# CS81 Homework 2

Andrew Sosanya, Dev Jhaveri, Anne Sherrill, Lisa Oh

October 4, 2018

# 1 Problem 1 - Spiderman

## 1.1 Forward Kinematics

We use the relations

$$x = \frac{d^2 - r^2 + R^2}{2d}$$
$$y = \sqrt{\frac{4d^2R^2 - (d^2 - r^2 + R^2)^2}{4d^2}}$$

to find the $(x, y)$ where the circles shaped by $l_1$ and $l_2$ intersect.

$$\text{Plug in: } d = \frac{1}{2}, R = l_1, r = l_2$$
$$x = \frac{(\frac{1}{2})^2 - l_2^2 + l_1^2}{2 * \frac{1}{2}} = \frac{1}{4} + l_1^2 - l_2^2$$
$$y = \sqrt{\frac{4(\frac{1}{2})^2 l_1^2 - ((\frac{1}{2})^2 - l_2^2 + l_1^2)^2}{4(\frac{1}{2})^2}} = -\sqrt{-\frac{1}{16} + \frac{l_1^2}{2} + \frac{l_2^2}{2} + 2l_1^2 l_2^2 - l_1^4 - l_2^4}$$

We choose the negative square root when calculating y because we know that Spiderman is at a negative y value, based on our coordinate system with y=0 at the connection of the web to the buildings.

## 1.2 Differential Kinematics

Starting with the equations:

$$x = \frac{1}{4} + l_1^2 - l_2^2$$
$$y = -\sqrt{l_1^2 - (\frac{1}{4} + l_1^2 - l_2^2)^2} = -\sqrt{-\frac{1}{16} + \frac{l_1^2}{2} + \frac{l_2^2}{2} + 2l_1^2 l_2^2 - l_1^4 - l_2^4}$$

We take the partials of $x$ and $y$ with respect to $l_1$ and $L_2$ to get the Jacobian matrix and multiply the Jacobian matrix by the time derivatives of $l_1$ and $l_2$ to get $\dot{x}, \dot{y}$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial l_1} & \frac{\partial x}{\partial l_2} \\ \frac{\partial y}{\partial l_1} & \frac{\partial y}{\partial l_2} \end{bmatrix} \begin{bmatrix} \dot{l_1} \\ \dot{l_2} \end{bmatrix} = \begin{bmatrix} 2l_1 & -2l_2 \\ -\frac{l_1+4l_1l_2^2-4l_1^3}{2\sqrt{-\frac{1}{16}+\frac{l_1^2}{2}+\frac{l_2^2}{2}+2l_1^2l_2^2-l_1^4-l_2^4}} & -\frac{l_2+4l_1^2l_2-4l_2^3}{2\sqrt{-\frac{1}{16}+\frac{l_1^2}{2}+\frac{l_2^2}{2}+2l_1^2l_2^2-l_1^4-l_2^4}} \end{bmatrix} \begin{bmatrix} \dot{l_1} \\ \dot{l_2} \end{bmatrix} \tag{1}$$

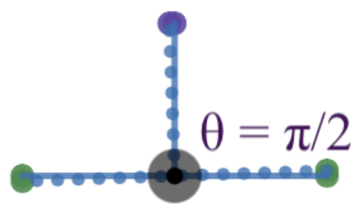
### 1.3 Inverse Kinematics

The inverse kinematics can be determined geometrically using the distance formula. If $(x, y)$ is known, $l_1 = \sqrt{x^2 + y^2}$ and $l_2 = \sqrt{(1/2 - x)^2 + y^2}$

## 2 Problem 2 - Drawbot

### 2.1 Modeling the Situation

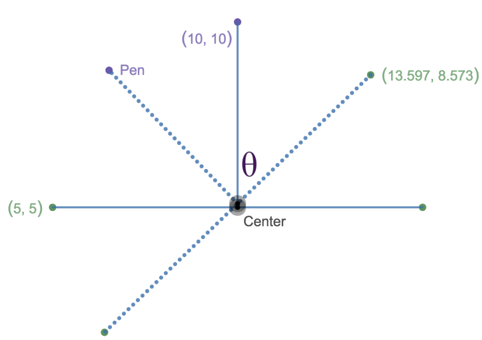
We chose to have the pen on the perpendicular bisector of the line segment that connects the contact points of the wheels, which also lines up with the center of the turtlebot. We chose the turtle bot to have its wheels 10cm apart and have the pen be 5cm in front of the




robot's center.

