MEAM 520
Mobile Robots

Katherine J. Kuchenbecker, Ph.D.
General Robotics, Automation, Sensing, and Perception Lab (GRASP)
MEAM Department, SEAS, University of Pennsylvania

Lecture 22: December 6, 2012

MEAM.Design : MEAM520-12C-P02-Rendering

GENERAL
Hall of Fame
Laboratories
Contact Info

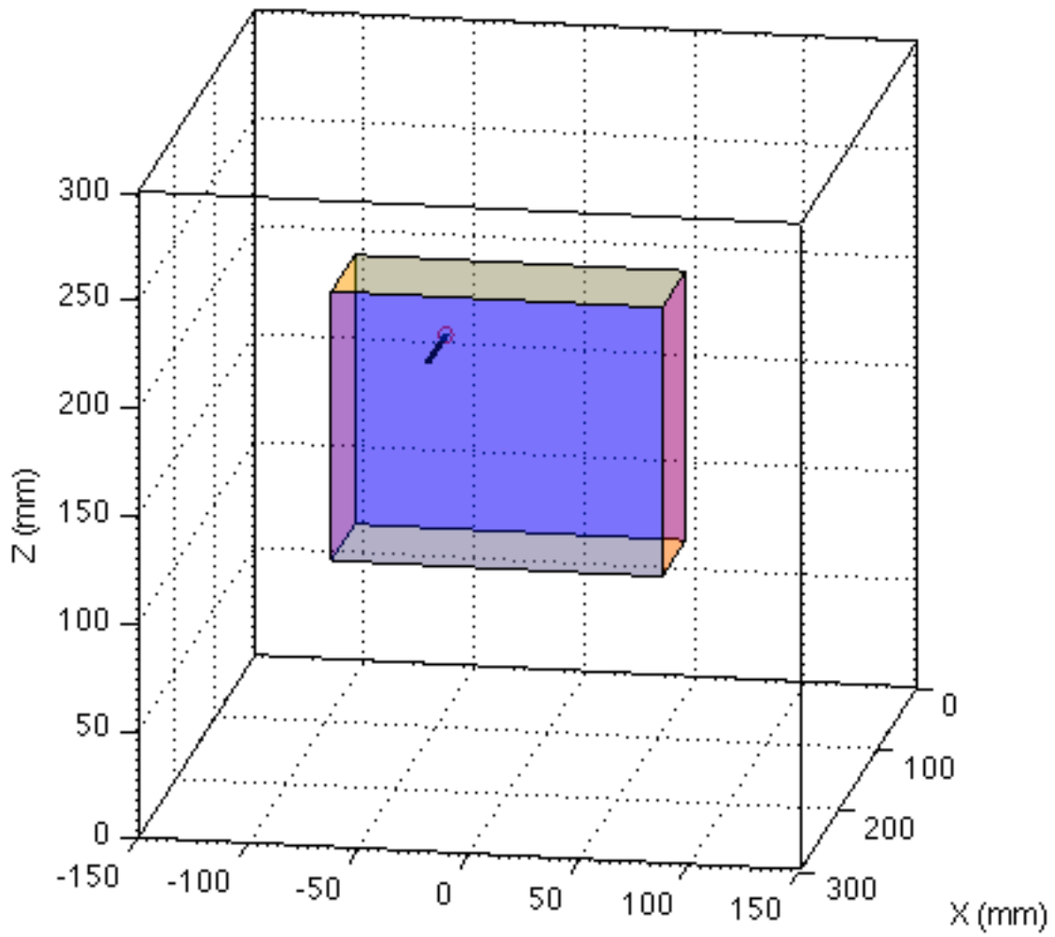
COURSES
MEAM 101
MEAM 201
MEAM 410/510
MEAM 520
IPD 501
SAAST

GUIDES
Materials
Laser Cutting
3D Printing
Machining
ProtoTRAK
PUMA 260
PHANToM
BeagleBoard
MAEVARM
Phidget
Tap Chart

SOFTWARE
SolidWorks
Matlab

Now that you have your team, it's time to get to work on project 2. This assignment is due by **5:00 p.m. on Tuesday, December 4**.

Start by downloading the starter code (v1). We are providing you with p-coded versions of all the functions described in the Phantom Guide. For example, calling `phantomStart(false);` starts the simulated phantom so you can work on your code on any computer. Instead of getting encoder readings from the real PHANToM, the system simulates the presence of a human user by reading a pre-recorded trajectory from the included `encsHistory.mat` file.

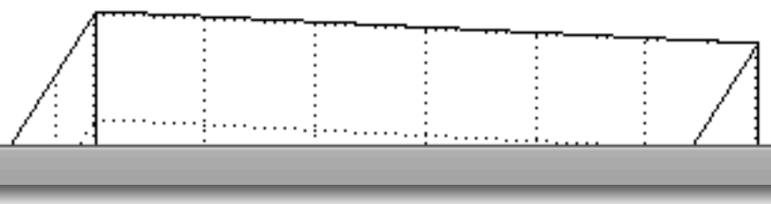


Demo: Haptic Box

Run `haptic_box_demo.m` and look at how it is written. This demonstration creates a virtual haptic box for the user to feel, as seen in the top illustration at right. The user is trapped inside the virtual box and feels a virtual spring force each time they contact a wall. The position of the PHANToM tip is shown as a red circle, the box is shown in transparent colors, and a scaled version of the force vector is shown as a thick black line.

