the two other axes. Which axis is which? Which direction is clockwise and which is counterclockwise? Clearly draw the coordinate system the phone is using - use the unit vectors $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$.
3. What happens if you move the phone along a straight line (on the table) without turning the phone? NO response on graphs
4. What happens if you move the phone uniformly in a circle **without turning the phone**? i.e. the orientation of the phone does not change. Acc=0 nothing
5. What happens if you move the phone uniformly in a circle **while turning the phone at the same time**? i.e. imagine the phone is a car and you are driving in a circle. Graphs change



Now that we've explored angular velocity using the phone, let's make it quantitative. We will use the following notation for the angular velocity

$$\boldsymbol{\omega} = \omega_x \hat{\mathbf{x}} + \omega_y \hat{\mathbf{y}} + \omega_z \hat{\mathbf{z}}$$

so that ω_x is the component of angular velocity corresponding to rotation about the x-axis and so on. The units of angular velocity are in radians per second, e.g., rotating in a complete circle in one second would be a 2π radian/second rotation.

Exercise 14.4

Quantitative: For each of the questions below, you should "plot" the data on the board (please interpret these quotes as meaning "sketch" what you think the data would look like, don't start writing MATLAB code), and interpret the data quantitatively in terms of the coordinate system of the phone.

1. Predict the angular velocity if you place the phone down on the table and then spin the phone in place so that it spins once in 5 seconds. Confirm your prediction using the data. ✓ $\omega_z = \frac{2\pi}{5}$, $\omega_x = \omega_y = 0$
2. Predict the angular velocity if you uniformly move the phone in a circle in 5 seconds (while turning it as in the last step of the previous problem), and confirm your prediction using the data. Does it matter how large the circle is? same as #1, no ✓
3. What type of motion would give rise to a constant ω_z of 2 radians per second for 5 seconds, with both $\omega_x = 0$ and $\omega_y = 0$? Confirm your prediction with the phone. #1 but faster
4. What type of motion would give rise to a sinusoidal ω_z with amplitude of 2 radians per second, and a period of 10 seconds, with both $\omega_x = 0$ and $\omega_y = 0$. Moving the phone at varying speeds
5. What type of motion would give rise to a constant ω_z and ω_x of 2 radians per second for 5 seconds, with $\omega_y = 0$. Confirm your prediction with your phone.

6. Sketch a graph of your own choosing of ω_x, ω_y, and ω_z and challenge yourself to produce it using the phone!

Chapter 15

Robo Homework 1: Parametric Curves and Motion

Contents

Learning Objectives

By the end of this assignment, you should feel confident with the following: *Concepts*

- Visualizing parametric curves.
- Defining vector functions for a given curve.
- Computing relevant unit vectors like tangent, normal, and binormal.
- Computing the curvature and torsion of a curve.
- Computing the length of a curve.
- Interpreting motion in terms of these concepts.

MATLAB skills

- Alter MATLAB starter code for a unit circle for the general case of a circle or ellipse.
- Compute the curvature and torsion of a curve using the symbolic math toolbox.
- Compute the length of a curve using the symbolic math toolbox.

Overview and Orientation

In Module 1 we introduced the concept of parametric curves. We are now going to return to this subject, but in a more general framework using vectors.

This assignment draws from material in multi-variable and vector calculus, and any textbook in these subjects will have related material. Keywords include **parametric curves**, **curve length**, and **line integral**. Good sources include **Paul's Online Math Notes**—the section on Calculus III. The relevant **Khan** videos are also useful.

Visualizing parametric curves and associated quantities like tangent vectors etc is a critical part of developing an understanding of the connection between parametric curves and the motion of moving bodies. Linda Vanasupa has created a video about using **CalcPlot3D** to visualize parametric curves—it is available here.

Although you can evaluate the derivatives and integrals in this assignment by hand, we would also like you to use the **Symbolic Toolbox in MATLAB** for this purpose—we will be using this toolbox throughout this robotics module. You will find a starter MATLAB script here that uses symbolic Matlab to work through the circle example below.

15.1 Parametric Curves

In Module I we considered curves in the plane, represented by either an explicit function, $y = f(x)$ or $x = f(y)$, an implicit function $f(x, y) = 0$, or a set of parametric equations

$$x = f(u), y = g(u)$$

where we treat u as a parameter. Each value of u defines a point $(f(u), g(u))$ which we can plot. If we collect all the points defined by $u \in [a, b]$, then we get a parametric curve. In Module 1, we did not limit ourselves to curves in the plane. For example, in 3D we defined

$$x = f(u), y = g(u), z = h(u)$$

and the collection of points so defined trace out a curve in 3D.

An alternative to these coordinate definitions involves representing each point with a position vector, $\mathbf{r}(u)$. Since the position vector depends on a single parameter u, the end of the position vector traces out a curve in space. If we limit ourselves to 3D, we will usually use the following notation

$$\mathbf{r}(u) = x(u)\hat{\mathbf{i}} + y(u)\hat{\mathbf{j}} + z(u)\hat{\mathbf{k}}, \ u \in [a, b]$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ are the standard Cartesian unit vectors. In a sense the vector function $\mathbf{r}(u)$ lifts the interval $[a, b]$ and transforms it in order to produce a curve in 3D space.

One major advantage of this notation is that we can take derivatives of this vector function with respect to the parameter u

$$\begin{aligned} \mathbf{r}'(u) &= \frac{d}{du}\left(x(u)\hat{\mathbf{i}} + y(u)\hat{\mathbf{j}} + z(u)\hat{\mathbf{k}}\right), \\ &= x'(u)\hat{\mathbf{i}} + y'(u)\hat{\mathbf{j}} + z'(u)\hat{\mathbf{k}}, \end{aligned}$$

since the Cartesian unit vectors are constant. We can interpret the derivative as follows: for any given value of u this vector is tangent to the parametric curve. At times we might be more interested in a unit tangent vector $\hat{\mathbf{T}}$, which we can obtain by normalizing the derivative

$$\hat{\mathbf{T}} = \frac{\mathbf{r}'}{|\mathbf{r}'|}$$

The trace of the tangent vector tip as u varies, $\hat{\mathbf{T}}(u)$, also produces a curve; taking its derivative ($\frac{d}{du}$), produces a vector, tangent to $\hat{\mathbf{T}}(u)$, which we will define as the normal vector to the original curve, $\mathbf{r}(u)$. The unit normal vector $\hat{\mathbf{N}}$ is therefore

