Week 2

Monday, March 22, 2021 10:25 AM

RoboWeek 2

Chapter 16

Robo Week 2a: Curves and Motion

Schedule

Learning Objectives

Concepts

- From $\mathbf{r}(t)$ data for a rigid body,
  - estimate average speed and distance traveled.
  - approximate $\hat{\mathbf{T}}$, $\hat{\mathbf{N}}$, and $\hat{\mathbf{B}}$ at various points on the space curve.
- From an $\mathbf{r}(t)$ parametric vector function for a rigid body,
  - approximate its space curve (i.e., path of travel)
  - compute the vector functions for $\mathbf{r}'(t)$ and $\mathbf{r}"(t)$
  - Compute $\hat{\mathbf{T}}$, $\hat{\mathbf{N}}$, and $\boldsymbol{\omega}$

Matlab Skills

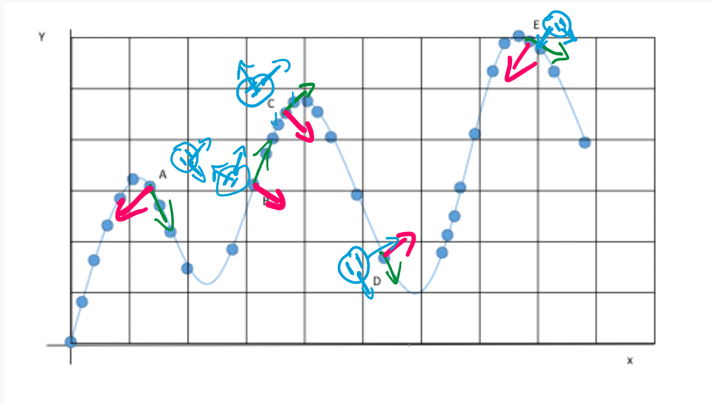
- Compute the angular velocity vector and its amplitude using the symbolic toolbox

16.1 The NEATO Goes for a Drive on Parametric Avenue

Here we will work to connect the work you did in week 1 to the upcoming challenge.

A hypothetical NEATO has just gone for a drive, and we have recorded its position vector $\mathbf{r}(t)$. Its path is shown below, and the dots indicate its position at equally-spaced points in time. (You can download a printable version of this image here). It starts in the bottom left corner and moves along the path until it reaches the last point. Let's assume that this is a relatively long drive - the grid spacing is 1m and the time between samples is 1s.

Exercise 16.1



1. Roughly how long does the NEATO travel for? Roughly how far does it travel? Roughly what is its average speed? $L = \int |r'(u)| dt$ 33s
2. Thinking about the curve only, draw the unit tangent vector $\hat{\mathbf{T}}$ and unit normal vector $\hat{\mathbf{N}}$ at each of the points indicated: A, B, C, D and E.
3. Thinking about the motion of the NEATO now, choose one of the points (A,B,C,D or E) and compute the magnitude of the velocity vector. Draw the velocity vector to scale at that point. Do the same for the acceleration vector, decomposing it into tangential and normal components. How do these relate to the unit tangent and unit normal vectors? (Do more points if you have time!) $|v| = 0.25 m/s$ $a_T \approx 0$ $a_N > 0$ but small
4. Draw a picture of the NEATO at A, B, C, D, and E, i.e. you are looking down on the NEATO as it drives along the curve.
5. The NEATO has its own internally defined coordinate system which is fixed to the robot (this is called a 'body-fixed' coordinate system). This coordinate system is used, among other things, for the readouts from the LIDAR scanner. The NEATO's coordinate system has the x-axis pointing forward, the y-axis pointing left, and the z-axis pointing up. For each of your NEATO pictures on the curve, indicate the orientation of the NEATO's own coordinate system. How does this relate to the $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ vectors? X is $\hat{T}$, y is same dir (but maybe flipped of) $\hat{N}$
6. In order for the NEATO to follow the curve, what must be true about the orientation of the NEATO's x-axis as it traverses the curve? What about its y-axis?
7. The NEATO is an example of a 'rigid body': an object which has a size (unlike a point particle, which you often consider in introductory physics) but for which the different parts of the body do not move with respect to one another (no stretching, bending, etc). When we study the motion of a rigid body, we can think about decomposing the motion into two components: the motion OF the center of the object and the motion ABOUT the center of the object. In other words, when we talk about the motion of the NEATO, in order to give a complete description, we need to specify the velocity of the center of the robot in the lab x direction, the velocity of the center of the robot in the lab y direction, and the rotational motion of the NEATO around its center relative to the x axis ($\theta = 0$ when the NEATO is going forward). This system has three degrees of freedom: x,y, and θ and we have to give the velocity for each!

(a) On your picture, indicate the orientation θ of the NEATO at each of the indicated points. Define $+\theta$ as the angle of the forward direction of the NEATO (NEATO's x axis) measured counter-clockwise from the lab x direction.

(b) The angular velocity $\boldsymbol{\omega}$ is a vector quantity, with magnitude

$$|\boldsymbol{\omega}| = \frac{d\theta}{dt}$$

and its direction is the axis of rotation. For an object constrained to move in the xy plane, we can express the angular velocity as

$$\boldsymbol{\omega} = \omega_z \hat{\mathbf{k}}$$

where $\omega_z = d\theta/dt$ is the rate of change of angle of orientation. What direction is the vector, $\boldsymbol{\omega}$, if the NEATO is curving to the right? What direction is the vector, $\boldsymbol{\omega}$, if the NEATO is curving to the left? (Recall: the angular velocity follows the right hand rule.) What characteristic vector of your parametric curve is also along this direction?

(c) If the object is constrained to always be oriented along its path (a NEATO that isn't slipping for example), then a consistent mathematical definition of angular velocity $\boldsymbol{\omega}$ is

$$\boldsymbol{\omega} = \hat{\mathbf{T}} \times \frac{d\hat{\mathbf{T}}}{dt}$$

Note: you'll see a derivation of this relationship in the night assignment, but for now let's just use it as a given. With this definition in mind, give a rough indication of the angular velocity vector of the NEATO at points A,B,C,D and E.

Exercise 16.2

Consider a NEATO that moves according to the following position vector

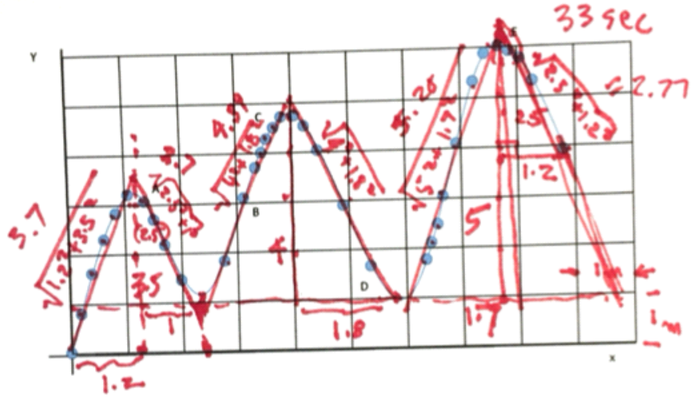
$$\mathbf{r}(t) = 0.05t\hat{\mathbf{i}} + 0.05t^2\hat{\mathbf{j}}$$

where $\mathbf{r}(t)$ has units of meters and t has units of seconds.

1. On the board, sketch the path that the NEATO moves along in 10 seconds, roughly indicating its location every second.
2. Approximate and sketch $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ at 1, 2, 5 and 10 seconds on your curve.
3. Compute by hand a vector function for $\hat{\mathbf{T}}$ as a function of time.
4. The direction of $\hat{\mathbf{N}}$ is defined by $\frac{d\hat{\mathbf{T}}}{dt}$. Compute $\frac{d\hat{\mathbf{T}}}{dt}$ (you can leave it as an un-simplified equation).
5. Use CalcPlot3D to visualize unit vectors ($\hat{\mathbf{T}}$, $\hat{\mathbf{N}}$ and $\hat{\mathbf{B}}$) for the 10 second period. How does each change during the period?
6. Derive the vector functions that represent the linear velocity and acceleration as the NEATO moves along the path.
7. Pull up the MATLAB symbolic starter code. Discuss how you would alter it to determine the angular velocity and visualize $\hat{\mathbf{T}}$, $\hat{\mathbf{N}}$ and $\hat{\mathbf{B}}$ as the NEATO travels the path.

Solution 16.1

1. Roughly how long does the NEATO travel for? Roughly how far does it travel? Roughly what is its average speed? By counting up the points and using 1 second between points, it traveled for 33 seconds. To determine the distance traveled, there are lots of ways to estimate these values. I turned the path into a series of triangles and computed the approximate length.

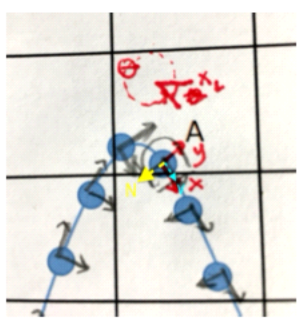


From this, I got 23.2 m. The average speed from these values is $0.7\overline{0}m/s$. The overline indicates that the "0" is significant. I used 2 significant figures since I only had 2 in the estimation of the triangles.