The system simulates the presence of a human user by default because you probably don't have a PHANToM connected to your computer. Look at how the forces Fx, Fy, and Fz are calculated from the positions hx, hy, and hz. This is the type of mapping you will need to create in this assignment.

Once you understand how the haptic box demo works, your team's task on this project is to complete the following two haptic rendering scenes created for the PHANToM.



Task 1: Haptic Ball

Complete the **haptic ball** scene that has been started for you in `haptic_ball_team50.m`. Change the filename to match

T 12/4 6:30-7:30pm
W 12/5 12:30-1:30pm
R 12/6 6:30-7:30pm
T 12/11 4:00-5:00pm
T 12/11 5:00-6:00pm
W 12/12 12:30-1:30pm
W 12/12 1:30-2:30pm
R 12/13 6:30-7:30pm

Homework 6: Teleoperation

MEAM 520, University of Pennsylvania
Katherine J. Kuchenbecker, Ph.D.

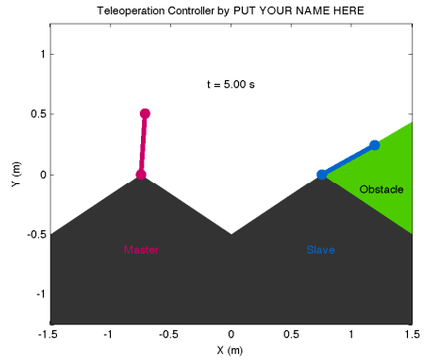
December 3, 2012

This assignment is due on **Friday, December 7**, by 5:00 p.m. If you don't finish by that time, you may turn it in with no penalty by 5:00 p.m. on Wednesday, December 12. After that deadline, no further assignments may be submitted. Because it is short, this assignment is worth 30 points (half the value of homework assignments 1 through 5).

You may talk with other students about this assignment, ask the teaching team questions, use a calculator and other tools, and consult outside sources such as the Internet. To help you actually learn the material, what you submit should be your own work, not copied from a peer or a solution manual.

Teleoperation Controller *(30 points)*

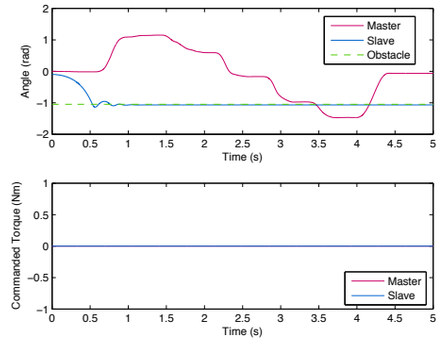
Your task is to write a good controller for a simple simulated teleoperation system. The image below shows a snapshot of the simulated teleoperator. It includes a one-degree-of-freedom master robot (left, in magenta) and an identical one-degree-of-freedom slave robot (right, in blue). Each device consists of a single revolute joint, much like the pair of Immersion Impulse Engine 2000 joysticks that Professor Kuchenbecker discussed in Lecture 18 (on November 20). Each robot's joint angle is measured in radians, with counterclockwise positive and straight up equal to zero.



The stationary bracket to which the robots are attached is shown in dark gray. The robots can move freely through this region because they are not in the same plane. There are no obstacles in the master's workspace, and the robots are too short to touch each other directly. There is one obstacle in the slave's workspace; shown in green, it begins at `obstacleAngle` and extends infinitely in the negative direction. You should move the obstacle around to test different environments; the controller that you write should work for any obstacle location, so it should not use the variable `obstacleAngle` in any way.

To simulate the presence of a human user holding onto the end of the master robot, the master moves through a pre-determined trajectory that you select. Six trajectories are provided (`masterMovement1.mat` ... `masterMovement6.mat`), and you can also write your own. The slave has pre-programmed dynamics that are hidden from your view inside the function `getSlaveTheta.p`. These dynamics include but are not

limited to inertia, gravity, friction, actuator saturation, and encoder quantization. When you first run the starter code, you will see that the slave just falls into the obstacle and stays there, while the master robot follows the default pre-determined trajectory. To help you understand what is happening in the simulation, the starter code animates the entire interaction and graphs the resulting angles and commanded torques over time, as shown in the sample graph below.



The simulated teleoperation system runs a servo loop at 1000 Hz, which you should not change. At each time step, it obtains the new position of the master (`masterTheta`) and the slave (`slaveTheta`) in radians. Your job is to specify the torque to command to the master (`masterTau`) and the slave (`slaveTau`) in newton·meters to yield good transparency (good tracking and good feel in free space, good feel in contact with the obstacle) and good stability (no extraneous ongoing oscillations). There should be no motion scaling or clutching between the two devices. The slave torque that you specify will directly affect the movement of the slave robot, while the master torque that you specify will merely be graphed. Following standard robotics convention, a positive torque moves the joint in the positive direction. It is expected that your controller will include gravity compensation, a proportional term, and a derivative term on both devices.

