Chapter 8

In-Class Activity Day 5: Introduction to Vector-Valued Functions

Today we're going to use vectors to track the position and velocity of the Neato!

Learning Objectives

- Introduce the concept of vector-valued functions (vector functions)
- Examine one class of vector functions: parametric equations
- Build a motion model for the position, velocity, and heading of the Neato using vectors

8.1 Vector-Valued Functions and Parametric Curves

In the most general sense, a vector-valued function (equivalently called a vector function) can take n^{th}-order vectors as input, and yield m^{th}-order vectors as output. For the Rainbow Road project, we will focus on one class of vector functions, which take (real) scalar variables and transform them into (real) vectors in 3-space, $\mathbb{R}^3$ (in this notation, the $\mathbb{R}$ symbol indicates the space of all real numbers, and the 3 indicates the dimensionality of our vector). This class of real-value functions is called a parametric equation (or parametric curve). Within the lens of mobile robotics, vector functions occur frequently when describing the *state* of the robot (where it is, and what it is doing) or the trajectory which it takes. Let's spend some time getting to know parametric curves and some of their properties.

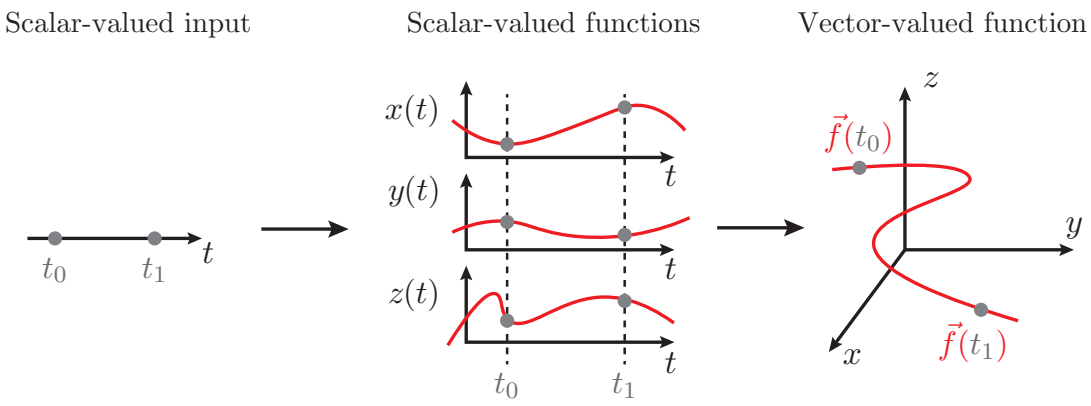


Figure 8.1: Vector-valued function, $\vec{f}$, transforms a scalar, t, to a vector, $\vec{f}(t)$ in $\mathbb{R}^3$.

8.1.1 Position

Describing the path that a robot will take, or the trajectory of any object in kinematics, is fundamental to controller development, localization, state estimation, and any complex engineering system. *Position* in space can be represented as a vector function that takes in the independent (scalar) variable/input time, t, and outputs the dependent variables as a vector in 3-space, e.g., which we can express in terms of the Cartesian coordinates, x, y, and z. The complete set of (x, y, z) values is called a path (or trajectory or curve). The position along the path changes as t increments. Typically the position vector is denoted by $\vec{r}(t)$ where

$$\vec{r}(t) = (x(t), y(t), z(t)) \tag{8.1}$$

When the coordinates functions are written out separately as individual (scalar) equations,

$$x(t) = f_1(t), \qquad y(t) = f_2(t), \qquad z(t) = f_3(t) \tag{8.2}$$

they are referred to as *parametric equations* where t is the parameter. So in this course, we will call the path that describes the robot's trajectory a *parametric* or *parameterized curve*. Let's consider an example: the following parametric equations describe the path/trajectory of a circular helix:

$$x(t) = \cos(t), \qquad y(t) = \sin(t), \qquad z(t) = \frac{t}{2\pi} \tag{8.3}$$

A plot of $\vec{r}(t) = (x(t), y(t), z(t))$ on the domain $t \in [0, 4\pi]$ is depicted in the figure below:

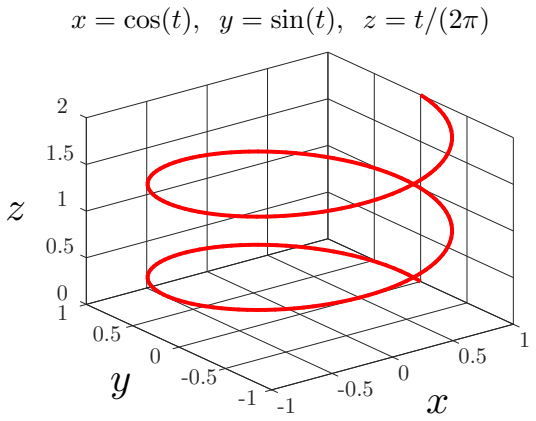


Figure 8.2: Plot of circular helix path.

We can generate a plot of this path in MATLAB via the plot3 command:

```
%Define t as a vector of 201 evenly spaced points
%on the range [0,4*pi] (including the endpoints)
t = linspace(0., 4*pi, 201);
%Evaluate x(t), y(t), z(t) for the
%values defined in the vector t
x = cos(t);   y = sin(t);   z = t/(2*pi);
%generate a 3D plot of [x(t),y(t),z(t)]
plot3 (x,y, z)
```

Alternatively, we could use symbolic variables and fplot3:

```
%Define t as a symbolic variable
syms t
%Define x(t), y(t), and z(t) as symbolic expressions
x = cos(t); y = sin(t); z = t/(2*pi);
%use fplot3 to plot the f(t) = [x(t),y(t),z(t)]
%on the range t = [0 to 4*pi]
fplot3(x,y, z, [0, 4*pi])
```

Exercise 8.1

Using MATLAB, make plots of the circular helix path using both sets of code. Consider changing the values in these equations to see how the path changes (for instance, setting $x = A * \cos(t)$ and adjusting the value of A). Do your predictions about how the path will change based on your modifications match your predictions?

8.1.2 Velocity

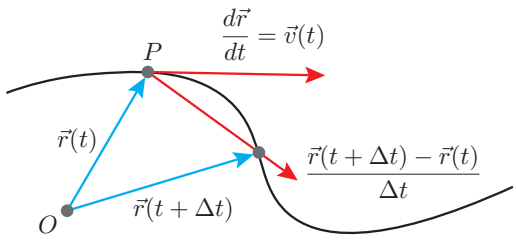


Figure 8.3: For a particle, P, moving along a path, its velocity is the time derivative of its position.

It naturally follows that *velocity* can also be described using a vector-valued function. It is the derivative of the position function with respect to time:

$$\vec{v}(t) = \lim_{\Delta t \to 0} \frac{\vec{r}(t+\Delta t) - \vec{r}(t)}{\Delta t} = \frac{d\vec{r}(t)}{dt}. \tag{8.4}$$

As seen in the figure above, the velocity vector is tangent to the path at point P.

Exercise 8.2

Continuing from our helix example, what is the expression for the velocity along the helical path?

$$\vec{v}(t) = \frac{d\vec{r}(t)}{dt} \tag{8.5}$$

Intuitively, what does your solution imply about the velocity around a helix?

8.1.3 Speed

The speed of a particle is equal to the magnitude of its velocity:

$$\text{speed} = ||\vec{v}(t)|| \tag{8.6}$$

Note that speed is a non-negative scalar quantity that indicates how fast the particle is traveling.

Exercise 8.3

1. Compute the expression for the speed of the particle from the helix example.
2. Use your expression for the speed to find the total distance that the particle travels (i.e. the path length) from $t = 0$ to $t = 4\pi$

8.1.4 Unit Tangent Vector

The *unit tangent vector*, $\hat{T}$, is the vector of length 1 that is tangent to the parametric curve at any given point:

$$\hat{T} = \frac{\vec{v}(t)}{||\vec{v}(t)||} = \frac{d\vec{r}(t)/dt}{||d\vec{r}(t)/dt||} \tag{8.7}$$

Comment: A unit vector is a vector of length 1. Given an arbitrary vector $\vec{u}$, the unit vector that points in the same direction of $\vec{u}$ is:

$$\hat{u} = \frac{\vec{u}}{||\vec{u}||}, \qquad \rightarrow \qquad \vec{u} = ||\vec{u}||\ \hat{u} \tag{8.8}$$

where $||\vec{u}||$ denotes the magnitude (or length) of $\vec{u}$, and that little hat, $\hat{}$, indicates that $\hat{u}$ is a vector of length 1.