2. Thinking about the curve only, draw the unit tangent vector $\hat{\mathbf{T}}$ and unit normal vector $\hat{\mathbf{N}}$ at each of the points indicated: A, B, C, D and E. See the image below. $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ are in pencil.

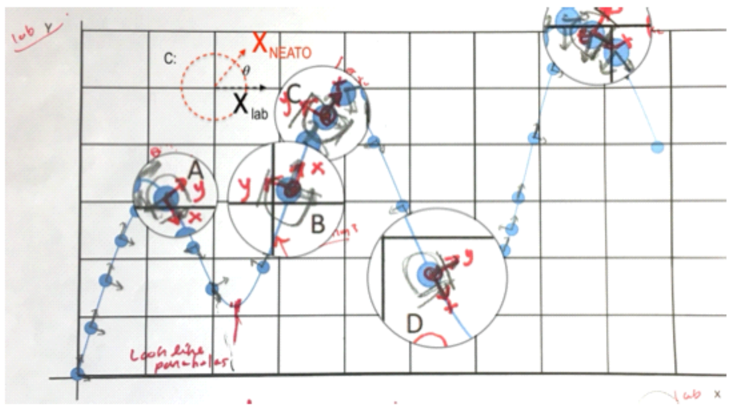
3. Thinking about the motion of the NEATO now, choose one of the points (A,B,C,D or E) and compute the magnitude of the velocity vector. Draw the velocity vector to scale at that point. Do the same for the acceleration vector, decomposing it into tangential and normal components. How do these relate to the unit tangent and unit normal vectors? (Do more points if you have time!) Point A: I'm going to use the points on the curve to estimate the ΔL over the 2 second interval in the point before and after A. Assuming that the grid is "to scale," I estimate the speed at A to be,

$$v = \frac{\Delta L}{\Delta t} \approx \frac{0.77}{2s} = 0.39m/s$$



4. Draw a picture of the NEATO at A, B, C, D, and E, i.e. you are looking down on the NEATO as it drives along the curve.

5. The NEATO has its own internally defined coordinate system which is fixed to the robot (this is called a 'body-fixed' coordinate system). This coordinate system is used, among other things, for the readouts from the LIDAR scanner. The NEATO's coordinate system has the x-axis pointing forward, the y-axis pointing left, and the z-axis pointing up. For each of your NEATO pictures on the curve, indicate the orientation of the NEATO's own coordinate system. How does this relate to the $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ vectors? $\hat{\mathbf{T}}$ is co-incident with x-axis of the NEATO. $\hat{\mathbf{N}}$ is parallel to y-axis of the NEATO.



6. In order for the NEATO to follow the curve, what must be true about the orientation of the NEATO's x-axis as it traverses the curve? What about its y-axis? The NEATO x-axis must be co-incident with $\hat{\mathbf{T}}$. The NEATO y-axis has to be parallel or anti-parallel to $\hat{\mathbf{N}}$.

7. The NEATO is an example of a 'rigid body': an object which has a size (unlike a point particle, which you often consider in introductory physics) but for which the different parts of the body do not move with respect to one another (no stretching, bending, etc). When we study the motion of a rigid body, we can think about decomposing the motion into two components: the motion OF the center of the object and the motion ABOUT the center of the object. In other words, when we talk about the motion of the NEATO, in order to give a complete description, we need to specify the velocity of the center of the robot in the lab x direction, the velocity of the center of the robot in the lab y direction, and the rotational motion of the NEATO around its center relative to the x axis (θ=0 when the NEATO is going forward). This system has three degrees of freedom: x,y, and θ and we have to give the velocity for each!

   (a) On your picture, indicate the orientation θ of the NEATO at each of the indicated points. Define +θ as the angle of the forward direction of the NEATO (NEATO's x axis) measured counter-clockwise from the lab x direction. See image above. θ is shown for the point C, measured relative to the x-axis of the lab.

   (b) The angular velocity $\boldsymbol{\omega}$ is a vector quantity,

$$|\boldsymbol{\omega}| = \frac{d\theta}{dt}$$

   and its direction is the axis of rotation. For an object constrained to move in the xy plane, we can express the angular velocity as

$$\boldsymbol{\omega} = \omega\hat{\mathbf{k}}$$

where $\omega = d\theta/dt$ is the rate of change of angle of orientation. What direction is the vector, $\boldsymbol{\omega}$, if the NEATO is curving to the right? What direction is the vector, $\boldsymbol{\omega}$, if the NEATO is curving to the left? (Recall: the angular velocity follows the right hand rule.) What characteristic vector of your parametric curve is also along this direction? When curving to the left, $\frac{d\theta}{dt} > 0$ and $\boldsymbol{\omega}$ points in +$\hat{\mathbf{k}}$. When curving to the right, $\frac{d\theta}{dt} < 0$ and $\boldsymbol{\omega}$ points in -$\hat{\mathbf{k}}$. Our $\hat{\mathbf{B}}$ is along this same direction.

(c) If the object is constrained to always be oriented along its path (a NEATO that isn't slipping for example), then a consistent mathematical definition of angular velocity $\boldsymbol{\omega}$ is

$$\boldsymbol{\omega} = \hat{\mathbf{T}} \times \frac{d\hat{\mathbf{T}}}{dt}$$

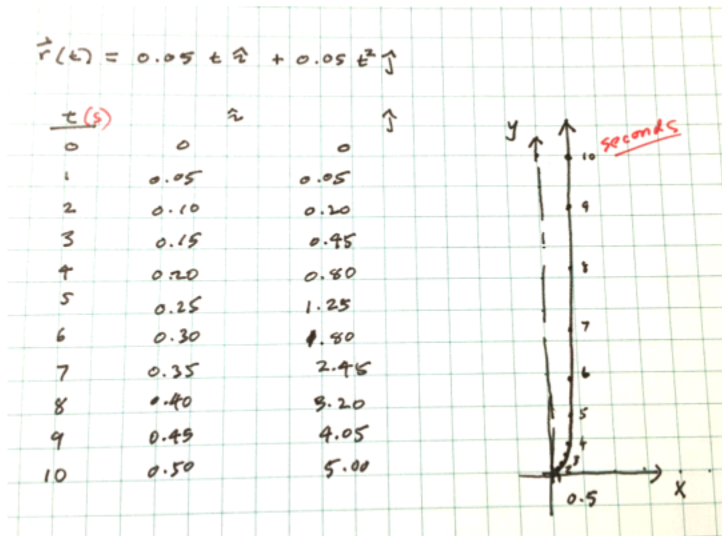
Note that $\boldsymbol{\omega}$ is a vector obeys the right-hand rule. With this definition in mind, which general direction does $\boldsymbol{\omega}$ point for the NEATO at points A,B,C,D and E? (This definition will be explored more in the overnight). For all points, $\hat{\mathbf{T}}$ is co-incident with the $+x$-axis of the NEATO. To figure out the direction of $\boldsymbol{\omega}$, we need to consider $\frac{d\hat{\mathbf{T}}}{dt}$ for each point. Conceptually, the vector $\frac{d\hat{\mathbf{T}}}{dt}$ represents the change in the direction of $\hat{\mathbf{T}}$ with time (the magnitude of $\hat{\mathbf{T}}$ is a constant of 1).We could attempt to assess $\frac{d\hat{\mathbf{T}}}{dt}$ at each of the points, but we can also remember that $\frac{d\hat{\mathbf{T}}}{dt}$ will be in the same direction as $\hat{\mathbf{N}}$, as

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

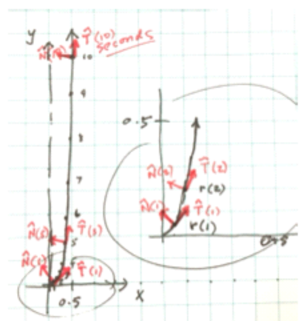
So we can use our drawings of $\hat{\mathbf{N}}$ as a proxy for $\frac{d\hat{\mathbf{T}}}{dt}$ at points A, B, C, D and E. From the image above, we will get the following result, using the right-hand rule for $\boldsymbol{\omega}$ at these points: A: $-\hat{\mathbf{k}}$; B: Hard to tell, just past the inflection point, so guessing $-\hat{\mathbf{k}}$; C: $-\hat{\mathbf{k}}$; D: $+\hat{\mathbf{k}}$; E: $-\hat{\mathbf{k}}$. Notice these results are consistent with the results above–turning left gives $+\hat{\mathbf{k}}$, turning right gives $-\hat{\mathbf{k}}$.

Solution 16.2

1. On the board, sketch the path that the NEATO moves along in 10 seconds, roughly indicating its location every second. For one second increments, a sketch might look like this:



2. Approximate and sketch $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ at 1, 2, 5 and 10 seconds on your curve.