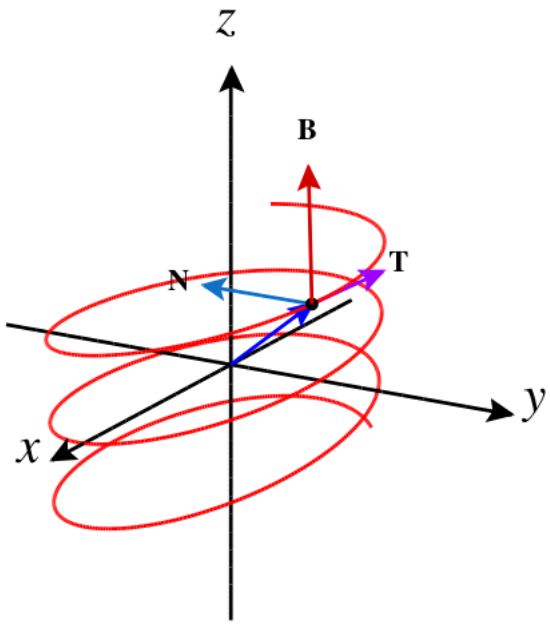
$$\hat{\mathbf{N}} = \frac{\hat{\mathbf{T}}'}{|\hat{\mathbf{T}}'|}$$

Finally, we can use both the unit tangent vector and the unit normal vector to define a unit binormal vector $\hat{\mathbf{B}}$ as follows

$$\hat{\mathbf{B}} = \hat{\mathbf{T}} \times \hat{\mathbf{N}}$$

Perp to both

Taken together, these three unit vectors are mutually orthogonal and therefore form an orthonormal basis of 3D space. This is known as the Frenet-Serret frame, and some applications can be found on the Wikipedia page concerning `Frenet-Serret Formulas`.



In addition to these unit vectors, parametric curves in 2D and 3D are often described in terms of their curvature κ and torsion τ. The curvature is the normalized rate of change of the unit tangent vector

$$\kappa = \frac{|\hat{\mathbf{T}}'|}{|\mathbf{r}'|}$$

?

and measures how quickly a curve is changing direction - a large value of the curvature means the curve is changing direction rapidly. The curvature is always non-negative. A straight-line would have zero curvature.

The torsion is the normalized rate of change of the unit binormal vector in the direction opposite to the unit normal

$$\tau = -\hat{\mathbf{N}} \cdot \frac{\hat{\mathbf{B}}'}{|\mathbf{r}'|}$$

and measures the rate at which a curve is twisting out of the plane - a large value of the torsion means the curve is rapidly twisting out of the plane. A curve in the plane has zero torsion. The torsion can be positive or negative, and convention dictates that a right-handed curve has positive torsion. As Wikipedia

says "Intuitively, curvature measures the failure of a curve to be a straight line, while torsion measures the failure of a curve to be planar."

Now that we know how to define a general parametric curve, we are ready to compute with it. For example, we could compute the length of the curve. In order to do so, let's lay down a set of points in the u-domain separated by Δu. Each point is mapped to the space curve, so the approximate length of each section of the curve is,

$$\Delta L = |\mathbf{r}'(u)|\Delta u$$

Refining this for smaller Δu and then summing up the pieces results in the integral

$$L = \int_a^b |\mathbf{r}'(u)|\, du$$

which defines the length of the curve.

15.1.1 The Parametric Circle

Let's take a few minutes to digest some of the theory by looking at a very common example—the parametric representation for a circle of radius R, centered at the origin in the xy-plane. If we define the position vector

$$\mathbf{r}(u) = R\cos u\,\hat{\mathbf{i}} + R\sin u\,\hat{\mathbf{j}},\ u \in [0, 2\pi]$$

then the circle is traced out once in the counterclockwise direction starting at $(R, 0)$. In this way, we can identify the parameter u as being the angle from the x-axis to a point on the circle.

Let's compute (by hand) the various unit tangent vectors, the curvature, the torsion, and the length of the curve. The first derivative of the position vector is

$$\mathbf{r}'(u) = -R\sin u\,\hat{\mathbf{i}} + R\cos u\,\hat{\mathbf{j}}$$

The unit tangent vector is

$$\begin{aligned}\hat{\mathbf{T}} &= \frac{\mathbf{r}'}{|\mathbf{r}'|} \\ &= -\sin u\,\hat{\mathbf{i}} + \cos u\,\hat{\mathbf{j}}\end{aligned}$$

and is tangent to the circle (you could check that $\mathbf{r}\cdot\hat{\mathbf{T}} = 0$). The unit normal vector is

$$\begin{aligned}\hat{\mathbf{N}} &= \frac{\hat{\mathbf{T}}'}{|\hat{\mathbf{T}}'|} \\ &= -\cos u\,\hat{\mathbf{i}} - \sin u\,\hat{\mathbf{j}}\end{aligned}$$

and is normal to the circle, pointing inward (you could check that $\hat{\mathbf{T}}\cdot\hat{\mathbf{N}}$). The unit binormal vector is

$$\begin{aligned}\hat{\mathbf{B}} &= \hat{\mathbf{T}}\times\hat{\mathbf{N}} \\ &= (-\sin u\,\hat{\mathbf{i}} + \cos u\,\hat{\mathbf{j}}) \times (-\cos u\,\hat{\mathbf{i}} - \sin u\,\hat{\mathbf{j}}) \\ &= (\sin^2 u + \cos^2 u)\,\hat{\mathbf{k}} \\ &= \hat{\mathbf{k}}\end{aligned}$$

and is out of the plane of the circle. The curvature of the circle is

$$\begin{aligned}\kappa &= \frac{|\hat{\mathbf{T}}'|}{|\mathbf{r}'|} \\ &= \frac{1}{R}\end{aligned}$$

and is inversely proportional to the radius of the circle. The torsion of the circle is

$$\begin{aligned}\tau &= -\hat{\mathbf{N}} \cdot \frac{\hat{\mathbf{B}}'}{|\mathbf{r}'|} \\ &= 0\end{aligned}$$

is zero because the circle is a plane curve and the binormal vector is a constant. For the length of the curve, we have

$$|\mathbf{r}'(u)| = +\sqrt{(-R)^2 \sin^2 u + (R)^2 \cos^2 u}$$

and

$$|\mathbf{r}'(u)| = R$$

and the integral becomes

$$L = \int_0^{2\pi} R du = 2\pi R$$

which is the circumference of a circle of radius R as expected!