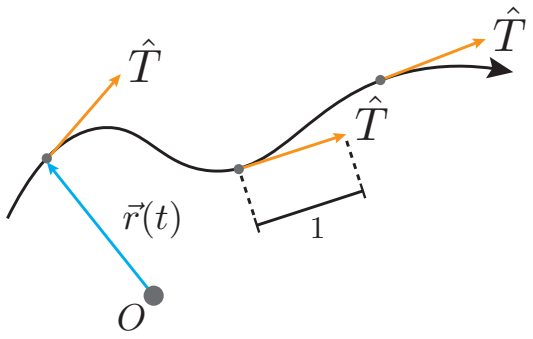


Figure 8.4: The tangent vector, $\hat{T} = \vec{v}(t)/|\vec{v}(t)|$ is the unit vector that is tangent to the curve, $\vec{r}(t)$.

Exercise 8.4

1. Compute an expression for the unit tangent vector (as a function of t) for the helical path.
2. In MATLAB, update the 3D plot of the helix from the previous exercise to include plots of the tangent vector for at least **two different** values of t on the range $[0, 4\pi]$:

8.1.5 Acceleration

The *acceleration* of a particle, $\vec{a}(t)$ is the rate of change of its velocity (and the second-derivative of its position):

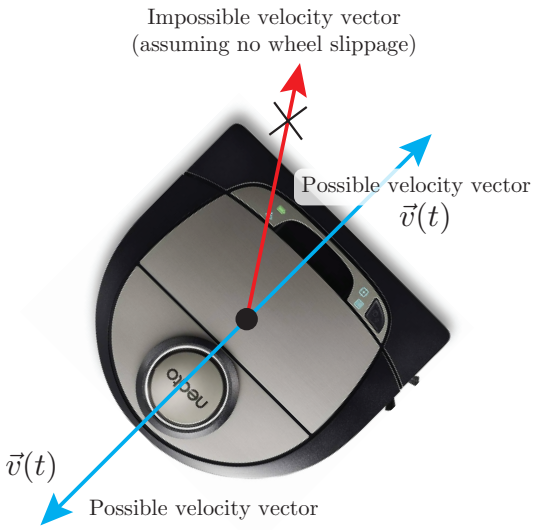
$$\vec{a}(t) = \frac{d\vec{v}(t)}{dt} = \frac{d^2\vec{r}(t)}{dt^2} \tag{8.9}$$

Exercise 8.5

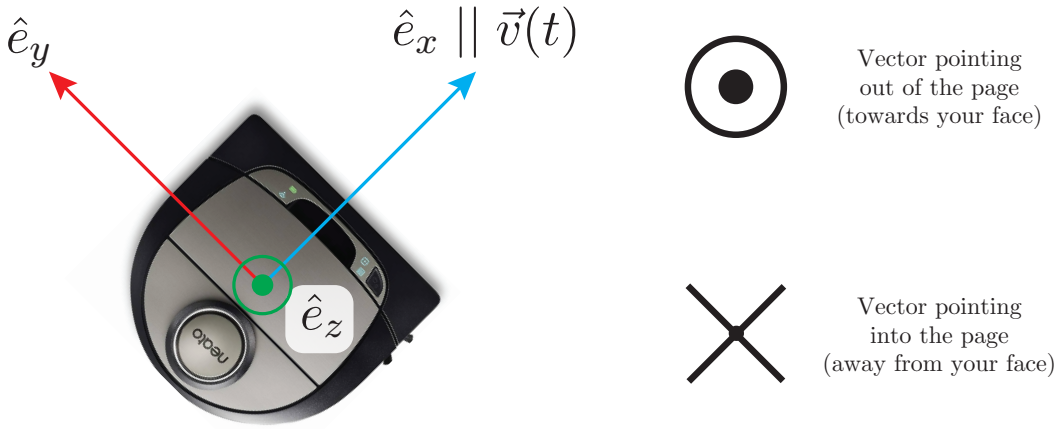
Compute the expression for the acceleration of the particle from the helix example.