### 2.2 Forward Kinematics

Rotating the frame a little bit, keeping the same $(x, y)$, we see that we can parameterized the location of the pen as $(x_p, y_p)$, where $x_p = x + 5\cos(\theta)$ and $y_p = y + 5\sin(\theta)$



## 2.3 Differential Kinematics

We write the time derivatives of the pen's motion, $(\dot{x_p}, \dot{y_p})$ as the product of the Jacobian $J$ as a function of $\dot{q}$ and the time derivative $\dot{p}$ of the turtlebot's motion, $(\dot{x}, \dot{y}, \dot{\theta})$, such that $J\dot{q} = \dot{p}$.

$$\begin{bmatrix} \dot{x_p} \\ \dot{y_p} \end{bmatrix} = \begin{bmatrix} \frac{\partial x_p}{\partial x} & \frac{\partial x_p}{\partial y} & \frac{\partial x_p}{\partial \theta} \\ \frac{\partial y_p}{\partial x} & \frac{\partial y_p}{\partial y} & \frac{\partial y_p}{\partial \theta} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -5\sin\theta \\ 0 & 1 & 5\cos\theta \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

## 2.4 Controlling the Pen

Consider the radius of the turtlebot's wheels $r$. We say that the velocity of the entire robot can be calculated as $v = \frac{r\omega_1 + \omega_2}{2}$, where $\omega_1$ and $\omega_2$ are the angular velocities of each of the turtlebot's wheels. Considering the differential drive configuration of the robot, the angular velocity of the robot is $\omega = \frac{r\omega_1 - \omega_2}{2b}$.

We have that $J\dot{q} = \begin{bmatrix} \dot{x_p} \\ \dot{y_p} \end{bmatrix}$

Because our outputs will be $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$, which will effect $\dot{q}$, we can start by inverting the Jacobian:

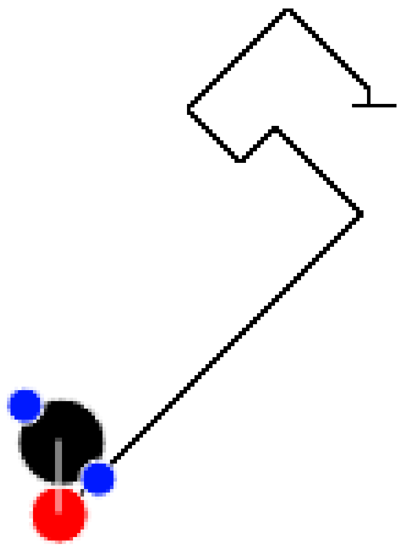
$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{x_p} \\ \dot{y_p} \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix} = \begin{bmatrix} r\cos\theta(\frac{\omega_1+\omega_2}{2}) \\ r\sin\theta(\frac{\omega_1+\omega_2}{2}) \\ r(\frac{\omega_1-\omega_2}{2b}) \end{bmatrix} = \frac{r}{2}(\omega_1 \begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{1}{b} \end{bmatrix} + \omega_2 \begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{-1}{b} \end{bmatrix}) = \frac{r}{2}( \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ \frac{1}{b} & \frac{-1}{b} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix})$$

Let's call $\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ \frac{1}{b} & \frac{-1}{b} \end{bmatrix} = J'$. Therefore $\frac{r}{2}J' \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{x_p} \\ \dot{y_p} \end{bmatrix}$ and $\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \frac{2}{r} J'^{-1} J^{-1} \begin{bmatrix} \dot{x_p} \\ \dot{y_p} \end{bmatrix}$

And now, we have an equation that takes in our desired velocities for the pen, $(\dot{x_p}, \dot{y_p})$, and computes the wheel velocities needed to conduct the motion.