Exercise 15.1

1. Define a vector function $\mathbf{r}(u)$ in the xy-plane whose trace is a circle centered at (x_0, y_0) with radius R.
2. Use **CalcPlot3D** to visualize the circle for different centers and radii, along with the unit tangent and normal vectors.
3. Determine (by hand) the unit tangent vector.
4. Determine (by hand) the unit normal vector.
5. Determine (by hand) the unit binormal vector.
6. Determine (by hand) the curvature and torsion. Does the curvature or torsion depend on the location of the circle center?
7. Set up the integral to compute the perimeter of the circle, and evaluate it by hand. (Given that it's a circle, what do you expect the length of the curve to be for u=0 to 2π?)

Exercise 15.2

1. A helix in 3D can be defined by the vector function

$$\mathbf{r}(u) = a\cos u\,\hat{\mathbf{i}} + a\sin u\,\hat{\mathbf{j}} + bu\,\hat{\mathbf{k}}, a > 0, b > 0, u \geq 0$$

   Use **CalcPlot3D** to visualize the helix for different centers and radii, along with the unit tangent, normal, and binormal vectors.
2. Determine (by hand) the unit tangent vector.
3. Determine (by hand) the unit normal vector.
4. Determine (by hand) the unit binormal vector.

$$= \frac{\sqrt{a^2\sin^2(v) + a^2\cos^2(v) + b^2}}{\sqrt{a^2(\sin^2 v + \cos^2 v) + b^2}}$$

$$= \frac{1}{\sqrt{a^2+b^2}}$$

$$= \frac{-a\sin(v)\hat{i} + a\cos(v)\hat{j} + b\hat{k}}{\sqrt{a^2+b^2}}$$

$$= \left(\frac{1}{\sqrt{a^2+b^2}}\right)\left(-a\sin(v)\hat{i} + a\cos(v)\hat{j} + b\hat{k}\right)$$

3) $\hat{N}(u) = \dfrac{\hat{T}'(u)}{|\hat{T}'(u)|} = \dfrac{\left(\frac{1}{\sqrt{a^2+b^2}}\right)(-a\cos(v)\hat{i} - a\sin(v)\hat{j})}{\sqrt{\left(\frac{-a}{\sqrt{a^2+b^2}}\right)^2 + \left(\frac{-a}{\sqrt{a^2+b^2}}\right)^2}}$

$$= \frac{1}{\frac{-a}{\sqrt{a^2+b^2}}\sqrt{2}}$$

$$= \left(\frac{1}{\sqrt{a^2+b^2}}\right)\left(\frac{\sqrt{a^2+b^2}}{-a}\right)\frac{a\cos(v)\hat{i} + a\sin(v)\hat{j}}{\sqrt{2}}$$

$$= \left(\frac{1}{\sqrt{2}}\right)(\cos(v)\hat{i} + \sin(v)\hat{j})$$

4) $\hat{B}(v) = \hat{T}(v) \times \hat{N}(v) = |\hat{T}(v)||\hat{N}(v)|\sin(\theta)n$

$$= |\hat{T}(v)||\hat{N}(v)|(1)k$$

$$|\hat{T}(v)| = \sqrt{\left(\frac{-a}{\sqrt{a^2+b^2}}\right)}$$

5. Determine (by hand) the curvature and torsion.
6. Set up the integral to compute the length of the helix corresponding to 5 complete turns, and evaluate it (by hand).

The helix is a great example of a curve in 3D, and shows up in all sorts of places. Visualizing the unit vectors associated with the curve is more challenging because they live in 3D, and we are asking you to define the domain so that the helix completes 5 turns. You should also find that the curvature and torsion of the helix are constant, and indeed any space curve with constant curvature and torsion is an helix. We often define a helix in terms of its radius a, and its pitch $2\pi b$, which is the height of the helix after one complete turn.

15.1.2 Using Symbolic Solvers

As the parametric functions that define our curves get more complicated, finding the various vectors and properties of the curve by hand can become tedious. Instead, we will rely on symbolic solvers to perform the calculations. In QEA we will primarily use the **Symbolic Toolbox** in Matlab for these operations.

Exercise 15.3

Download the Matlab script here that walks through solving the parametric circle example using Matlab's Symbolic Toolbox. **Note:** You can save these files as Matlab Live Scripts (.mlx) which can make the symbolic output easier to read.

1. Review the syntax used in the script for creating symbols, symbolic functions, and performing symbolic operations.
2. Modify the script for the circle located at (x_o, y_o).

Exercise 15.4

Define a vector function $\mathbf{r}(u)$ in the xy-plane whose trace is an ellipse centered at (x_0, y_0) with semi-major axis a, and semi-minor axis b, $b < a$.

1. Use **CalcPlot3D** to visualize the ellipse for different centers and semi-major axes, along with the unit tangent, normal, and binormal vectors.
2. Determine the unit tangent vector using the **Symbolic Toolbox** in MATLAB.
3. Determine the unit normal vector using the **Symbolic Toolbox** in MATLAB.
4. Determine the unit binormal vector using the **Symbolic Toolbox** in MATLAB..
5. Determine the curvature and torsion using the **Symbolic Toolbox** in MATLAB.
6. Set up the integral to compute the perimeter of the ellipse using the **Symbolic Toolbox** in MATLAB, and numerically evaluate it for a specific case of $(x_0, y_0), a, b$.

You will find that the integral for the perimeter of the ellipse involves elliptic integrals which cannot be evaluated using elementary functions. Check out this page at the American Mathematical Society for an interesting review.

15.2 Motion of Bodies

So far we've been talking about the intrinsic geometry of curves. However, there is an intimate connection between the geometry of curves and the motion of bodies. Assume a body is moving in space and is described by a position vector $\mathbf{r}(t)$ defined in terms of a fixed frame. In component form we write

$$\mathbf{r}(t) = x(t)\hat{\mathbf{i}} + y(t)\hat{\mathbf{j}} + z(t)\hat{\mathbf{k}}$$

where t is time. The units of $\mathbf{r}(t)$ are length.

The derivative with respect to time of this position vector defines the velocity of the body

$$\mathbf{v}(t) = \mathbf{r}'(t) = x'(t)\hat{\mathbf{i}} + y'(t)\hat{\mathbf{j}} + z'(t)\hat{\mathbf{k}}$$

and the second derivative with respect to time of this position vector defines the acceleration of this body

$$\mathbf{a}(t) = \mathbf{r}''(t) = x''(t)\hat{\mathbf{i}} + y''(t)\hat{\mathbf{j}} + z''(t)\hat{\mathbf{k}}$$

It will be instructive now to ask how the velocity vector and acceleration vector are oriented with respect to the path of the body as it moves through space. If we treat t as a parameter, and view the motion of the body as a parametric curve, then we can use the machinery of parametric curves to answer this question.