8.2 Building a Motion Model for the Neato

Motion models describe how an object or vehicle moves in the world. For our Neato, its mechanical design (e.g., position of the wheels on the chassis) ensures that its local velocity vector is always pointing straight ahead (or straight behind). In other words, the Neato is not capable of moving sideways, it can only move forwards, backwards, rotate, or simultaneously rotate while moving either forwards or backwards.



Therefore, in order to travel along a path, the Neato will have to continuously rotate in order to align its heading direction to the geometry of the path of travel (see this video for a notable exception to this rule). In order to describe the motion of the Neato, we can define three unit vectors: $(\hat{e}_x, \hat{e}_y, \hat{e}_z)$ that have been fixed to the Neato. In other words, these vectors are *body-fixed*: think of them as being glued onto the Neato.



For today, let's focus on $\hat{e}_x$, the unit vector that points in the forward direction of the Neato. Since the Neato can only travel forwards/backwards and not side-to-side, we see that $\hat{e}_x$ must be parallel to the Neato's velocity vector, $\vec{v}(t)$:

$$\hat{e}_x \,||\, \vec{v}(t) \tag{8.10}$$

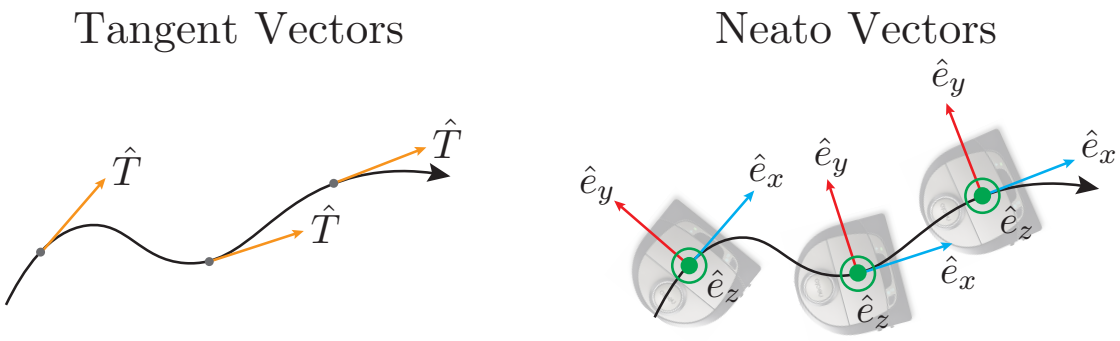
Here, $\vec{A} \,||\, \vec{B}$ can be read as "vectors $\vec{A}$ and $\vec{B}$ are parallel to one another". Note that $\hat{e}_x$ and $\vec{v}(t)$ could potentially point in opposite directions (if the Neato were driving backwards), though they would still be parallel to one another. The unit tangent vector, $\hat{T} = \vec{v}(t)/||\vec{v}(t)||$ is also parallel to the velocity vector, and is therefore parallel to $\hat{e}_x$:

$$\hat{T} \,||\, \vec{v}(t), \quad \rightarrow \quad \hat{T} \,||\, \hat{e}_x \tag{8.11}$$

Since $\hat{e}_x$ and $\hat{T}$ both have a length of 1 and are parallel to one another, they are either equal or opposites:

$$\hat{e}_x = \pm\hat{T} \tag{8.12}$$

In other words, they are equal to one another ($\hat{e}_x = \hat{T}$) when the Neato is moving forwards and are opposites ($\hat{e}_x = -\hat{T}$) when the Neato is moving backwards ($\hat{T}$ is undefined when the Neato is at rest).