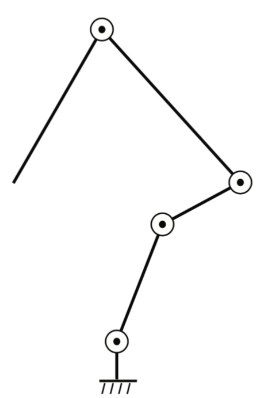
### 2.5 Simulating the System



$$(2)$$

Code to run the simulation can be found in the submission in The object-oriented simulation implements the methods used in 2c to calculate the $\dot{q}$ and model the trajectories of the pen and robot. We used the cs1lib graphic library and NumPy for the matrix operations. The robot is marked by the black circle, the pen in red, and the wheels of the robot in blue.

## 3 Problem 3 - Robot Arm Geometry



$$(3)$$

### 3.1 Estimating the Jacobian

From eyeballing the robot, the coordinates of the four revolute joints and end effector appear to be (assuming the axis is that the origin):

$$r_0 = (0,0), r_1 = (3,4), r_2 = (5,5), r_3 = (0,10), EE = (-4,5).$$

The vectors from each joint to the end effector are therefore:

$$v_0 = [-4,5], v_1 = [-7,1], v_2 = [-9,0], v_3 = [-4,-5]$$

Rotating each vector gives us the four columns of the Jacobian:

$$J = \begin{bmatrix} -5 & -1 & 0 & 5 \\ -4 & -7 & -9 & -4 \end{bmatrix}$$

### 3.2 Estimating Workspace Velocity

We plug in the values $(1,1,3,1)$ for $[\theta_1, \theta_2, \theta_3, \theta_4]$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} = \begin{bmatrix} -5 & -1 & 0 & 5 \\ -4 & -7 & -9 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -42 \end{bmatrix}$$

### 3.3 Analytical Jacobian

The forward kinematics of this system is:

$$x_{EE} = l_1 \cos_1 + l_2 \cos_{12} + l_3 \cos_{123} + l_4 \cos_{1234}$$

$$y_{EE} = l_1 \sin_1 + l_2 \sin_{12} + l_3 \sin_{123} + l_4 \sin_{1234}$$

The analytical Jacobian would be: $\begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} & \frac{\partial x}{\partial \theta_4} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} & \frac{\partial y}{\partial \theta_4} \end{bmatrix} =$

$$\begin{bmatrix} -l_1 \sin_1 - l_2 \sin_{12} - l_3 \sin_{123} - l_4 \sin_{1234} & -l_2 \sin_{12} - l_3 \sin_{123} - l_4 \sin_{1234} & -l_3 \sin_{123} - l_4 \sin_{1234} & -l_4 \sin_{1234} \\ l_1 \cos_1 + l_2 \cos_{12} + l_3 \cos_{123} + l_4 \cos_{1234} & l_2 \cos_{12} + l_3 \cos_{123} + l_4 \cos_{1234} & l_3 \cos_{123} + l_4 \cos_{1234} & l_4 \cos_{1234} \end{bmatrix}$$

# 4 Problem 2 - Turtlebots Arise!

### 4.1 Timed_drive

In our video, our turtlebot drives for 4 seconds at a velocity of 0.1 m/sec. It travels 0.4 meters, or 40 centimeters.

## 4.2 Timed_right and Timed_left

In our video, our turtlebot drives for 5 seconds at a velocity of $-\pi/10$ and $\pi/10$ radians/sec, respectively, turning $\pi/2$ radians to the right and left.

## 4.3 Driving in a square

Our turtlebot drives forward with a velocity of 0.1 m/sec and turns with a velcity of $-\pi/10$ radians/sec.