Let's start with the velocity vector. We know from our earlier work that the derivative $\mathbf{r}'(t)$ is tangent to the curve, which implies that the velocity must be tangent to the curve. Recall that earlier we defined the unit tangent to be

$$\hat{\mathbf{T}}(t) = \frac{\mathbf{r}'(t)}{|\mathbf{r}'(t)|}$$

Re-arranging this definition we can express the velocity in terms of the unit tangent vector $\hat{\mathbf{T}}$

speed ← Mag of tan line $\quad \mathbf{v}(t) = v(t)\hat{\mathbf{T}}(t)$ ← Vel = speed · tan unit

where $v(t) = |\mathbf{r}'(t)|$ is the linear speed of the body in the tangent direction, and we explicitly note that $\hat{\mathbf{T}}(t)$ is a function of t, i.e. the unit tangent direction changes as we move along the curve.

What about the acceleration $\mathbf{a}(t)$? If we take the derivative of the velocity we see that

$$\begin{aligned} \mathbf{a}(t) &= \frac{d}{dt}\left(v(t)\hat{\mathbf{T}}(t)\right) \\ &= \frac{dv}{dt}\hat{\mathbf{T}} + v\frac{d\hat{\mathbf{T}}}{dt} \end{aligned}$$

because both the magnitude and direction of velocity can change in time, and we need to use the product rule for derivatives. Recall from earlier that the rate of change of the unit tangent vector is related to the unit normal vector

$$\frac{d\hat{\mathbf{T}}}{dt} = |\hat{\mathbf{T}}'|\hat{\mathbf{N}}$$

and the magnitude of the rate of change of the unit tangent vector is related to the curvature

$$|\hat{\mathbf{T}}'| = \kappa|\mathbf{r}'|$$

Since $|\mathbf{r}'| = v$ we have

$$\frac{d\hat{\mathbf{T}}}{dt} = \kappa v \hat{\mathbf{N}}$$

so that the acceleration becomes

$$\mathbf{a}(t) = \frac{dv}{dt}\hat{\mathbf{T}} + \kappa v^2 \hat{\mathbf{N}}$$

Let's pause for a moment to consider this. We have expressed the acceleration of a moving body in terms of the unit tangent vector and the unit normal vector, so we will often talk about the tangential acceleration and normal acceleration of a body. You might recall that because velocity is a vector, it can accelerate by changing in magnitude ("speed") or changing in direction. The magnitude of the tangential acceleration is just the rate of change of the linear speed. On the other hand, the magnitude of the normal acceleration is caused by the constant change in the direction of the velocity vector as it moves around the circle. This normal acceleration is proportional to the square of the linear speed and the curvature of the path along which the body is moving and is directed normal to the curve.

15.2.1 The Circle Example

Consider the example of a body moving in a circle of radius R at some constant linear speed v. What is the position vector for such a body? Well, we know that it should look like a parametric circle, so let's define

$$\begin{aligned} x &= R\cos(\omega t) \\ y &= R\sin(\omega t) \end{aligned}$$

where ω is a variable that we need to define. In vector notation we have

$$\mathbf{r}(t) = R\cos(\omega t)\hat{\mathbf{i}} + R\sin(\omega t)\hat{\mathbf{j}}$$

Using the approach from the previous section we can compute the unit tangent vector and unit normal vector to this curve,

$$\hat{\mathbf{T}} = -\sin(\omega t)\hat{\mathbf{i}} + \cos(\omega t)\hat{\mathbf{j}}$$

$$\hat{\mathbf{N}} = -\cos(\omega t)\hat{\mathbf{i}} - \sin(\omega t)\hat{\mathbf{j}}$$

The velocity vector is

$$\mathbf{v}(t) = -R\omega\sin(\omega t)\hat{\mathbf{i}} + R\omega\cos(\omega t)\hat{\mathbf{j}}$$

which we expect to be tangential to the curve. One way to check is to find the tangential and normal components of velocity by evaluating $\mathbf{v}\cdot\hat{\mathbf{T}}$ and $\mathbf{v}\cdot\hat{\mathbf{N}}$. We obtain

$$\mathbf{v}\cdot\hat{\mathbf{T}} = R\omega$$

$$\mathbf{v}\cdot\hat{\mathbf{N}} = 0$$

which means that the velocity can be expressed as

$$\mathbf{v}(t) = R\omega\hat{\mathbf{T}}$$

We should now recognize ω as the angular velocity of the body as it moves in uniform circular motion. The circular motion is uniform because it moves a constant number of radians per second. The acceleration vector is

$$\mathbf{a}(t) = -R\omega^2\cos(\omega t)\hat{\mathbf{i}} - R\omega^2\sin(\omega t)\hat{\mathbf{j}}$$

Since the linear speed of motion is constant ($v = R\omega$) we would expect the acceleration to be in the normal direction only. Again we can compute the tangential and normal components of acceleration by evaluating $\mathbf{a}\cdot\hat{\mathbf{T}}$ and $\mathbf{a}\cdot\hat{\mathbf{N}}$. We obtain

$$\mathbf{a}\cdot\hat{\mathbf{T}} = 0$$

$$\mathbf{a}\cdot\hat{\mathbf{N}} = R\omega^2$$

Normal accel changes direction
tangential accel changes speed

which means that the acceleration can be expressed as

$$\mathbf{a}(t) = R\omega^2\hat{\mathbf{N}}$$

A body in uniform circular motion has no tangential component of acceleration - it is purely normal. Using the earlier expression for the angular velocity we could just as easily write the normal component of the acceleration as

$$a_N = \frac{v^2}{R}$$

which hopefully agrees with some results you saw a long time ago in school. It also connects to our earlier expression since the normal component of acceleration should be κv^2, and $\kappa = \frac{1}{R}$ for a circle.

Exercise 15.5

Consider a body undergoing non-uniform circular motion with radius R so that it's angular velocity is linearly increasing with time, $\omega = \alpha t$, i.e. the body starts off at rest and speeds up. Assume position is measured in meters and time is measured in seconds.