3. Compute by hand a function that represents $\hat{\mathbf{T}}$ as a function of time. $\hat{\mathbf{T}} = \frac{\mathbf{r}'(t)}{|\mathbf{r}'(t)|}$, so

$$\mathbf{r}'(t) = \frac{d}{dt}(0.05t\hat{\mathbf{i}} + 0.05t^2\hat{\mathbf{j}})$$

$$= 0.05\hat{\mathbf{i}} + 0.10t\hat{\mathbf{j}}$$

and

$$|\mathbf{r}'(t)| = \sqrt{0.05^2 + (0.10t)^2}$$
$$= 0.05\sqrt{1 + 4t^2}$$

so

$$\hat{\mathbf{T}} = \frac{1}{\sqrt{1 + 4t^2}}(\hat{\mathbf{i}} + 2t\hat{\mathbf{j}})$$

4. The direction of $\hat{\mathbf{N}}$ is defined by $\frac{d\hat{\mathbf{T}}}{dt}$. Compute $\frac{d\hat{\mathbf{T}}}{dt}$ (you can leave it as an un-simplified equation).

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

$$\frac{d\hat{\mathbf{T}}}{dt} = \frac{d}{dt}\frac{1}{\sqrt{1+4t^2}}(\hat{\mathbf{\imath}} + 2t\hat{\mathbf{\jmath}})$$

We would need to use the chain rule:

$$= \frac{1}{\sqrt{1+4t^2}}\frac{d}{dt}(\hat{\mathbf{\imath}} + 2t\hat{\mathbf{\jmath}}) + (\hat{\mathbf{\imath}} + 2t\hat{\mathbf{\jmath}})\frac{d}{dt}\frac{1}{\sqrt{1+4t^2}}$$

$$= \frac{1}{\sqrt{1+4t^2}}(2\hat{\mathbf{\jmath}}) + (\hat{\mathbf{\imath}} + 2t\hat{\mathbf{\jmath}})(\frac{-1}{2})(1+4t^2)^{-\frac{3}{2}}(8t)$$

To evaluate $|\hat{\mathbf{T}}'|$, we would need to compute the magnitude of the vector that results from the messy chain-rule derivative. (MATLAB to the rescue!) Computing $\hat{\mathbf{N}}$ involves computing $\hat{\mathbf{T}}'/|\hat{\mathbf{T}}'|$.

5. Use CalcPlot3D to visualize unit vectors ($\hat{\mathbf{T}}$, $\hat{\mathbf{N}}$ and $\hat{\mathbf{B}}$) for the 10 second period. How does each change during the period? We can see in this short video that $\hat{\mathbf{T}}$ begins aligned with the x-axis and turns toward the y-axis as t increases; $\hat{\mathbf{N}}$ begins aligned with the y-axis at t=0 and turns toward -x as t increases; $\hat{\mathbf{B}}$ does not change–it remains in the $\hat{\mathbf{k}}$ direction throughout the time of travel.

6. Derive the vector functions that represent the linear velocity and acceleration as the NEATO moves along the path. From above,

$$\mathbf{r}'(t) = \frac{d}{dt}(0.05t\hat{\mathbf{\imath}} + 0.05t^2\hat{\mathbf{\jmath}})$$

$$= 0.05\hat{\mathbf{\imath}} + 0.10t\hat{\mathbf{\jmath}}$$

and

$$\mathbf{r}"(t) = \frac{d}{dt}(0.05\hat{\mathbf{\imath}} + 0.10t\hat{\mathbf{\jmath}})$$
$$= 0.10\hat{\mathbf{\jmath}}$$

7. Pull up the MATLAB symbolic starter code. Discuss how you would alter it to determine the angular velocity, $\boldsymbol{\omega}$, and visualize $\hat{\mathbf{T}}$, $\hat{\mathbf{N}}$ and $\hat{\mathbf{B}}$ as the NEATO travels the path? From the starter code, we would need to change the parametric equations, change the t range ("u" in the code), expand the plotting range, and add the equations to compute $\boldsymbol{\omega}$, using the equation,

$$\boldsymbol{\omega} = \hat{\mathbf{T}} \times \frac{d\hat{\mathbf{T}}}{dt}$$

An example of a MATLAB line of code that would compute this is

```
%Compute the angular velocity vector
omega=simplify(cross(T_hat,dT_hat))
```

Here is an example of altered MATLAB script that would work.

Chapter 17

Robo Week 2b: Introduction to the Simulated Neato

Schedule

Learning Objectives

Matlab Skills

- Direct the NEATO to drive forward
- Use NEATO position information to verify a motion model.

17.1 Differential Drive in Action [70 minutes]

In the first overnight you learned how to determine the velocity and acceleration of a particle moving along a parametric curve. Earlier this week, you connected these quantities to the motion of the Neato moving along the curve. Next, you'll extend this by considering not just the overall linear and rotational motion of the Neato as it moves along the curve, but how the motion of the Neato's wheels must be set in order to achieve this overall motion. To accomplish this you'll be working through the basic mechanics of differential drive vehicles. Specifically, you'll need to understand how the movement of each of the Neato's wheels translates into movement of the robot itself. In this section you will be solving two important, and closely related, problems related to robot motion:

a. Given a desired forward and angular velocity, determine the appropriate velocities of each of the robot's wheels.

b. Given the robot's current position, heading, and the velocities of its wheels, determine the robot's new position and heading. *Note: solving this problem can be quite useful when you have a sensor that estimates the actual wheel velocities of your robot (as the Neato does). In this way you can correct for discrepancies between the intended motion and what your robot actually did.*

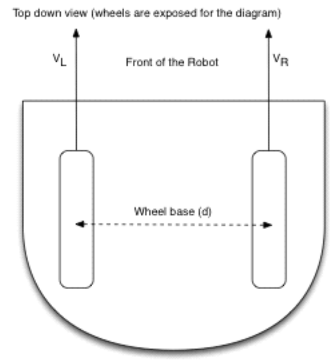


Figure 17.1: A diagram of the Neato's differential drive system.

Formalizing the Problem The Neato has two wheels equally spaced about its centerline (see Figure 17.1). As we saw during week 1, driving the wheels at different velocities (labeled V_L and V_R in the diagram) will achieve different linear and angular velocities.

Exercise 17.1

In week 1 of this module you built up an intuition for how movement of each of the wheels for a differential drive vehicle would translate into motions of the robot. Before we dive into a quantitative treatment of the subject, let's remind ourselves of some limiting cases. Determine qualitatively how the Neato would move (in terms of both linear and rotational motion) in these cases.

- What if both wheels move forward (positive velocity) with equal speed? forward
- What if both wheels move backward (negative velocity) with equal speed?
- What if one wheel drives forward and the other moves backward with equal speed?
- What if one wheel drives forward while the other remains stationary?

Now that you have refreshed your intuition, let's solve the problem quantitatively. The key insight is that the robot cannot move laterally, but instead must have a linear velocity parallel to the direction of its wheels. As the robot moves along a curve, the robot rotates about its center in order to keep itself aligned with the forward motion. Let's assume that the center of rotation of the robot is located midway between the wheels.

Exercise 17.2

Assuming no wheel slippage, the linear speed in the Neato's direction of motion V and angular velocity ω can be expressed in terms of the left and right wheel velocities V_L and V_R, and the robot's wheel base d (the distance between the two wheels).

$$V = \frac{V_L + V_R}{2} \tag{17.1}$$

$$\omega = \frac{V_R - V_L}{d} \tag{17.2}$$

In the equation above, ω is a scalar value that can be positive or negative. Do these expressions make sense? Can you confirm these expressions? Can you think of some test cases to validate these expressions? (hint: you just thought about some in the previous exercise!)
Note: we are giving you these equations rather than deriving them. If you want to explore how to derive theses equation for yourself, you can consult section 1 of this document.

Exercise 17.3

Solve Equations 17.1 and 17.2 from the the previous exercise in order to express the left and right wheel velocities in terms of the linear and angular velocities and wheel base and show that

$$V_L = V - \omega\frac{d}{2} \tag{17.3}$$

$$V_R = V + \omega\frac{d}{2} \tag{17.4}$$

Does this make sense? How can you use some of your test cases to validate these expressions?

17.1.1 Validating your Model

The Equations 17.1 and 17.2 define a motion model for your robot in the sense that they allow us to figure out what the resultant linear and angular motion of the robot would be for a given left and right wheel velocity. Similarly, we could use Equations 17.3 and 17.4 to figure out what the left and right wheel velocities of our robot should be in order to achieve a particular linear and angular motion. This computation can be extremely useful because we often have a goal of making our robot move in a particular fashion and the left and right wheel velocities are just a means to achieving that goal.

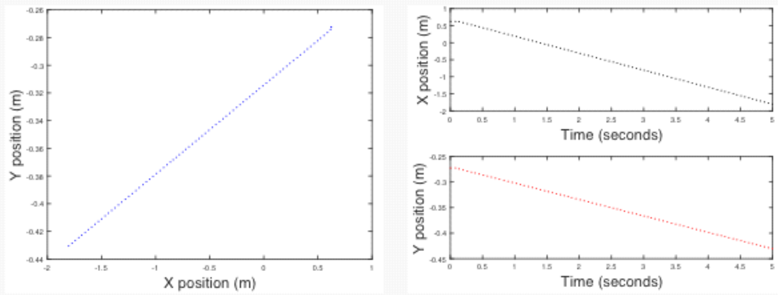
Let's further suppose that someone has measured the wheel base parameter d and obtained an estimate of $0.2m$. If we look at Equations 17.1 and 17.2 we see that now that we have an estimate of d we can plug in the wheel velocities V_L and V_R and compute the linear and angular motion of the robot This all sounds great, but of course as with any model (this is after all a motion model of the robot), you shouldn't expect it to be perfect. What we'd like to do is run an experiment to see if our model's predictions actually match what happens in real life (or simulated life in our current mode of operation).