Download the starter code from this assignment's page on the class wiki, change the name of the provided script (`teleoperation_starter.m`) to include your PennKey, put your name where it says `'PUT YOUR NAME HERE'`, and make sure the starter code works correctly before starting to modify it. Near the top, you can change the movement of the master, the initial position of the slave, the angle of the obstacle, and the speed of the animation. When you're ready, put your controller code between the two lines of stars, modify whatever other simulation settings you want to elucidate the behavior of the system, and comment the final code you write. Follow the instructions below to submit your Matlab files.

Submitting Your Code

Follow these instructions to submit your code:

1. Start an email to `meam520@seas.upenn.edu`
2. Make the subject *Homework 6: Your Name*, replacing *Your Name* with your name.
3. Attach your correctly named MATLAB script (`teleoperation_yourpennkey.m`) to the email, along with any other files that you created. You do not need to submit the provided `masterMovement.mat` or `getSlaveTheta.p` files. Please **do not zip your files together** before attaching them; just attach them as individual files.
4. Optionally include any comments you have about this assignment.
5. Send the email.

You are welcome to resubmit your code if you want to make corrections. To avoid confusion, please state in the new email that it is a resubmission, and include all of your MATLAB files, even if you have updated only some of them.

Due by 5:00 p.m. on Wednesday 12/12

MAGIC 2010
DOWN UNDER
Multi Autonomous
Ground-robotic
International Challenge

Penn Takes Second Place at MAGIC 2010

The University of Pennsylvania finished in second place at the worldwide Multi Autonomous Ground-Robotic International Challenge (MAGIC) 2010 competition, earning the Penn team a research award of $250,000. To compete, the team traveled to Australia, where the event was held in conjunction with the Australian Land Warfare Conference.

The Penn team, consisting of General Robotics, Automation, Sensing and Perception (GRASP) Laboratory members Jon Butzke, Alex Kushleyev, Cody Phillips and Mike Phillips, spent the past few weeks constructing, programming, shipping, and reassembling a team of heterogeneous robots to map, navigate, search, and neutralize objects of interests in a large area using minimal human supervision. The team is led by Daniel Lee, Evan C Thompson Term Associate Professor and Raymon S. Markowitz Faculty Fellow in the department of Electrical and Systems Engineering.

The actual competition consisted of the Penn team searching and mapping a 250,000 square-meter area of the Adelaide Fairgrounds in under three-and-a-half hours, using five sensor robots and two disrupter robots. The team was able to find and "neutralize" eight different items, including both static and mobile objects, during the different phases of the competition.

In a separate challenge competition for members of the media, the Penn team successfully mapped a large 50x150m shed filled with hay mazes and miscellaneous objects in 30 minutes. The team's effort in this phase of the competition netted the winning trophy as well as much interest from the military observers in attendance.

MAGIC 2010, jointly sponsored by the Australian and U.S. Departments of Defense, was organized to attract innovative proposals from worldwide research organizations to develop next-generation fully autonomous ground vehicle systems that can be deployed effectively in