1. Define a position vector for this moving body. How long should it take the body to make its first loop? How about its second loop? $r(t) = R\cos(\alpha t^2)\mathbf{i} + R\sin(\alpha t^2)\mathbf{j}$ $2\pi = \alpha t^2$, $t = \sqrt{\frac{2\pi}{\alpha}}$
2. Visualize the motion in CalcPlot3D. ✓
3. Determine (by hand) the velocity of this moving body. How does the linear speed depend on R and α?
4. Determine (by hand) the acceleration of this moving body, and decompose the acceleration into the unit tangent and unit normal directions. Explain the result.

   You've probably seen and thought about uniform circular motion before. Often it is easier to understand a concept by looking at one just a little more complicated, so here we ask you to consider non-uniform circular motion. Although the path is the same curve, the motion is different.

Exercise 15.6

Helix. a relates to diameter, b relates to height, c relates to speed (vs. time)

1. Consider a body moving in 3D with position vector
$$\mathbf{r}(t) = a\cos(ct)\hat{\mathbf{i}} + a\sin(ct)\hat{\mathbf{j}} + bct\hat{\mathbf{k}}$$
   Assume that the position is measured in meters and time is measured in seconds. Describe the path that the body should take. Interpret the variables a, b, and c.
2. Visualize the motion in CalcPlot3D.
3. Determine (by hand) the velocity of this moving body. How does the linear speed depend on a, b, and c?
4. Determine (by hand) the acceleration of this moving body, and decompose the acceleration into the unit tangent and unit normal directions.

   Here we have a body moving in 3D on a curve that we have seen before. Pay attention to the direction of the unit tangent and unit normal as the body moves along this curve.

15.5.3) $r'(t) = v(t) = -R\sin(\alpha t^2)(2\alpha t)\mathbf{i} + R\cos(\alpha t^2)(2\alpha t)\mathbf{j} = -2\alpha R\sin(\alpha t^2)\mathbf{i} + 2\alpha R\cos(\alpha t^2)\mathbf{j}$

15.5.4) $v'(t) = a'(t) = -2\alpha R(\sin(\alpha t^2) + 2\alpha t^2\sin(\alpha t^2))\mathbf{i} + 2\alpha R(\cos(\alpha t^2) - (2\alpha)t^2\cos(\alpha t^2))\mathbf{j}$

Exercise 15.7

Consider a body moving with the following position vector

$$\mathbf{r}(t) = a\cos(ct)\mathbf{i} + b\sin(ct)\mathbf{j}$$

where $a > 0$, $b > 0$, and $c > 0$. Assume that position is measured in meters and time in seconds.

1. Describe the path that the body takes. How long does it take to return to its starting position?
2. Visualize the motion in CalcPlot3D. Circular Motion, $\frac{2\pi}{c}$
3. Use the Symbolic Toolbox in MATLAB to determine the velocity of this moving body. How does the linear speed depend on a, b, and c? $v(t) = -ac\sin(ct)\mathbf{i} + bc\cos(ct)\mathbf{j}$
4. Use the Symbolic Toolbox in MATLAB to determine the acceleration of this moving body, and decompose the acceleration into the unit tangent and unit normal directions.
   $a(t) = -ac^2\cos(ct)\mathbf{i} - bc^2\sin(ct)\mathbf{j}$
   Here we have a body moving on a path that corresponds to a curve that we studied earlier, and so in many ways this is nothing new. However, the motion of a body consists of both the path and "how" it moves along it. Interpreting the velocity and acceleration in terms of the variables a, b, and c will help you build your understanding of these concepts.

$a_N = \frac{abc^2}{\sqrt{a^2\sin^2(ct) + b^2\cos^2(ct)}}$

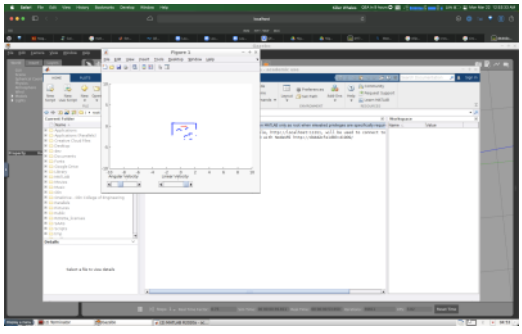
$a_T = \frac{(2\sin(2ct))(a^2 - b^2)}{2\sqrt{-a^2\cos^2(ct) + a^2 + b^2\cos^2(ct)}}$

15.3 Simulator Setup

We will begin using the Robo simulator during week 2, and we want to make sure that everyone has it working.

Exercise 15.8

1. Work through the simulator setup steps on the Meeto Your Neato page.
2. Make sure the simulator is operating correctly by running the "teleopAndVisualizer" script and driving the robot using your keyboard.
3. Take a screenshot of the MATLAB figure (LIDAR output), and include it with your homework submission.



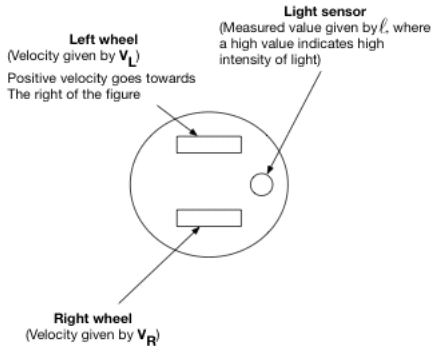
I'd like to register my amusement with the fact that I'm running a simulation inside MATLAB inside a Container inside a VM inside (sort of) MATLAB.

And they say kids these days are "spoiled by powerful computers"

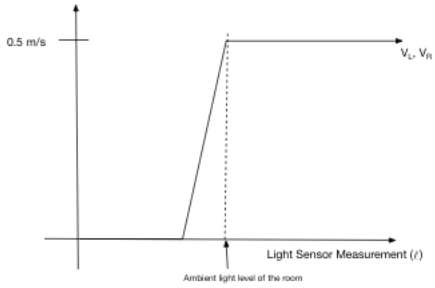
Solution 14.1

1. Here is a potential solution for the *wary* behavior. You could also do this one similarly to the example given above by writing out the functions as in Equations 14.1-14.2, but we wanted to show you an example of writing the "program" graphically.