Exercise 17.4

Design an experiment to determine how well the motion model in Equations 17.1 and 17.2 matches the robot's actual behavior. To further clarify what we mean, here are some things to keep in mind.

- Assume that you can set the left and right wheel velocities to any values you'd like between -3m/s and 3m/s (each wheel could be set to a different velocity).
- If you'd like you can vary the left and right wheel velocities over time.
- Assume that you can measure the robot's position as a function of time.

For example, you might decide that for your experiment you are going to set V_L and V_R to 0.5m/s for 5 seconds and then stop your robot. The results of this experiment might look like this.



Note that in this case the robot started in the upper right corner and moved to the lower left (as you can see from the right plots). You should notice that if you take the distance between the two end points in the left figure you get approximately 2.5m (which makes sense given you went at 0.5m/s for 5 seconds). In this case we are able to easily plot the position of the robot since we are using a simulator. For a real robot, it would be a more challenging (although doable with some additional instrumentation of your robot or its environment).

1. Design an experiment you would carry out to validate the motion model in equations 17.1 and 17.2. Describe in some detail what your experiment would entail.
2. Sketch some potential results of the experiment if the motion model proves to be relatively accurate.
3. What if it turns out that the motor on the left wheel is underpowered and moves at only 80% of the velocity you command it to. Qualitatively, how might your results look in this case?
4. Suppose you were to run your experiment and obtain some results, how might you use this experimental data to quantify the accuracy of the motion model in equations 17.1 and 17.2? Focus on high-level strategy rather than necessarily coming up with an equation.
5. Do the results of your proposed experiment tell you everything you need to know about your motion model? If not, what other experiments might you carry out and what information would you hope to gain from running them?

Exercise 17.5

In this problem we're going to run through a very basic experiment to test the validity of the motion model. In this experiment we're going to set $V_L = 0.2m/s$ and $V_R = 0.1m/s$ and let the robot go for 20 seconds. For those 20 seconds we'll be measuring the x and y position of our robot. Once 20 seconds have elapsed, we're going to stop the robot.

1. Assuming the motion model is relatively accurate, sketch some potential results for the experiment. If you get stuck, check the solution given for the previous problem.
2. Suppose you run the experiment and collect this experimental data (graph 1 is the y position versus the x position of the center of the Neato, graph 2 has x-position versus time and y-position versus time) (note: we made it a link so you won't see the results and spoil part 1).

   Based on this data is the motion model accurate? What are some potential sources of mismatch between the model and the experimental results? If you were going to revise your motion model, what might you change?

Exercise 17.6

For this exercise, we would like you to run the experiment described in the previous problem on one of your computers. (detailed instructions on how to run the experiment are given below).

- To startup the simulator, open MATLAB and navigate to `Shared QEASimulatorsV2` in your MATLAB Drive (if you have never gotten the simulator working before, go to the Meeto your Neato page).
- Start the simulator by running the following command in the MATLAB command window:

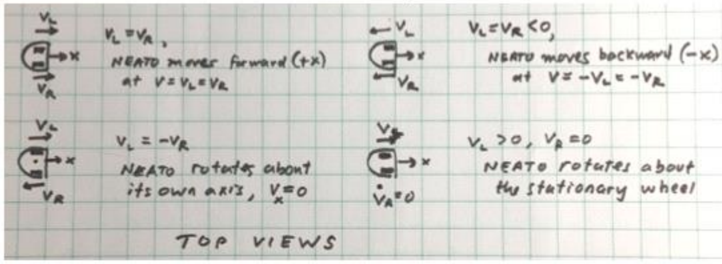
  ```
  >> qeasim start empty_no_spawn
  ```

- Run the function `runBasicWheelVelocityExperiment` to execute the experiment and generate plots. The script sends messages back and forth to the different nodes through ROS topics. Note: we are not expecting you to be able to read through the code in that script at this time. This is not something we have taught you yet and you should not feel like it is something you should know how to do (yet).

Other (fun) stuff to try.

- Try different wheel velocities by specifying inputs to the function (e.g., `runBasicWheelVelocityExperiment(1.0, 0.8)` would drive the left wheel with velocity 1.0m/s and the right wheel with velocity 0.8m/s).
- Load the ice rink world by executing step 2 above but replacing *gauntlet_no_spawn* with *ice_rink*. In this new world, see what happens if you drive the wheels too fast!

Solution 17.1



Solution 17.2

Let's take the case where $V_L = V_R > 0$. We know intuitively in this case that the NEATO will be moving forward at $V = V_L = V_R$. Using the equation for V above, we get,

$$V = \frac{V_L + V_R}{2} = \frac{2V_L}{2} = V_L$$

Let's take another case where $V_L = -V_R$ and $V_L < 0$. In this case, we know intuitively that the forward motion of the NEATO stops and it is simply spinning to the left about its center (when viewed from the top–counter clockwise). The velocity is then,

$$V = \frac{V_L + -V_R}{2} = 0$$

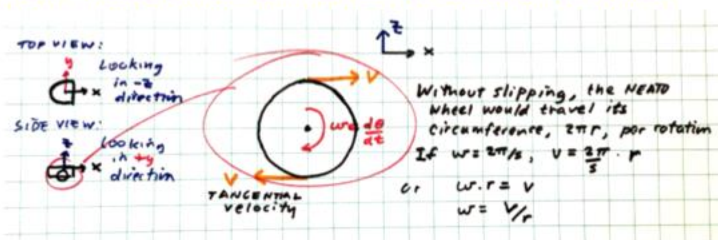
Let's use the same scenarios for computing ω. We find that for $V_L = V_R$,

$$\omega = \frac{V_R - V_L}{d} = 0$$

In other words, the NEATO is traveling in a straight line. That checks out. What about $V_L = -V_R$ and $V_L < 0$? We know intuitive that this result in a counter clockwise rotation about the NEATO center and therefore $\boldsymbol{\omega} > 0$. We get the magnitude ω from,

$$\omega = \frac{V_R - V_L}{d} = \frac{-V_L - V_L}{d} = \frac{-2V_L}{d}$$

Because $V_L < 0$, we can picture this motion, if viewing the NEATO along its y-axis, as



The wheel turns at a rate of $\omega = \frac{d\theta}{dt}$. For uniform circular motion, where ω is a constant value, $\omega = \frac{\Delta\theta}{\Delta t}$. The linear velocity would be the distance covered in the Δt, which would be $v = \frac{2\pi r}{\Delta t}$, giving,

$$v = \frac{2\pi r\omega}{\Delta\theta} = \frac{2\pi r\omega}{2\pi} = r\omega$$

Solution 17.3

Our goal will be to eliminate V_L from Equations 17.1 and 17.2. We can do this by solving each equation for V_L and then equating the two results.

$$\begin{aligned} V &= \frac{V_L + V_R}{2} && \text{Starting from Equation 17.1} \\ 2V &= V_L + V_R \\ 2V - V_R &= V_L \\ \omega &= \frac{V_R - V_L}{d} && \text{Starting from Equation 17.2} \\ d\omega &= V_R - V_L \\ V_R - d\omega &= V_L \\ V_R - d\omega &= 2V - V_R && \text{Equating our two expressions for } V_L \\ 2V_R &= 2V + d \\ V_R &= V + \frac{d}{2} \end{aligned}$$

To get an expression for V_L we can follow the same strategy, but solve each equation for V_R.

$$\begin{aligned} V &= \frac{V_L + V_R}{2} && \text{Starting from Equation 17.1} \\ 2V &= V_L + V_R \\ 2V - V_L &= V_R \\ \omega &= \frac{V_R - V_L}{d} && \text{Starting from Equation 17.2} \\ d\omega &= V_R - V_L \\ V_L + d\omega &= V_R \\ V_L + d\omega &= 2V - V_L && \text{Equating our two expressions for } V_L \\ 2V_L &= 2V - d \\ V_L &= V - \frac{d}{2} \end{aligned}$$

To sanity check our solution, we could set $\omega = 0$ and observe that this would result in both wheels being commanded to move at the same velocity. For a positive ω we have that the right wheel goes faster, which makes sense since it is on the outside of a counterclockwise turn.

Solution 17.4

1. A reasonable experiment would be to set $V_L = 0.1m/s$ and $V_R = 0.2m/s$. If we let the robot travel with these velocities for some amount of time, e.g., 30 seconds, we would expect it to trace out a circular path in the counterclockwise direction.
2. We would expect the resultant path to be a counterclockwise circle. Given $d = 0.2m$ we expect $\omega = \frac{0.2m/s - 0.1m/s}{0.2m} = 0.5rad/s$. It would take the Neato $\frac{2\pi}{0.5rad/s} = 4\pi$ seconds to make a full trip around the circle. Further, we expect the linear speed to be $0.15m/s$. Given this we would expect the circumference of the circle the Neato traces to be 0.6π and the radius to be 0.3 m.
3. In this case we would expect the Neato to travel in a tighter circle since ω will be larger.
4. We could compute the radius of the circle the Neato actually drives and compare it to the predicted radius. We could also compare the time it takes to complete a traversal of the full circle to the predicted value.