MAGIC 2010
DOWN UNDER

Multi **A**utonomous
Ground-robotic
International **C**hallenge

Demo of a Penn MAGIC robot by James Yang

Magic 2010

Photo Gallery

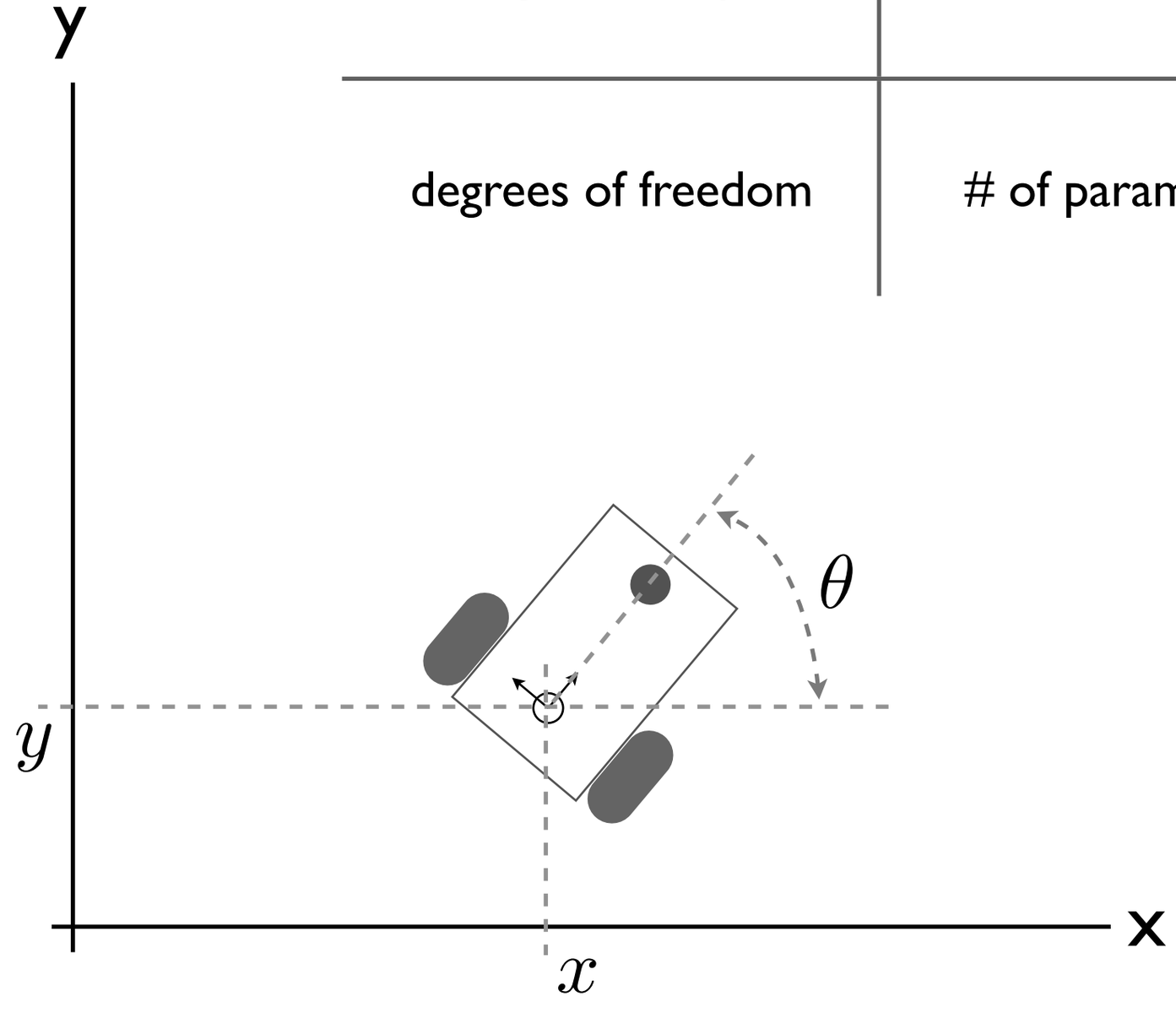
Mobile ground robots typically operate in a planar environment, so their movement is easier to describe than that of a manipulator.