Equipped with these definitions, let's analyze some data collected from the Neato: we just took it out for a drive (a video can be found here), and have recorded its position as a function of time. The position data has been provided as the file 'neato_position_data.mat', which can be found on the Canvas page for this assignment. The file contains three variables: t_list, x_list, and y_list, which correspond a list of measurements of t, $x(t)$, and $y(t)$ respectively. Note that t has units of seconds, and both $x(t)$ and $y(t)$ have units of meters. We can load these variables into the MATLAB environment using the following block of code (which relies on the load command):

```
%define the path of the file location as a string
my_path = 'C:\Users\taylorott\Documents\';
%define the file name as a string
fname = 'neato_data_day_assignment05.mat';
%load the data stored in the .mat file into neato_data (a MATLAB struct)
neato_data = load([my_path,fname]);
%unpack neato_data into tlist, xlist, and ylist
tlist = neato_data.tlist; xlist = neato_data.xlist; ylist = neato_data.ylist;
```

Note that the variable neato_data is a MATLAB struct (a combination of variable containing multiple sub-variables) that we unpack into t_list, x_list, and y_list. In the next exercise, we will analyze this data using what we have learned about parametric curves.

Exercise 8.6

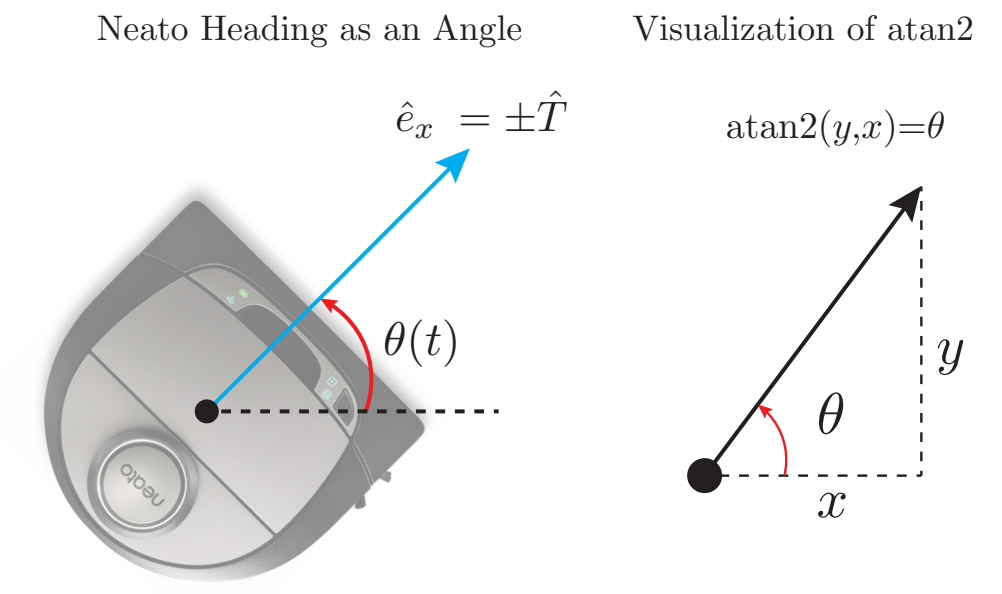
1. Generate a plot of the path traveled by the Neato. Make sure to label your coordinate axes (and provide units).

2. Approximate the Neato's velocity, $\vec{v}(t)$, from the measured data using the finite difference (secant) approximation. Generate plots of:

$$x(t) \text{ vs. } t, \qquad y(t) \text{ vs. } t, \qquad v_x(t) \text{ vs. } t, \qquad v_y(t) \text{ vs. } t \tag{8.13}$$

   We'd recommend that this be done in a single figure with two separate subplots (the top subplot for the position plots, and the bottom for the velocity plots).

3. Use your velocity approximation to compute the Neato's speed, $||\vec{v}(t)||$, and then compute the tangent vector, $\hat{T} = \vec{v}(t)/||\vec{v}(t)||$. Use the tangent vector to find the orientation of the Neato, $\theta(t)$ (shown in the figure below). We recommend that you use MATLAB's atan2 function to compute $\theta(t)$ from $\hat{T}$. Essentially, atan2(y, x) computes the angle that the vector $\vec{V} = (x, y)$ forms with the horizontal axis (this is visualized in the figure below). Plot the Neato's speed, $||\vec{v}(t)||$, and heading, $\theta(t)$, as functions of time on two separate sets of axes (we recommend that you generate a single figure with two subplots).



4. Add tangent vectors (for at least two different time points) to the path plot you generated in the first part of this exercise. If you want to make your figure look nicer, you may want to scale the lengths of the tangent vectors when plotting them.

5. Are the plots that you generated consistent with the behavior that you observed in the video? Why or why not?