5. This experiment will tell us a fair amount. We wouldn't be able to distinguish between the case where one wheel is underpowered (e.g., the hypothetical case posed above) or the wheel base being measured incorrectly. To distinguish between these two cases, we could add an additional experiment of having the robot drive straight for some time to see if it fits predictions.

Solution 17.5

1. Using the same logic as in the solution to the previous problem, we'd expect the Neato to travel around a circle of radius 0.3m with a period of 4π seconds. The only difference is the Neato will now be moving in clockwise direction.
2. The experimental data doesn't match the prediction as well as we might look since the measured radius of the circle the Neato moves around is about 0.35m. Perhaps the wheel base is incorrect?

Solution 17.6

There's no solution to this problem. We want you to verify that the simulator is working on your computer and get comfortable interacting with the robot.

Chapter 18

Robo Homework 2: Angular Velocity, NEATOs, and Partial Derivatives

Contents

Learning Objectives

Concepts

- Predict the direction of $\boldsymbol{\omega}$ based on an object's space curve.
- Compute the partial derivatives of functions of more than one variable.

Matlab Skills

- Compute an object's motion properties from its parametric vector function.
- Use basic ROS commands to send instructions and receive sensor data from a simulated NEATO.

18.1 Angular Velocity Revisited

Suppose we have a 2D parametric curve $\mathbf{r}(t) = f(t)\hat{\mathbf{i}} + g(t)\hat{\mathbf{j}}$. We saw in the week 1 homework assignment that the linear velocity vector is given by $\mathbf{r}'(t) = f'(t)\hat{\mathbf{i}} + g'(t)\hat{\mathbf{j}}$.

Determining the expression for the angular velocity $\boldsymbol{\omega}(t)$ of our robot is more involved. Before we derive the correct expression, we will revisit the notion of angular velocity vectors. In week 1 of this module you encountered this idea when you experimented viewing the components of the angular velocity vector in a mobile app. Here, we'll expand on it further. While expressing angular velocity as a vector may seem overly complex. Since the robot moves on a 2D plane, we know that the robot will rotate about the z-axis, and it seems like we should just be able to compute the scalar magnitude and whether to turn clockwise or counterclockwise. However, thinking about the angular velocity as a vector will enable our derivation to be done in a much more straightforward and generalizable manner.

Angular velocity vectors point in the direction of the axis about which the body rotates (for our robot, either the positive or negative z-axis). For right-handed coordinate systems (such as the one we are using here), a positive rotation happens **counterclockwise** about the direction of the rotation axis. The magnitude of the angular velocity vector indicates the speed of rotation.

Earlier this week, we discussed the coordinate system attached to the robot: the body fixed frame. Because the heading of the robot is locked to the tangent vector $\hat{\mathbf{T}}$ of the curve, we can think of the vector $\hat{\mathbf{T}}$ as being a constant in the body fixed frame of the robot. The body fixed frame is rotating with some unknown angular velocity vector $\boldsymbol{\omega}$ with respect to the room coordinate system. If we wish to know the time derivative of the tangent vector $\hat{\mathbf{T}}$ in the room coordinate system, we can use the generalized relationship between the time derivatives of vectors in two coordinate systems which are rotating with an angular velocity vector $\boldsymbol{\omega}$ with respect to each other. This expression is

$$\frac{d\hat{\mathbf{T}}}{dt}|_{room} = \frac{d\hat{\mathbf{T}}}{dt}|_{body} + \boldsymbol{\omega} \times \hat{\mathbf{T}} \tag{18.1}$$

(Full mathematical derivation here; nice heuristic explanation here.) In the body frame of the robot, $\hat{\mathbf{T}}$ is unchanging, since it is always aligned with the forward direction, so the term $\frac{d\hat{\mathbf{T}}}{dt}|_{body}$ is zero leaving us with

$$\frac{d\hat{\mathbf{T}}}{dt}|_{room} = \boldsymbol{\omega} \times \hat{\mathbf{T}} \tag{18.2}$$

Next we can make use of the scalar triple product, which states that $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{c}(\mathbf{a} \cdot \mathbf{b})$. Using this we can derive our angular velocity vector as follows

$$\begin{aligned}
\frac{d\hat{\mathbf{T}}}{dt}|_{room} &= \boldsymbol{\omega} \times \hat{\mathbf{T}} \\
\hat{\mathbf{T}} \times \frac{d\hat{\mathbf{T}}}{dt}|_{room} &= \hat{\mathbf{T}} \times \boldsymbol{\omega} \times \hat{\mathbf{T}} \\
&= \boldsymbol{\omega}(\hat{\mathbf{T}} \cdot \hat{\mathbf{T}}) - \hat{\mathbf{T}}(\hat{\mathbf{T}} \cdot \boldsymbol{\omega}) \\
&= \boldsymbol{\omega}(1) - \hat{\mathbf{T}}(0) \\
&= \boldsymbol{\omega} \\
\Rightarrow \boldsymbol{\omega} &= \hat{\mathbf{T}} \times \frac{d\hat{\mathbf{T}}}{dt}|_{room}
\end{aligned} \tag{18.3}$$

The x and y components of the angular velocity vector will always be zero because $\hat{\mathbf{T}}$ and $\frac{d\hat{\mathbf{T}}}{dt}|_{room}$ are in the x-y plane and orthogonal. The magnitude of the z-component is the angular speed. If the z-component is positive, we turn counterclockwise at that speed. When it is negative, we turn clockwise at that speed.

Exercise 18.1

In the Week 1 homework assignment you found the unit tangent and normal vectors for various parameterized curves. We will use that information to find linear and angular velocities, then translate those to left and right wheel velocities for the NEATO.
The vector for a circle centered at the origin in the x-y plane is given by:

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

We will make the following assumptions about our parameters:

- t starts at 0 seconds and increases with time ($\mathbf{r}(t)$ gives us the Neato's position at time t).
- $R > 0$ and provides the radius of the circle.
- α can be positive or negative (we'll ask you to interpret its meaning later in the problem).

If you decide to work in MATLAB (either to check your solutions or to do the problems), we'd advise you to build off the your code from last week's assignment (or use the starter code provided in that assignment). If you're finding your answers are not in a form that you expect, considering the following hints.

- Specify assumptions (e.g., t, α, R are all real and $R > 0, t \geq 0$). Note as stated above: α does not have to be positive.
- Use the MATlAB function `simplify` on your answer.

1. How does the sign of α affect the path that the Neato takes around the circle? Alpha controls speed and direction
2. What are the linear velocity vector and linear speed? v = $(-R\alpha \sin(\alpha t) \quad R\alpha \cos(\alpha t) \quad 0)$ speed = $R\,|\alpha|$
3. What is the unit tangent vector for the circle? Make sure your answer makes sense for both positive and negative values of α. T_hat = $\left(-\frac{\alpha \sin(\alpha t)}{|\alpha|} \quad \frac{\alpha \cos(\alpha t)}{|\alpha|} \quad 0\right)$
4. What is the unit normal vector? N_hat = $(-\cos(\alpha t) \quad -\sin(\alpha t) \quad 0)$
5. What is the angular velocity vector? omega = $(0 \quad 0 \quad \alpha)$
6. For the uniform circular motion we have been investigating so far, what does the parameter we have labeled α represent? How is it related to the time it takes to complete one traverse of the circular trajectory? Rotational Velocity. T_{complete rotation} = 2pi/alpha
7. How would you modify the initial equation for the position vector when its trace is circle of radius 1 m? R = 1m
8. What value would you choose for α if you want your robot to complete a counterclockwise path around the circle in 30 seconds? Alpha = 2pi / 30s
9. What are the equations for the left and right wheel velocities for the uniform circle? (d=wheel displacement). You can leave your answer in terms of d or substitute a reasonable value for the Neato of $d = 0.235m$ v_l = $R\,|\alpha| - \frac{d\,|\alpha|}{2}$ v_r = $R\,|\alpha| + \frac{d\,|\alpha|}{2}$
10. What are the left and right wheel velocities needed for a 1 m radius counterclockwise circle to be completed in 30 seconds?

```
v_l_actual = double(subs(v_l)) % m/s
```

v_l_actual = 0.1848

```
v_r_actual = double(subs(v_r)) % m/s
```

v_r_actual = 0.2340

Exercise 18.2

The vector for a counterclockwise path around an ellipse is given by:

$$\mathbf{r}(t) = a \cos \alpha t\ \hat{\mathbf{i}} + b \sin \alpha t\ \hat{\mathbf{j}},\ \ \alpha t \in [0, 2\pi]$$

In this problem we can assume that α, a, b are all positive and that $t \geq 0$.