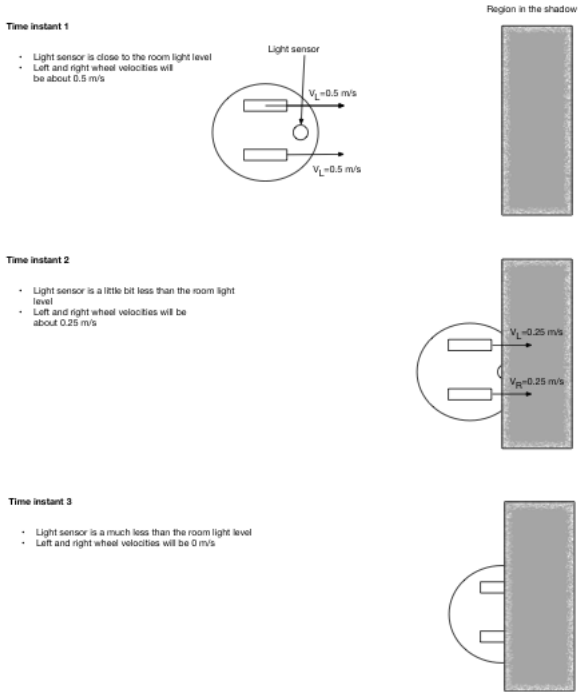
   First, sketch the robot and label the sensors (in this case the light sensor) and its actuators (the wheels). Assign notation to relevant quantities.



   Write down V_L and V_R as functions of the light sensor reading ℓ.



   Sketch some keyframes to build confidence in your solution.



2. If you are strict about defining the velocity just based on the sensor readings (and not on the current velocities of the wheels), then the obsessive behavior would be impossible to implement without some sort of memory. The reason would be that when you are halfway between the two shoes the robot wouldn't know whether to go left or right (since the sensor of "no bump detected" would be the same in either case. If you imagine the robot's motor system working a bit differently (e.g., maintaining the current velocity unless otherwise commanded), then you could argue that no memory is needed.

Solution 14.2

For the obsessive behavior you could do something like.
Start with $flag = 0$.

$$V_L(bump_F, bump_R, flag) = \begin{cases} -0.5\frac{m}{s}, & bump_F = 1 \\ 0.5\frac{m}{s}, & \text{else if } bump_R = 1 \\ 0.5\frac{m}{s}, & \text{else if } flag = 0 \\ -0.5\frac{m}{s}, & \text{else if } flag = 1 \end{cases} \tag{14.3}$$

$$V_R \text{ same as } V_L \tag{14.4}$$

We would also need to update the flag whenever a bump occurs by changing it from a 1 to a 0 or vice-versa.

Solution 14.3

1. The graph shows a positive angular velocity int the z-direction. When you spin it clockwise, the sign of the angular velocity becomes negative.
2. On my iPhone the positive y-axis runs along the long axis of the phone away from the bottom of the phone (where the home button would be on older models). I can tell this since a counterclockwise rotation about this axis yields a positive angular velocity in the y-direction. Using similar logic, I can tell that my phone's positive x-axis points to the right, across the short edge of the phone screen. You can see this in the iOS documentation using this link: https://developer.apple.com/documentation/coremotion/getting_raw_gyroscope_events.
3. The gyroscope reads values close to 0 for all axes.
4. The gyroscope reads values close to 0 for all axes.
5. The gyroscope reads a relatively constant value in the z-direction. The magnitude of the value corresponds to how long it takes to go around the circle (radius does not matter) and the sign tells us which direction we are traveling around the circle.

Solution 14.4

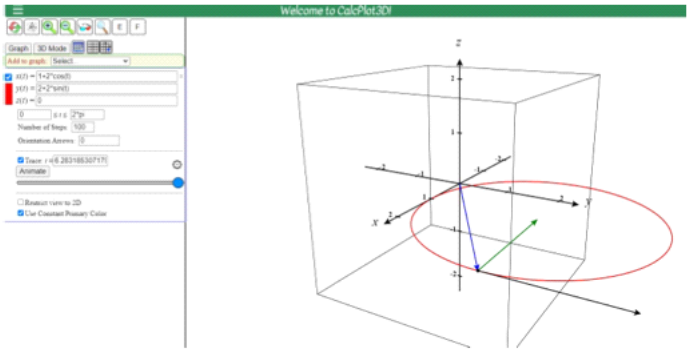
1. The angular velocity should be approximately $\frac{2\pi}{5}$ radians/s $\hat{\mathbf{z}}$ (assuming we spin it counter clockwise).
2. The angular velocity should be approximately $\frac{2\pi}{5}$ radians/s $\hat{\mathbf{z}}$ (assuming we move counter-clockwise around the circle). The radius of the circle doesn't matter.
3. Any motion that involves rotating the phone purely about $\hat{\mathbf{z}}$ (the screen is aligned to $\hat{\mathbf{z}}$) at the specified rate. An example would be moving the phone uniformly around a circle a total of $\frac{5s \times 2\ radians/s}{2\pi radians/revolution} \approx 1.6\ revolutions$ in 5 seconds (other motions will work as well, e.g., the linear motion can be anything you want).
4. While many motions could achieve this, one possibility is to spin the phone about its $\hat{\mathbf{z}}$ axis, starting with slow rotation, gradually ramping up in speed, then down in speed, reversing direction, getting faster, slowing down, and ultimately reversing direction. This complete cycle should take 10 seconds.
5. Spinning the phone around an axis pointing out of the phone at an angle of 45 degrees to the right (good luck actually doing this!). The phone should rotate once in about 2.25 seconds.
6. Let's see what you can come up with!

Solution 15.1

1. Define a vector function $\mathbf{r}(u)$ in the xy-plane whose trace is a circle centered at (x_0, y_0) with radius R. The equation for a circle centered at (x_0, y_0) with radius R is:

$$\mathbf{r}(u) = (x_0 + R\cos u)\mathbf{i} + (y_0 + R\sin u)\mathbf{j}$$

2. Use **CalcPlot3D** to visualize the circle for different centers and radii, along with the unit tangent and normal vectors. In **CalcPlot3D** you can use the "Space Curve: r(t)" option under the "Add to graph" menu with the settings shown below as an example.



