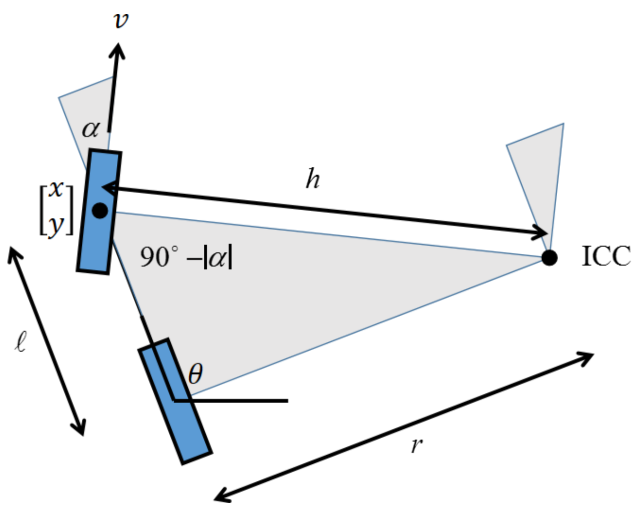
EECS4421: Lab 5

Thu 9 Mar, 2017
Due: Fri 24 Mar, 2017 by end of class

1 Velocity motion model for a bicycle

Consider the bicycle shown in the figure below.



The length of the bicycle frame l is the distance between the centers of the front and back wheels. The pose of the bicycle is defined by the location of the center of the front wheel $[x\ y]^T$ in meters and the angle θ measured from the x axis to the direction of the bicycle frame. The bicycle is steered by pointing the front wheel at an angle α relative to the frame of the bicycle (turning the front wheel counter-clockwise relative the bicycle frame results in a positive value for α, and turning the front wheel clockwise relative the bicycle frame results in a negative value for α). The forward linear velocity of the front wheel is v meters per second.

2 Question 1

Create a Matlab function `[newpose, icc] = bicycle_forward(pose, control, L, dt)` that computes the forward kinematics of the bicycle. `pose` is the current pose of the bicycle $[x\ y\ \theta]^T$, `control` is the vector of control inputs $[v\ \alpha]^T$, `L` is the length of the bicycle frame, and `dt` is the time interval of the motion in seconds. `newpose` is the new pose of the bicycle, and `icc` is the location of the instantaneous center of curvature.

To compute `newpose`, first compute h (shown in the figure), use h to compute `icc`, compute the angular velocity ω around the ICC, and then compute the new pose using the formula from the Day 14 lecture slides.

3 Question 2

Create a Matlab function `newpose = sample_bicycle(pose, control, L, dt)` that uses your `bicycle_forward` function to generate a sample pose of the bicycle `newpose` given the starting pose `pose`, control vector `control`, frame length `L`, and time interval `dt`.

To generate a sample pose, assume that the control inputs are noisy. That is, `control` is the noise-free control vector $[v\ \alpha]^T$ but the actual control vector is made up of a noisy velocity $\hat{v}$ and a noisy steering angle $\hat{\alpha}$. Assume that the $\hat{v} = v + \varepsilon_v$ where ε_v is a zero-mean Gaussian random variable with variance $\sigma_v^2 = 0.1v^2$ (i.e., the noise is proportional to velocity squared). Assume that $\hat{\alpha} = \alpha + \varepsilon_\alpha$ where ε_α is a zero-mean Gaussian random variable with variance $\sigma_\alpha^2 = (5°)^2$. Using `control`, generate a random noisy control vector `noisycontrol`, and call your `bicycle_forward` function with `pose`, `noisycontrol`, `L`, and `dt` to compute `newpose`.

Important: Place limits on the value of $\hat{\alpha}$ such that $|\hat{\alpha}| \leq 80°$.

4 Question 3

Using your `sample_bicycle` function, create 5 simulations showing samples of the bicycle position after moving over one interval of time `dt = 1` second. Assume a bicycle frame length $L = 1$ meter and a starting position $[x\ y]^T = [0\ 0]^T$. For each simulation, plot 1000 samples of `newpos` computed using `sample_bicycle`. The 5 simulations should use different values for the control inputs as shown in the table below:

simulation	α	v
1	$25°$	$1m/s$
2	$-25°$	$1m/s$
3	$25°$	$5m/s$
4	$75°$	$1m/s$
5	$75°$	$5m/s$

Use the same axis scales for all 5 of your plots. Make sure to use `axis equals` so that 1 unit on the x axis of your plot is equal in length to 1 unit on the y axis of your plot.

5 Submit

Create a report (like you did for Lab 1) containing the results of your simulations. Submit your files using

```
submit 4421 L5 <your submission files>
```