1. What is the tangent vector for the ellipse?
2. What is the unit tangent vector for the ellipse?
3. What is the linear velocity vector? How does it differ from the example of the circle?
4. What is the unit normal vector?
5. What is the angular velocity vector? How does it differ from the circle?
6. What are the left and right wheel velocities?

tangent_vector = $\begin{pmatrix}-a\,w\sin(t\,w) & b\,w\cos(t\,w) & 0\end{pmatrix}$

unit_tangent_vector =

$$\begin{pmatrix}-\frac{a\sin(t\,w)}{\sqrt{a^2\sin(t\,w)^2-b^2\sin(t\,w)^2+b^2}} & \frac{b\cos(t\,w)}{\sqrt{a^2\sin(t\,w)^2-b^2\sin(t\,w)^2+b^2}} & 0\end{pmatrix}$$

linear_velocity_vector = $\begin{pmatrix}-a\,w\sin(t\,w) & b\,w\cos(t\,w) & 0\end{pmatrix}$

As opposed to the circle, the scalar coefficients on are different for each dimension.

unit_normal_vector =

$$\begin{pmatrix}-\frac{b\cos(t\,w)}{\sqrt{-a^2\cos(t\,w)^2+a^2+b^2\cos(t\,w)^2}} & -\frac{a\sin(t\,w)}{\sqrt{-a^2\cos(t\,w)^2+a^2+b^2\cos(t\,w)^2}} & 0\end{pmatrix}$$

angular_velocity_vector =

$$\begin{pmatrix}0 & 0 & \frac{a\,b\,w}{a^2\sin(t\,w)^2-b^2\sin(t\,w)^2+b^2}\end{pmatrix}$$

As compared to the circle, the angular velocity vector is *way* more complicated to
reflect the fact that elptical motion involves a constantly changing rotational speed

left_wheel_velocity =

$$\sqrt{a^2\,w^2\,|\sin(t\,w)|^2+b^2\,w^2\,|\cos(t\,w)|^2}-\frac{a\,b\,d\,w}{2\,|a^2\sin(t\,w)^2-b^2\sin(t\,w)^2+b^2|}$$

right_wheel_velocity =

$$\sqrt{a^2\,w^2\,|\sin(t\,w)|^2+b^2\,w^2\,|\cos(t\,w)|^2}+\frac{a\,b\,d\,w}{2\,|a^2\sin(t\,w)^2-b^2\sin(t\,w)^2+b^2|}$$

7. Plot the linear velocity vector as a function of time for various combinations of the parameters a, b, and α.
8. Plot the angular velocity vector as a function of time for various combinations of the parameters a, b, and α.
9. Plot the left and right wheel velocities as a function of time for various combinations of the parameters a, b, and α.

Done

18.2 Fun with (Simulated) NEATOs

In the previous section we found the left and right wheel velocities needed to drive a particular trajectory. In this section of the assignment, we will be thinking about how to translate the velocity vectors to a Matlab program that will control your NEATO.

The control of your NEATO is built on top of the Robotic Operating System (ROS), so you will be using ROS commands to control the velocity values for your robot. Prior to starting this section, you will need to have completed the setup steps in Meeto Your Neato

Here, we will start by playing with some very basic commands. Consider the program drivefoward.m.

This code snippet defines the function "driveforward" which will cause your NEATO to.... you guessed it, drive forward. You will notice that this function does not have a meaningful output, its sole purpose is to move your robot forward.

18.2.1 The structure of a Simple Robot Program

While the program has lots of comments, we elaborate on specific lines of the code to help you understand its structure on this page. Please read through the notes.

Exercise 18.3

`driveforward` is a Matlab function that can be called from the command window using the form:

`driveforward(distance, speed)`

where `distance` and `speed` are the numerical values that you give to the function `driveforward`. The function is built into the simulator, so you don't have to download it, but

- To startup the simulator so you can test the `driveforward command`, open MATLAB and navigate to `Shared`
  `QEASimulatorsV2` in your MATLAB Drive.
- Start the simulator by running the following command in the MATLAB command window:

  `>> qeasim start empty no spawn`

QEASimulatorsV2 in your MATLAB Drive.

- Start the simulator by running the following command in the MATLAB command window:

```
>> qeasim start empty_no_spawn
```

Using the `driveforward(distance,speed)` function, try driving the NEATO for several combinations of distances and speeds (Remember to specify values for `distance` (in meters) and `speed` (in m/s). Do the final distance and time match your expectations? Note that squares of the grid on the simulator visualization are 1 meter by 1 meter.

18.2.2 Receiving Sensor Data

In the previous example program we published to a ROS topic to set the NEATO wheel velocities. In ROS you can also subscribe to a topic to do things like receive sensor data. Download and open the program driveUntilBump in Matlab.

Exercise 18.4

To be successful in this exercise, you should have the simulator up and running, e.g., using the procedure described in the previous exercise. Instead of the 'empty_no_spawn' simulator environment, you should use the command:

```
qeasim start gauntlet_final
```

You can think of ROS topics as some aspect in your robot environment, such as the robot's odometer readings. Any node (i.e., software program) in the ROS environment can *publish* information to the topics or *subscribe* to a topic. The Master contains a list of the available topics and node *publishers* and *subscribers*.

1. In line 2 of driveUntilBump, the 'rossubscriber' command is introduced; it requires that you pass it the 'topic' that you want to subscribe to. For a list of active topics, in the MATLAB command window, type,

   ```
   rostopic list
   ```

   From the code, what sensor output topic are we monitoring? *Note*: You can see what the name of it is, but it's pretty hard to figure this out with the simulator since there is no way to actually physically touch the robot, just look at the solution for this one!

2. The variable `bumpMessage` is a structure. You might recall that a structure is a type of object that has properties. What topic provides the data for `bumpMessage`?

   The data in the property can be accessed by referring to its name, e.g., `bumpMessage.Data`. If you are interested in seeing how to drill down on the information in a rostopic, here is an example of the MATLAB commands, accessingROS.pdf

   What is the size of 'bumpMessage.Data'? What do the values contained in that variable mean? Note: it's pretty hard to figure this out with the simulator since there is no way to actually physically touch the robot, just look at the solution for this one!

3. What is the 'driveUntilBump' code commanding the robot to do? Drive till it hits the wall

4. Test the 'driveUntilBump' code on a NEATO and verify that your interpretation is correct.

5. Modify the 'driveUntilBump' code to make it a function where the robot velocity is an input.

6. Using what you have learned from the examples above, write a program that meets the following requirements:

   - The program commands the robot to drive a designated distance at a chosen speed, and stops when that distance is reached.
   - If the bump sensor is triggered, the robot reverses direction and backs up for 5 seconds then stops.

   Try developing this code on your own first, then if you get stuck, take a look at the program driveUntilBumpThenRunAway and this driveforward for inspiration. You may also find this video on rostic and rostoc to be helpful.

18.3 Partial Derivatives

The following notes about Partial Derivatives might be helpful. Mark has also put together videos on this topic. You can check your calculations using WolframAlpha or the Symbolic Toolbox in Matlab.

Consider a function of two variables $f(x, y)$. If we identify $z = f(x, y)$, then we can visualize this function as a surface in 3D. At any point on the surface, (a, b, c), we can ask about the slope of the tangent line in the x-direction and in the y-direction. In the first case, we intersect the surface with the plane $y = b$ and consider the rate of change of f in the x-direction only. In the second case, we use the plane $x = a$ and consider the rate of change of f in the y-direction only. There are therefore two fundamental derivatives,

$\frac{\partial f}{\partial x}$ is the partial derivative of f with respect to x
$\frac{\partial f}{\partial y}$ is the partial derivative of f with respect to y

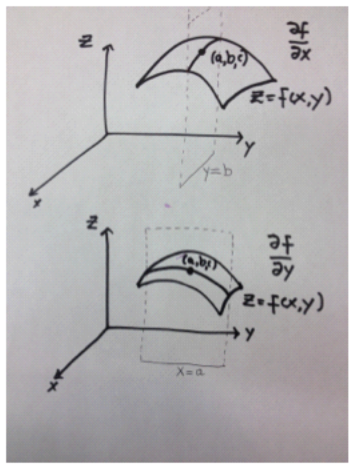


Figure 18.1: Partial derivatives of a function of two variables.

In each case we compute the derivative with respect to one variable by holding the other one fixed, i.e. treating it as a constant.

Exercise 18.5

Evaluate $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ for each of the following functions.

1. $f(x, y) = x^2 \sin(xy^2)$ $\frac{\partial f}{\partial x} = 2x\sin(xy^2) + y^2x^2\cos(xy^2)$ $\frac{\partial f}{\partial y} = 2x^3y\cos(xy^2)$
2. $f(x, y) = 4 + x^3 + y^3 - 3xy$ $\frac{\partial f}{\partial x} = 3x^2 - 3y$ $\frac{\partial f}{\partial y} = 3y - 3x$

We can also evaluate higher-order derivatives, but now there are several possibilities. We could take two derivatives with respect to x,

$$\frac{\partial}{\partial x}\frac{\partial f}{\partial x} = \frac{\partial^2 f}{\partial x^2}.$$

We could take two derivatives with respect to y,

$$\frac{\partial}{\partial y}\frac{\partial f}{\partial y} = \frac{\partial^2 f}{\partial y^2}.$$

Or we could take a derivative with respect to x and then with respect to y, and vice versa,

$$\frac{\partial}{\partial y}\frac{\partial f}{\partial x} = \frac{\partial^2 f}{\partial y \partial x}$$

$$\frac{\partial}{\partial x}\frac{\partial f}{\partial y} = \frac{\partial^2 f}{\partial x \partial y}.$$

In all of the functions that we will be dealing with, the mixed partials are equal

$$\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial^2 f}{\partial x \partial y}.$$

Exercise 18.6

Evaluate all four second-order derivatives of the following functions. What do you notice about the mixed partial derivatives?