3. Determine (by hand) the unit tangent vector.

$$\begin{aligned}
\mathbf{r}(u) &= (x_0 + R\cos u)\,\hat{\mathbf{i}} + (y_0 + R\sin u)\,\hat{\mathbf{j}} \\
\mathbf{r}'(u) &= -R\sin u\,\hat{\mathbf{i}} + R\cos u\,\hat{\mathbf{j}} \\
|\mathbf{r}'| &= \sqrt{R^2\sin^2 u + R^2\cos^2 u} = R \\
\hat{\mathbf{T}} &= \frac{\mathbf{r}'}{|\mathbf{r}'|} \\
\hat{\mathbf{T}} &= -\sin u\,\hat{\mathbf{i}} + \cos u\,\hat{\mathbf{j}}
\end{aligned}$$

4. Determine (by hand) the unit normal vector.

$$\begin{aligned}
\hat{\mathbf{N}} &= \frac{\hat{\mathbf{T}}'}{|\hat{\mathbf{T}}'|} \\
\hat{\mathbf{T}} &= -\sin u\,\hat{\mathbf{i}} + \cos u\,\hat{\mathbf{j}} \\
\hat{\mathbf{T}}' &= -\cos u\,\hat{\mathbf{i}} - \sin u\,\hat{\mathbf{j}} \\
|\hat{\mathbf{T}}'| &= \sqrt{(-\cos u)^2 + (-\sin u)^2} = 1 \\
\hat{\mathbf{N}} &= -\cos u\,\hat{\mathbf{i}} - \sin u\,\hat{\mathbf{j}}
\end{aligned}$$

5. Determine (by hand) the unit binormal vector.

$$\begin{aligned}
\hat{\mathbf{B}} &= \hat{\mathbf{T}} \times \hat{\mathbf{N}} \\
&= (-\sin u\,\hat{\mathbf{i}} + \cos u\,\hat{\mathbf{j}}) \times (-\cos u\,\hat{\mathbf{i}} - \sin u\,\hat{\mathbf{j}}) \\
&= (\sin^2 u + \cos^2 u)\,\hat{\mathbf{k}} \\
&= \hat{\mathbf{k}}
\end{aligned}$$

6. Determine (by hand) the curvature and torsion. Does the curvature or torsion depend on the location of the circle center? **Curvature:**

$$\begin{aligned}
\kappa &= \frac{|\hat{\mathbf{T}}'|}{|\mathbf{r}'|} \\
|\hat{\mathbf{T}}'| &= \sqrt{(-\cos u)^2 + (-\sin u)^2} = 1 \\
|\mathbf{r}'| &= \sqrt{R^2\sin^2 u + R^2\cos^2 u} = R \\
\kappa &= \frac{1}{R}
\end{aligned}$$

Torsion:

$$\begin{aligned}\tau &= -\hat{\mathbf{N}} \cdot \frac{\hat{\mathbf{B}}'}{|\mathbf{r}'|} \\ &= -(-\cos u\,\hat{\mathbf{i}} - \sin u\,\hat{\mathbf{j}}) \cdot \frac{0}{R} = 0\end{aligned}$$

No, we don't see x_o or y_o in the equations for curvature or torsion.

7. Set up the integral to compute the perimeter of the circle, and evaluate it by hand. (Given that it's a circle, what do you expect the length of the curve to be for u=0 to 2π?)

$$L = \int_a^b |\mathbf{r}'(u)|\,du = \int_0^{2\pi} \sqrt{R^2 \sin^2 u + R^2 \cos^2 u}\,du = 2\pi R$$

Solution 15.2

Solution 15.3

1. Review the syntax used in the script for creating symbols, symbolic functions, and performing symbolic operations.
2. Modify the script for the circle located at (x_o, y_o). An example modified script is here: link.

Solution 15.4

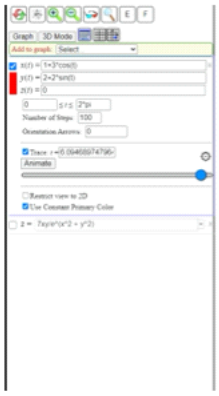
Define a vector function $\mathbf{r}(u)$ in the xy-plane whose trace is an ellipse centered at (x_0, y_0) with semi-major axis a, and semi-minor axis b, $b < a$.
A Matlab solution script can be found here: link.

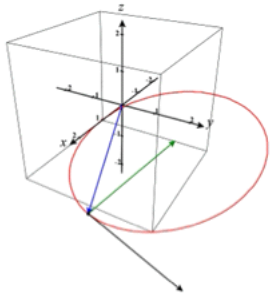
1. Use **CalcPlot3D** to visualize the ellipse for different centers and semi-major axes, along with the unit tangent, normal, and binormal vectors.

   The equation for an ellipse centered at (x_0, y_0) with semi-major axis a, and semi-minor axis b, $b < a$ is:

   $$\mathbf{r}(u) = (x_0 + a\cos u)\,\hat{\mathbf{i}} + (y_0 + b\sin u)\,\hat{\mathbf{j}} + 0\,\hat{\mathbf{k}}$$

   In **CalcPlot3D** you can use the "Space Curve: r(t)" option under the "Add to graph" menu with the settings shown below as an example.





2. Determine the unit tangent vector using the **Symbolic Toolbox** in MATLAB.

$$\hat{\mathbf{T}} = \frac{-a\sin u\,\hat{\mathbf{i}}}{\sqrt{a^2|\sin u|^2 + b^2|\cos u|^2}} - \frac{b\cos u\,\hat{\mathbf{j}}}{\sqrt{a^2|\sin u|^2 + b^2|\cos u|^2}} + 0\,\hat{\mathbf{k}}$$

3. Determine the unit normal vector using the **Symbolic Toolbox** in MATLAB.

$$\hat{\mathbf{N}} = \frac{-b\cos u\,\hat{\mathbf{i}}}{\sqrt{a^2\sin u^2 - b^2\sin u^2 + b^2}} + \frac{a\sin u\,\hat{\mathbf{j}}}{\sqrt{a^2\sin u^2 - b^2\sin u^2 + b^2}} + 0\,\hat{\mathbf{k}}$$

4. Determine the unit binormal vector using the **Symbolic Toolbox** in MATLAB.

$$\hat{\mathbf{B}} = 0\,\hat{\mathbf{i}} + 0\,\hat{\mathbf{j}} + 1\,\hat{\mathbf{k}}$$

5. Determine the curvature and torsion using the **Symbolic Toolbox** in MATLAB.

   Curvature:

$$\kappa = \frac{ab}{(a^2\sin u^2 - b^2\sin u^2 + b^2)^{3/2}}$$

   Torsion:

$$\tau = 0$$

6. Set up the integral to compute the perimeter of the ellipse using the **Symbolic Toolbox** in MATLAB, and numerically evaluate it for a specific case of $(x_0, y_0), a, b$.

   You will find that the integral for the perimeter of the ellipse involves elliptic integrals which cannot be evaluated using elementary functions. Check out this page at the American Mathematical Society for an interesting review.

Let x_o=0, y_o=0, a=1, and b=2.

$$L = \int_a^b |\mathbf{r}'(u)|\,du = 9.69$$

Solution 15.5

Consider a body undergoing non-uniform circular motion with radius R so that it's angular velocity is linearly increasing with time, $\omega = \alpha t$, i.e. the body starts off at rest and speeds up. Assume position is measured in meters and time is measured in seconds.