Slides created by
Jonathan Fiene

Kinematics of Mobile Robots

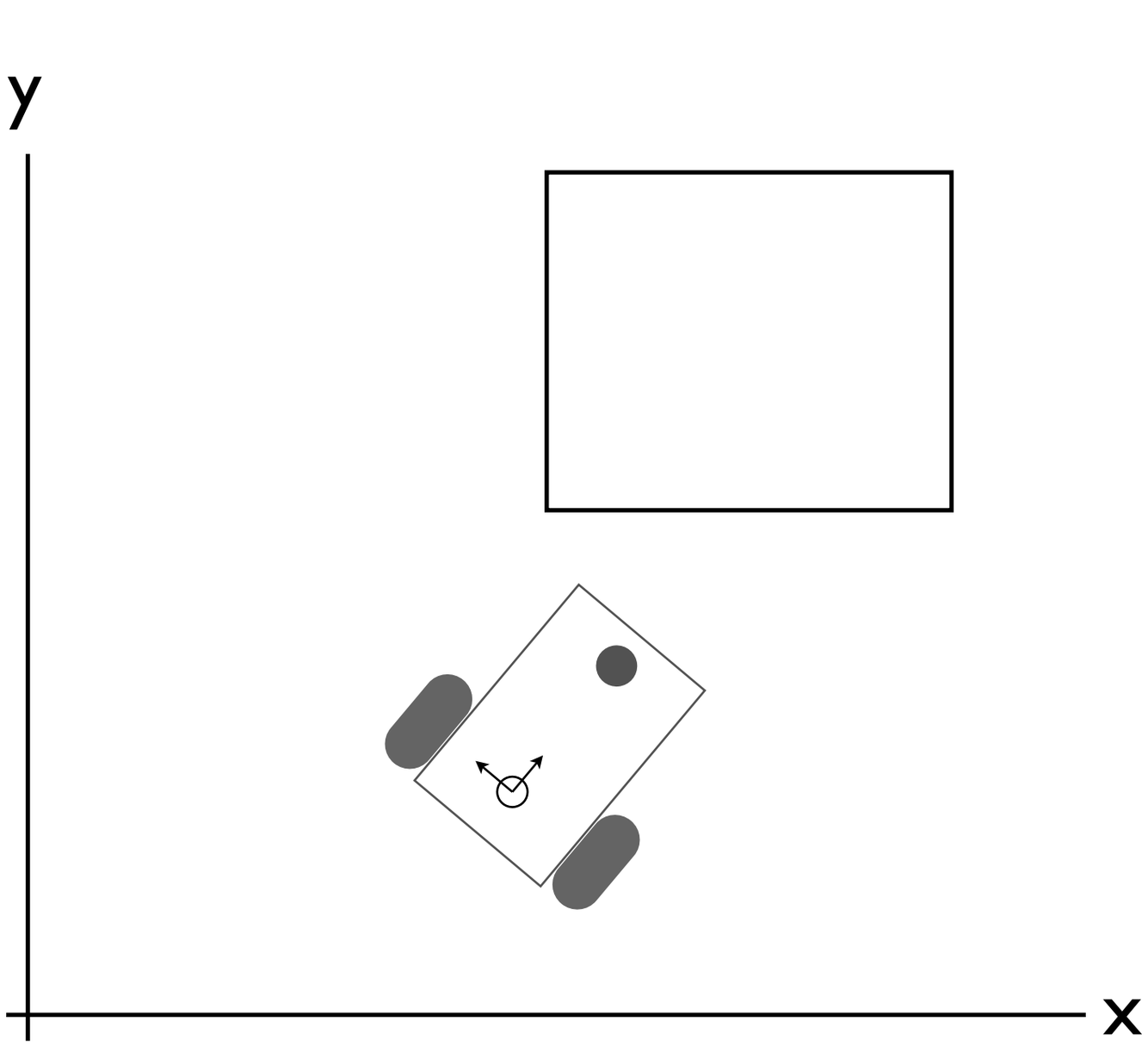
Configurations & Degrees of Freedom

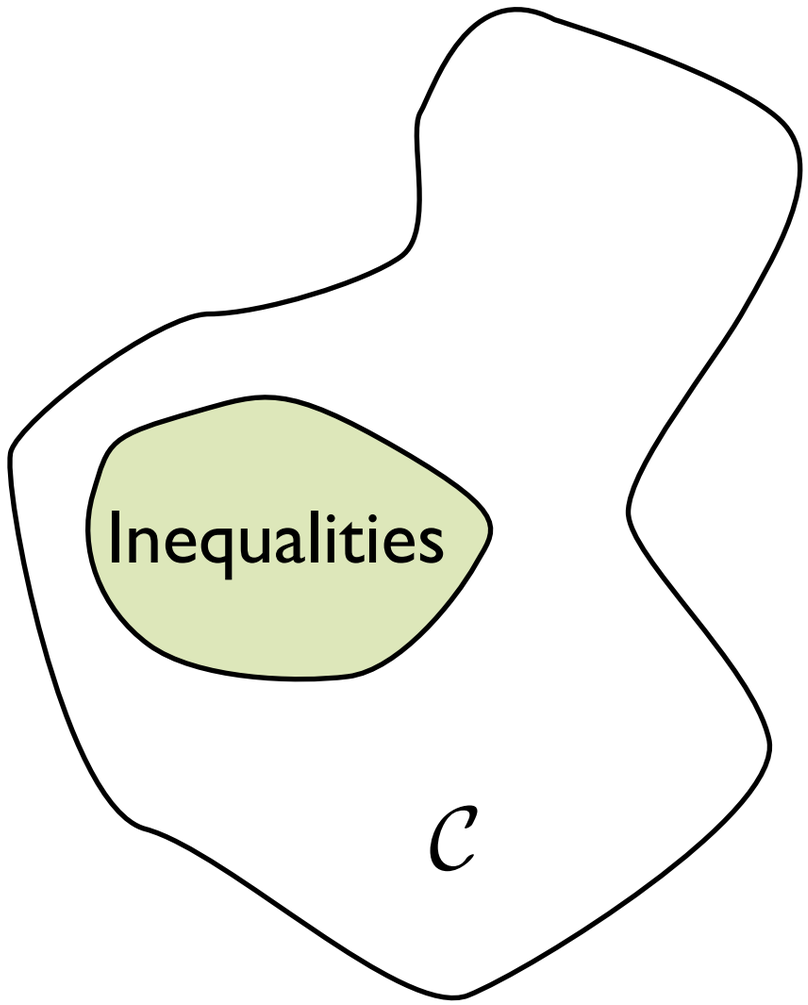
configuration	specification of all points on the robot
configuration space	set of all possible configurations
degrees of freedom	# of parameters to define the configuration