1. $f(x, y) = x^2 \sin(xy^2)$
2. $f(x, y) = 4 + x^3 + y^3 - 3xy$ $\frac{\partial f}{\partial x} = 3x^2+3y$ $\frac{\partial f}{\partial y} = 3y^2-3x$ $\frac{\partial f}{\partial x^2} = 6x$ $\frac{d^2 f}{\partial y^2} = 6y$

$\frac{\partial^2 f}{\partial x \partial y} = -3$

$\frac{d^2 f}{dy dx} = 3$

ans(x, y) = $2\sin(xy^2) - x^2y^4\sin(xy^2) + 4xy^2\cos(xy^2)$
ans(x, y) = $6x^2y\cos(xy^2) - 2x^3y^3\sin(xy^2)$
ans(x, y) = $6x^2y\cos(xy^2) - 2x^3y^3\sin(xy^2)$
ans(x, y) = $2x^3\cos(xy^2) - 4x^4y^2\sin(xy^2)$

Under very gentle conditions it is generally true that the mixed partial-derivatives are always equal, i.e.

$$\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial^2 f}{\partial x \partial y}$$

Exercise 18.7

Using the power of the internet, under what conditions are the mixed partial derivatives of f equal?

All partial second derivs. exist & are continuous: https://math.stackexchange.com/a/1075717

18.3.1 The Gradient and the Hessian

We've seen that a function of two variables, $f(x, y)$, has two partial derivatives, $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$. Rather than thinking of these separately, we can package them into a vector known as the gradient vector. In terms of notation we write

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Exercise 18.8

$f(x,y) = 3x^2y + y^3 - 3x^2 - 3y^2 + 2$

$\frac{\partial f}{\partial x} = 6xy - 6x$ $\frac{\partial f}{\partial y} = 3x^2 + 3y^2 - 6y$ $\nabla f = \begin{bmatrix} 6xy - 6x \\ 3x^2+3y^2-6y \end{bmatrix}$

Find the gradient vector of the function $f(x, y) = 3x^2y + y^3 - 3x^2 - 3y^2 + 2$ by hand and check your answer using WolframAlpha or the Symbolic Toolbox in Matlab. Evaluate it at the point $(1, 2)$.

$\nabla f = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$

In the same vein, we can package the four second-order partial derivatives into a matrix called the Hessian matrix. In terms of notation we write

$$Hf = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

Exercise 18.9

Find the Hessian matrix of the function $f(x, y) = 3x^2y + y^3 - 3x^2 - 3y^2 + 2$ by hand and check your answer using WolframAlpha or the Symbolic Toolbox in MATLAB. Evaluate it at the point $(1, 2)$.

$f(x,y) = 3x^2y + y^3 - 3x^2 - 3y^2 + 2$

$\frac{\partial f}{\partial x} = 6xy - 6x \quad \frac{\partial f}{\partial y} = 3x^2 + 3y^2 - 6y$

$\frac{\partial f^2}{\partial x^2} = 6y - 6 \quad \frac{\partial f}{\partial y^2} = +6y - 6$

$\frac{df^2}{\partial y \partial x} = 6x \quad \frac{df}{dx \delta y} = 6x$

$Hf = \begin{bmatrix} 6y-6 & 6x \\ 6x & +6y-6 \end{bmatrix}$

$= \begin{bmatrix} 6 & 6 \\ 6 & 6 \end{bmatrix}$

Solution 18.1

1. A positive value of α corresponds to a counterclockwise path around the circle, whereas a negative value of α corresponds to a clockwise path.

2. The linear velocity vector is given by $\mathbf{v}(t) = \mathbf{r}'(t) = x'(t)\hat{\mathbf{i}}+y'(t)\hat{\mathbf{j}}+z'(t)\hat{\mathbf{k}}$ or $\mathbf{v}(t) = v(t)\hat{\mathbf{T}}(t)$. For the case of the circle, we know:

$$\mathbf{r}'(t) = -R\alpha \sin \alpha t \ \hat{\mathbf{i}} + R\alpha \cos \alpha t \ \hat{\mathbf{j}}$$

   So, the linear velocity vector is $\mathbf{v}(t) = \alpha R(-\sin \alpha t \ \ \hat{\mathbf{i}} + \cos \alpha t \ \ \hat{\mathbf{j}})$ and the magnitude (linear speed) is $v(t) = |\alpha| R$ in units of $\frac{\text{m}}{\text{s}}$.

3. The unit tangent vector is:

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

   Note that the quantity $\frac{\alpha}{|\alpha|}$ gives us the sign of α. This means that if we negate α that will cause $\hat{\mathbf{T}}$ to negate as well (corresponding to, as we would expect, moving about the circle in the opposite direction).

4. The unit normal vector is

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

5. The angular velocity is constant: $\boldsymbol{\omega} = \alpha\hat{\mathbf{k}}$.

6. The parameter α is the angular frequency of motion, often denoted by the scalar ω (we'll use α here since we are already using $\boldsymbol{\omega}$ for the angular velocity vector). For a uniform circular motion, the frequency and angular velocity are equal. The time to complete one traverse of the circle is given by the period $T = \frac{2\pi}{|\alpha|}$.

7. Set R=1 m

8. For a positive value of α (corresponding to a counterclockwise path) we know that the product αt must go from 0 to 2π for the robot to complete one cycle of the parameterized curve. So, after one complete trip around the circle, $\alpha T = 2\pi$, so $\alpha = \frac{2\pi}{T} = 0.21$ with units $\frac{1}{s}$.

9. We can use the equations from Day 2 of the module that relate left and right wheel velocities to linear speed and angular velocity. Plugging the expressions for linear speed and angular velocity that we found earlier in this exercise into those equations we arrive at

$$V_L = R|\alpha| - \frac{d\alpha}{2}$$

   and

$$V_R = R|\alpha| + \frac{d\alpha}{2}$$

   Note: that for the simulated Neato (and the real Neato), the wheel base is approximately $d = 0.235m$.

10.

$$\begin{aligned} V_L &= R\alpha - \frac{d\alpha}{2} \\ &= (1m)\left(0.21\frac{1}{s}\right) - \frac{(0.235m)(0.21\frac{1}{s})}{2} \\ &= 0.185\frac{m}{s} \\ V_R &= R\alpha + \frac{d\alpha}{2} \\ &= (1m)\left(0.21\frac{1}{s}\right) + \frac{(0.235m)(0.21\frac{1}{s})}{2} \\ &= 0.235\frac{m}{s} \end{aligned} \tag{18.4}$$

Solution 18.2

1. What is the unit tangent vector for the ellipse? An altered version of the the symbolic MATLAB starter code is here: link. This code computes almost all the needed properties ($\hat{\mathbf{T}}$, etc.), but I have to change the parameters to fit the ellipse. In my code, I used 'w' for α. Here is a MATLAB solution. When you run the MATLAB code, you will see that if you substituted α for 'w' in the code and used the trigonometric identity, $cos^2(\theta) = 1 - sin^2(\theta)$, the unit tangent vector reduces to,

$$\hat{\mathbf{T}} = \frac{-asin(\alpha t)}{\sqrt{a^2 sin^2(\alpha t) + b^2 cos^2(\alpha t)}}\hat{\mathbf{i}} + \frac{bcos(\alpha t)}{\sqrt{a^2 sin^2(\alpha t) + b^2 cos^2(\alpha t)}}\hat{\mathbf{j}}$$

2. What is the unit normal vector? See $\hat{\mathbf{N}}$ in the output of the MATLAB code. Using the trig identity above, you will get a 'simplified' $\hat{\mathbf{N}}$,

$$\hat{\mathbf{N}} = \frac{-bcos(\alpha t)}{\sqrt{a^2 sin^2(\alpha t) + b^2 cos^2(\alpha t)}}\hat{\mathbf{i}} + \frac{-asin(\alpha t)}{\sqrt{a^2 sin^2(\alpha t) + b^2 cos^2(\alpha t)}}\hat{\mathbf{j}}$$

3. What is the angular velocity vector? How does it differ from the circle? See ω in the output below. Because the NEATO motion is confined to the xy-plane, the angular velocity vector is coincident with the $+\hat{\mathbf{k}}$. However, unlike the circle, which has a constant $\frac{d\theta}{dt}$, the $|\boldsymbol{\omega}| = f(t)$:

$$\boldsymbol{\omega} = \frac{ab\alpha}{a^2 sin^2(\alpha t) + b^2 cos^2(\alpha t)}\hat{\mathbf{k}}$$

For both the circle and the ellipse, $\boldsymbol{\omega} > 0$, since the motion in counter clockwise. *Note:*I've used the trigonometric identity, $cos^2(\theta) = 1 - sin^2(\theta)$, so your MATLAB output might look slightly different.