1. Define a position vector for this moving body. How long should it take the body to make its first loop? How about its second loop?

$$\mathbf{r}(t) = R\cos(\alpha t^2)\,\hat{\mathbf{i}} + R\sin(\alpha t^2)\,\hat{\mathbf{j}} + 0\,\hat{\mathbf{k}}$$

   We know the body will have completed one loop when $\alpha t^2 = 2\pi$. Therefore, $t_1 = \sqrt{\frac{2\pi}{\alpha}}$. Similarly, time to complete two loops is $\alpha t^2 = 4\pi$ and $t_2 = \sqrt{\frac{4\pi}{\alpha}}$ The time to complete just the second loop is $t_2 - t_1 = (2-\sqrt{2})\sqrt{\pi/\alpha}$. We expect the second loop to take less time than loop one.

2. Visualize the motion in CalcPlot3D.

3. Determine (by hand) the velocity of this moving body. How does the linear speed depend on R and α?

$$\begin{aligned}\mathbf{v}(t) &= \mathbf{r}'(t)\\ \mathbf{v}(t) &= -2Rt\alpha\sin(\alpha t^2)\,\hat{\mathbf{i}} + 2Rt\alpha\cos(\alpha t^2)\,\hat{\mathbf{j}} + 0\,\hat{\mathbf{k}}\end{aligned}$$

Find linear speed:

$$\begin{aligned}v(t) &= \sqrt{(-2Rt\alpha\sin(\alpha t^2))^2 + (2Rt\alpha\cos(\alpha t^2))^2}\\ v(t) &= 2Rt\alpha\end{aligned}$$

Linear speed increases with time t and is proportional to R and α.

4. Determine (by hand) the acceleration of this moving body, and decompose the acceleration into the unit tangent and unit normal directions. Explain the result.

You've probably seen and thought about uniform circular motion before. Often it is easier to understand a concept by looking at one just a little more complicated, so here we ask you to consider non-uniform circular motion. Although the path is the same curve, the motion is different.

$$\begin{aligned}\mathbf{a}(t) &= \mathbf{v}'(t)\\ \mathbf{v}(t) &= -2Rt\alpha\sin(\alpha t^2)\,\hat{\mathbf{i}} + 2Rt\alpha\cos(\alpha t^2)\,\hat{\mathbf{j}} + 0\,\hat{\mathbf{k}}\\ \mathbf{a}(t) &= -2R\alpha(\sin(\alpha t^2) + 2\alpha t^2\cos(\alpha t^2))\,\hat{\mathbf{i}} + 2R\alpha(\cos(\alpha t^2) - 2R\alpha t^2\sin(\alpha t^2))\,\hat{\mathbf{j}} + 0\,\hat{\mathbf{k}}\end{aligned}$$

$$\begin{aligned}\hat{\mathbf{T}}(t) &= \frac{\mathbf{r}'(t)}{|\mathbf{r}'(t)|}\\ \hat{\mathbf{T}}(t) &= -\sin(\alpha t^2)\,\hat{\mathbf{i}} + \cos(\alpha t^2)\,\hat{\mathbf{j}} + 0\,\hat{\mathbf{k}}\end{aligned}$$

$$\begin{aligned}\hat{\mathbf{N}}(t) &= \frac{\hat{\mathbf{T}}'(t)}{|\hat{\mathbf{T}}'(t)|}\\ \hat{\mathbf{N}}(t) &= -\cos(\alpha t^2)\,\hat{\mathbf{i}} - \sin(\alpha t^2)\,\hat{\mathbf{j}} + 0\,\hat{\mathbf{k}}\end{aligned}$$

$$\begin{aligned}a_N(t) &= \mathbf{a}(t)\cdot\hat{\mathbf{T}}(t) = 2R\alpha\\ a_T(t) &= \mathbf{a}(t)\cdot\hat{\mathbf{N}}(t) = 4Rt^2\alpha^2\end{aligned}$$

The tangential component of the acceleration is constant in time, but the normal component increases quadratically with time. Compare this to the case of circular motion at a constant speed which has zero tangential acceleration and constant normal acceleration.

Solution 15.6

Solution 15.7

Consider a body moving with the following position vector

$$\mathbf{r}(t) = a\cos(ct)\hat{\mathbf{i}} + b\sin(ct)\hat{\mathbf{j}}$$

where $a > 0$, $b > 0$, and $c > 0$. Assume that position is measured in meters and time in seconds. A solution script is here: link.

1. Describe the path that the body takes. How long does it take to return to its starting position?

   We know that this is an elliptical path with semi-major axis a and semi-minor axis b. We know that a complete orbit of the ellipse occurs when the parameter $ct = 2\pi$, so the time it takes to return to the starting position is $t = (2\pi)/c$.

2. Visualize the motion in CalcPlot3D.

   Setting for an ellipse in CalcPlot3D are above.

3. Use the Symbolic Toolbox in MATLAB to determine the velocity of this moving body. How does the linear speed depend on a, b, and c?

$$\mathbf{v}(t) = -ac\sin(ct)\,\hat{\mathbf{i}} + bc\cos(ct)\,\hat{\mathbf{j}} + 0\,\hat{\mathbf{k}}$$

$$v(t) = \sqrt{|ac\cos(ct)|^2 + |bc\cos(ct)|^2}$$

4. Use the Symbolic Toolbox in MATLAB to determine the acceleration of this moving body, and decompose the acceleration into the unit tangent and unit normal directions.

   Here we have a body moving on a path that corresponds to a curve that we studied earlier, and so in many ways this is nothing new. However, the motion of a body consists of both the path and "how" it moves along it. Interpreting the velocity and acceleration in terms of the variables a, b, and c will help you build your understanding of these concepts.

$$\begin{aligned}
\mathbf{a}(t) &= -ac^2\cos(ct)\,\hat{\mathbf{i}} - bc^2\sin(ct)\,\hat{\mathbf{j}} + 0\,\hat{\mathbf{k}} \\
\mathbf{a}(t)\cdot\hat{\mathbf{T}}(t) &= \frac{abc^2}{\sqrt{a^2\sin^2(ct) + b^2\cos^2(ct)}} \\
\mathbf{a}(t)\cdot\hat{\mathbf{N}}(t) &= \frac{c^2(a^2 - b^2)\sin(2ct)}{\sqrt{a^2\sin^2(ct) + b^2\cos^2(ct)}}
\end{aligned}$$