Constraints

Inequality Constraints : No two objects can occupy the same space

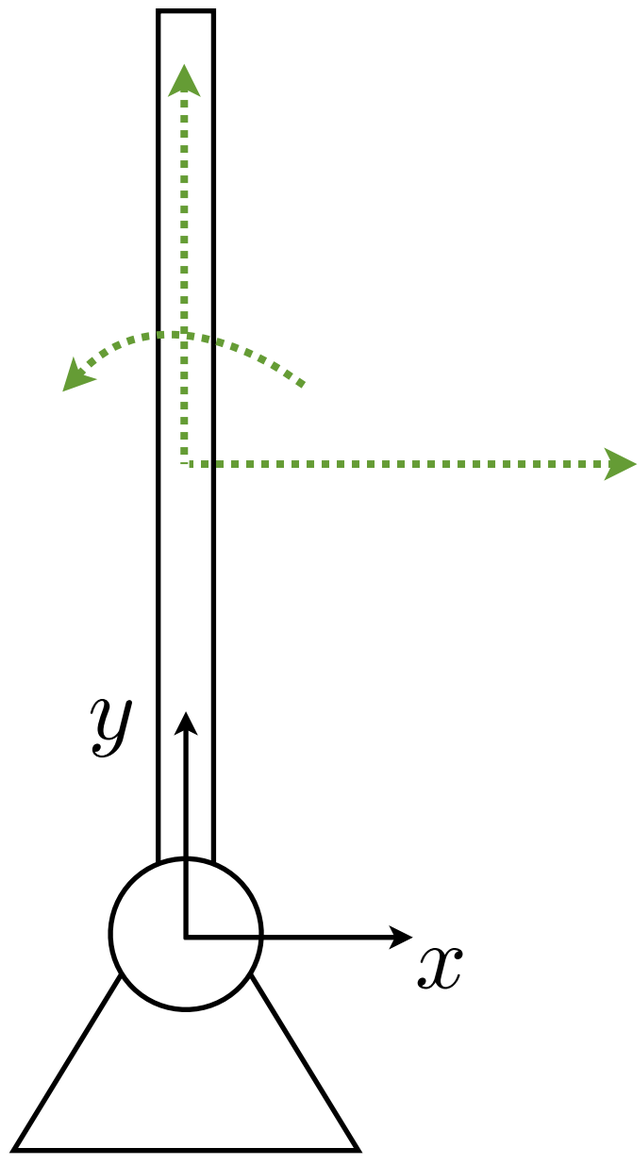




Constraints

Holonomic Constraints : Position is limited to a subset of the configuration space through a function equal to zero

$$f(x_1, \ldots, x_N, t) = 0$$



Free Configuration (pose) :

$$\mathbf{q} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

Holonomic Constraints

$$f(x, y, \theta, t) = 0$$

$$x = -\sin\theta \qquad y = \cos\theta$$

$$x + \sin\theta = 0 \qquad y - \cos\theta = 0$$

Constrained Configuration :

$$\mathbf{q} = \begin{bmatrix} \theta \end{bmatrix}$$

Constraints

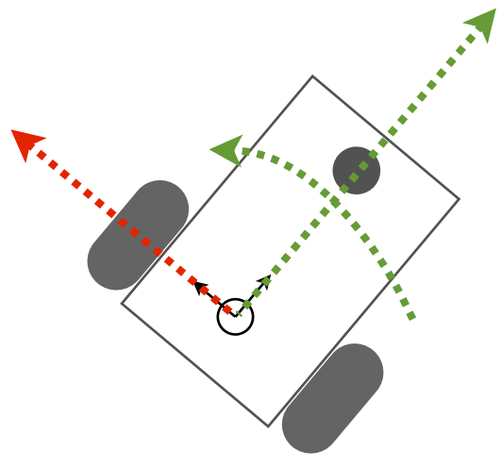
Non-Holonomic Constraints : any constraint that cannot be expressed as a function of the position coordinates, including limits on VELOCITY