4. What are the left and right wheel velocities? Using the equations, from Day 2 we have,

$$V_L = V - \omega\frac{d}{2} \tag{18.5}$$

$$V_R = V + \omega\frac{d}{2} \tag{18.6}$$

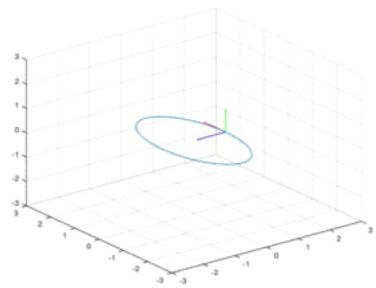
In order to map what we've been doing in this document to these equations, we can keep in mind the following.

- ω (notice that ω is not bolded and refers to a scalar) represents the component of the angular velocity in the $\hat{\mathbf{k}}$ direction
- V is the linear speed, which is given as $|r'(t)|$.

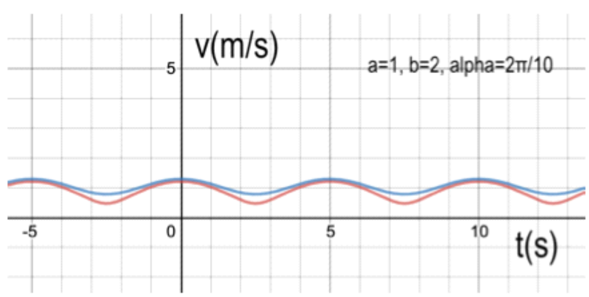
$$V_L = V - \omega\frac{d}{2} \tag{18.7}$$
$$= \alpha\sqrt{a^2sin(\alpha t)^2 + b^2cos(\alpha t)^2} - \frac{ab\alpha}{a^2sin^2(\alpha t) + b^2cos^2(\alpha t)}\frac{d}{2} \tag{18.8}$$
$$V_R = V + \omega\frac{d}{2} \tag{18.9}$$
$$= \alpha\sqrt{a^2sin(\alpha t)^2 + b^2cos(\alpha t)^2} + \frac{ab\alpha}{a^2sin^2(\alpha t) + b^2cos^2(\alpha t)}\frac{d}{2} \tag{18.10}$$

5. Plot the linear velocity vector as a function of time for various combinations of the parameters a, b, and α. When you do this, you will see that the trace of the linear velocity function vector follows the same path as the space curve, $r(t)$, but is shifted in phase by $\frac{\pi}{2}$. We say that $r'(t)$ leads $r(t)$ because at $\alpha t = 0$, $r'(t)$ is positioned at $\frac{\pi}{2}$ radians ahead of $r(t)$.

6. Plot the angular velocity vector as a function of time for various combinations of the parameters a, b, and α. Because $r(t)$ is constrained to counterclockwise motion in the xy-plane, $\boldsymbol{\omega} > 0$ and in the direction of $\hat{\mathbf{k}}$. Its magnitude varies with the choice of parameters. For $a = 1, b = 2, \alpha = \frac{2\pi}{10}$, a complete rotation of $r(t)$ around the ellipse takes takes 10 seconds. The trace of $r'(t)$ will follow the same ellipse, starting at $r'(0) = (0, \frac{2\pi}{2})$ and ending at $r'(10) = (0, \frac{2\pi}{2})$



7. Plot the left and right wheel velocities as a function of time for various combinations of the parameters a, b, and α. Choosing $a = 1$ and $b = 2$ and $\alpha = \frac{2\pi}{10}$, gives



Regardless of the combinations of the parameters, as long as the NEATO is moving in the counterclockwise direction, $V_L < V_R$.

Solution 18.3

The behavior of the actual should match closely with your expectations. However, notice the `Realtime` and `Simtime` in the simulator. Your `Simtime` will be less than your `Realtime` if your computer's processor is slow–*This difference between simulated and real time should be taken into account when programming the robot.* We have provided the functions `rostic` and `rostoc` that should work well for this purpose, however, if you want to access the simulator time directly you can use the `rostime` command in MATLAB. For example, in a MATLAB script, you can assign,

```
time = rostime('now')
```

This command assigns the a ROS time *object* to `time`. You can get the integer values of the simulated time in seconds (`time.Sec`) and nanoseconds (`time.Nsec`). However, it's better to directly access the simulation time in seconds as a double-precision array by using,

```
time.seconds
```

Don't forget that the robot has a maximum velocity!

Solution 18.4

1. From the code, what sensor output topic are we monitoring?

   We are monitoring the output of the Neato's bump sensor (which is located at the front of the robot and triggers when the front of the robot contacts something).

2. What is the size of 'bumpMessage.Data'? What do the values contained in that variable mean?

   The variable 'bumpMessage.Data' contains four numbers, one for each of the Neato's bump sensors. Note that in the simulated Neato the bump sensors are either all on or all off.

3. What is the 'driveUntilBump' code commanding the robot to do?

   The code tells the robot to drive forward with velocity of 0.1 m/s until it runs into something and then stops.

4. Test the 'driveUntilBump' code on a NEATO and verify that your interpretation is correct.

   You should run the code in the Robot simulator and observe the behavior. In the 'gauntlet_final' environment, you will notice several obstacles and concrete barriers. Based on the initial orientation of the Neato, you should observe the robot drive forward until it collides with a concrete barrier, then stop. You will also notice the 'bumpMessage' structure in your Matlab Workspace after the collision.

5. Modify the 'driveUntilBump' code to make it a function where the robot velocity is an input.

   An example solution code is here: driveUntilBumpWithVelInput.

6. Using what you have learned from the examples above, write a program that meets the following requirements:

   - The program commands the robot to drive a designated distance at a chosen speed, and stops when that distance is reached.
   - If the bump sensor is triggered, the robot reverses direction and backs up for 5 seconds then stops.

   Try developing this code on your own first, then if you get stuck, take a look at the program driveUntilBumpThenRunAway and this driveforward code for inspiration. You may also find this video on rostic and rostoc to be helpful.

Solution 18.5

1.

$$\begin{aligned}\frac{\partial f}{\partial x} &= 2x\sin(xy^2) + x^2y^2\cos(xy^2)\\ \frac{\partial f}{\partial y} &= 2x^3y\cos(xy^2)\end{aligned}$$

2.

$$\begin{aligned}\frac{\partial f}{\partial x} &= 3x^2 - 3y\\ \frac{\partial f}{\partial y} &= 3y^2 - 3x\end{aligned}$$

Solution 18.6

1.

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= (2 - x^2y^4)\sin(xy^2) + 4xy^2cos(xy^2)\\ \frac{\partial^2 f}{\partial y^2} &= 2x^3\cos(xy^2) - 4x^4y^2\sin(xy^2)\\ \frac{\partial^2 f}{\partial y\partial x} &= 6x^2y\cos(xy^2) - 2x^3y^3\sin(xy^2)\\ \frac{\partial^2 f}{\partial x\partial y} &= 6x^2y\cos(xy^2) - 2x^3y^3\sin(xy^2)\end{aligned}$$

As we would expect, the mixed partial derivatives are equal.

2.

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= 6x\\ \frac{\partial^2 f}{\partial y^2} &= 6y\\ \frac{\partial^2 f}{\partial y\partial x} &= -3\\ \frac{\partial^2 f}{\partial x\partial y} &= -3\end{aligned}$$

As we would expect, the mixed partial derivatives are equal.

Solution 18.7

If the derivatives exist and are continuous then they are equal. See Paul's Online Math Notes about this.

Solution 18.8

Let's first evaluate the first-derivatives:

$$\begin{aligned}\frac{\partial f}{\partial x} &= 6xy - 6x\\ \frac{\partial f}{\partial y} &= 3x^2 + 3y^2 - 6y\end{aligned}$$

and then we simply package them into a vector

$$\nabla f = \begin{bmatrix} 6xy - 6x \\ 3x^2 + 3y^2 - 6y \end{bmatrix}$$

If we evaluate the gradient vector at $(1, 2)$ we see that

$$\nabla f(1, 2) = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$$

Solution 18.9

Let's first evaluate the second-derivatives:

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &= 6y - 6 \\ \frac{\partial^2 f}{\partial y^2} &= 6y - 6 \\ \frac{\partial^2 f}{\partial y \partial x} &= 6x \\ \frac{\partial^2 f}{\partial x \partial y} &= 6x \end{aligned}$$

and then we simply package them into a matrix

$$Hf = \begin{bmatrix} 6y - 6 & 6x \\ 6x & 6y - 6 \end{bmatrix}.$$

Note that we expect the Hessian to be symmetric because the mixed partials are equal. If we evaluate the Hessian matrix at $(1, 2)$ we see that

$$Hf(1, 2) = \begin{bmatrix} 6 & 6 \\ 6 & 6 \end{bmatrix}$$