y

The robot can reach anywhere in the configuration space

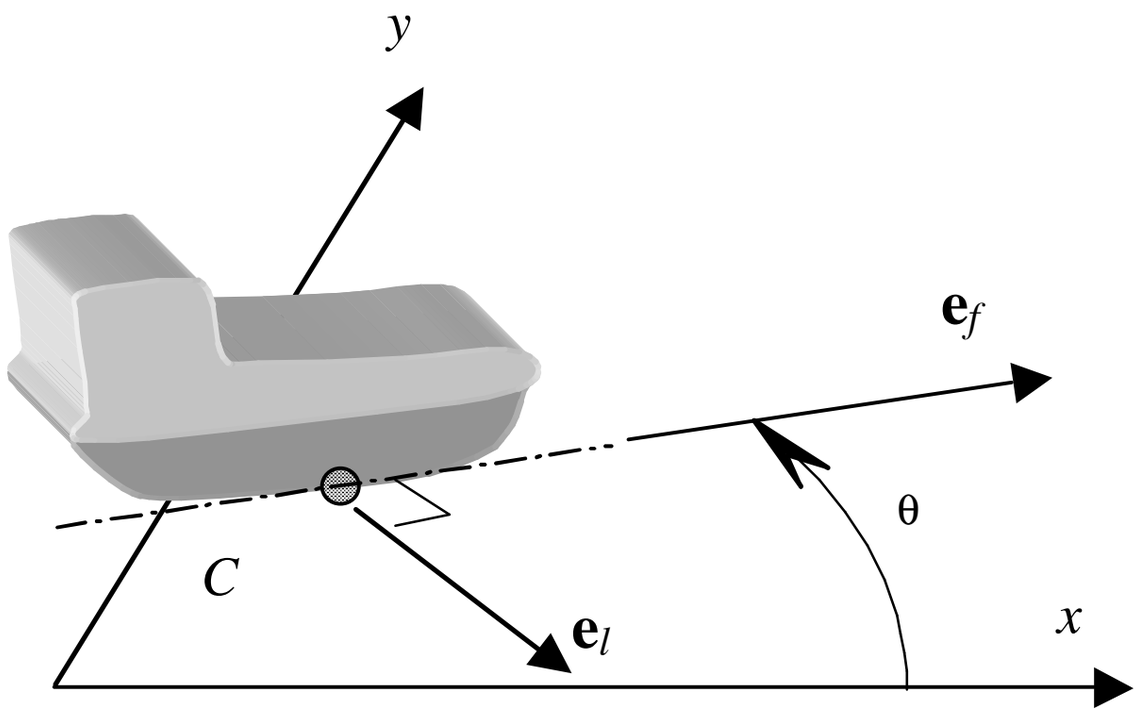
but

it is under-actuated, and thus the velocity is constrained



x

Idealized knife-edge (non-holonomic) constraint

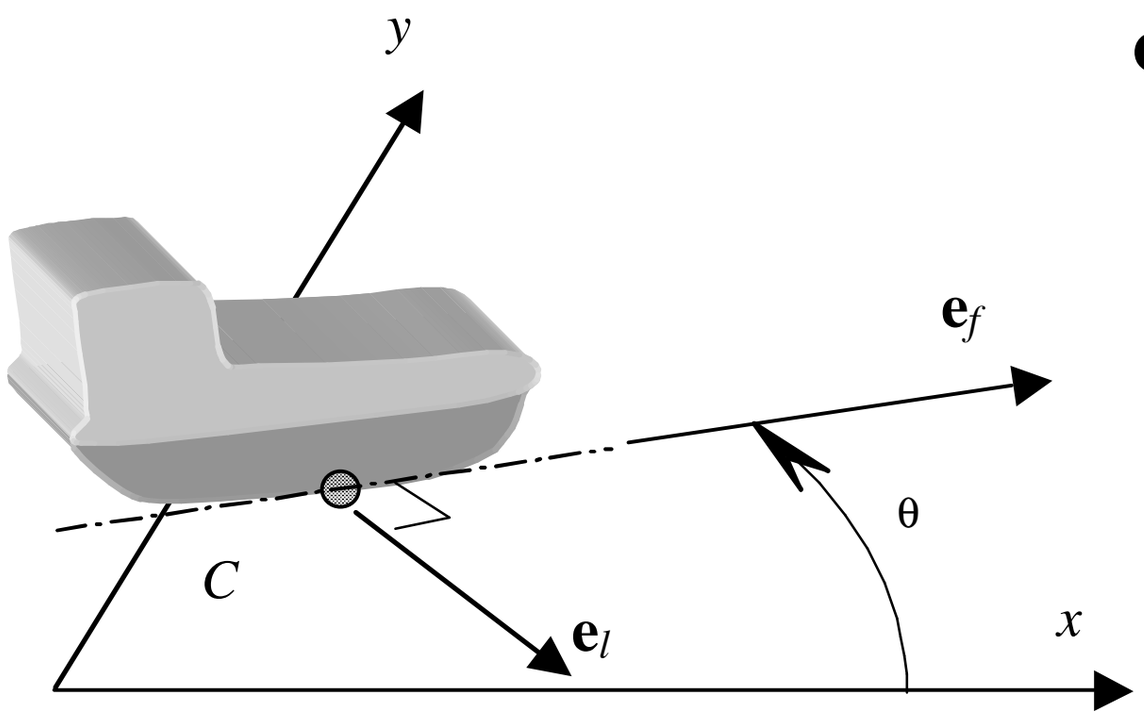


Single point of contact at C

Velocity constrained to be along the knife edge

$$v_l = 0$$

Idealized knife-edge (non-holonomic) constraint



$$\mathbf{e}_f = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \qquad \mathbf{e}_l = \begin{bmatrix} \sin\theta \\ -\cos\theta \end{bmatrix}$$

Velocity at point C :

$$\mathbf{v}_C = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

Single point of contact at C

$$\mathbf{v}_C = v\,\mathbf{e}_f = \begin{bmatrix} v\cos\theta \\ v\sin\theta \end{bmatrix}$$

Velocity constrained to be along the knife edge

$$v_l = 0$$

Constrained velocity :

$$\frac{\dot{y}}{\dot{x}} = \tan\theta$$

but the position is NOT constrained

Mobile Robot Drives

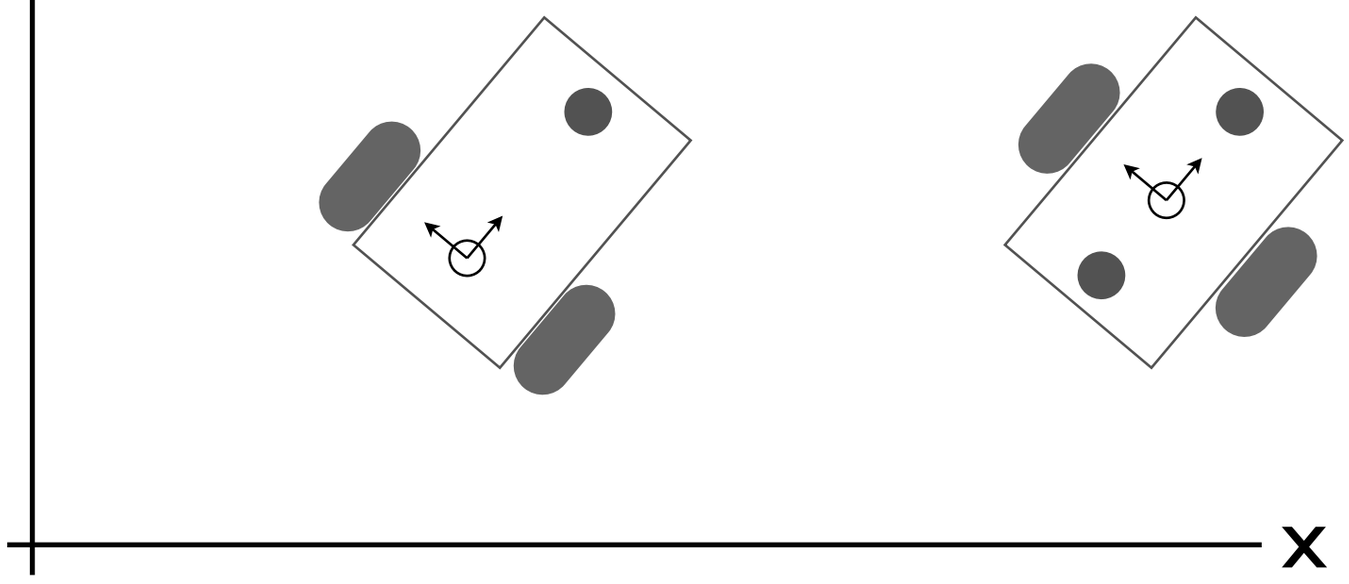
Differential steering

Co-axial wheels

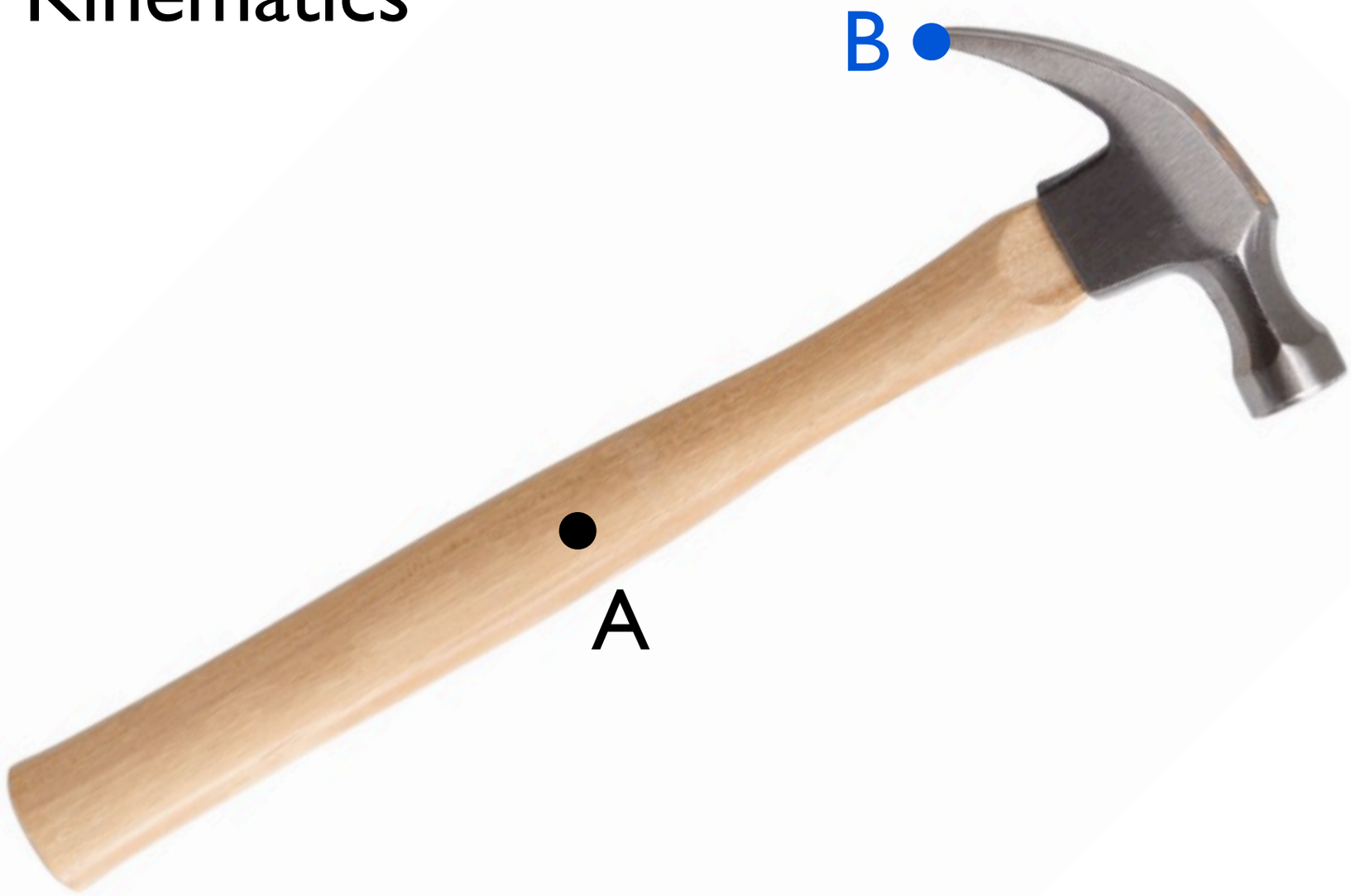
Independently driven

Two-dimensional

Non-holonomically constrained

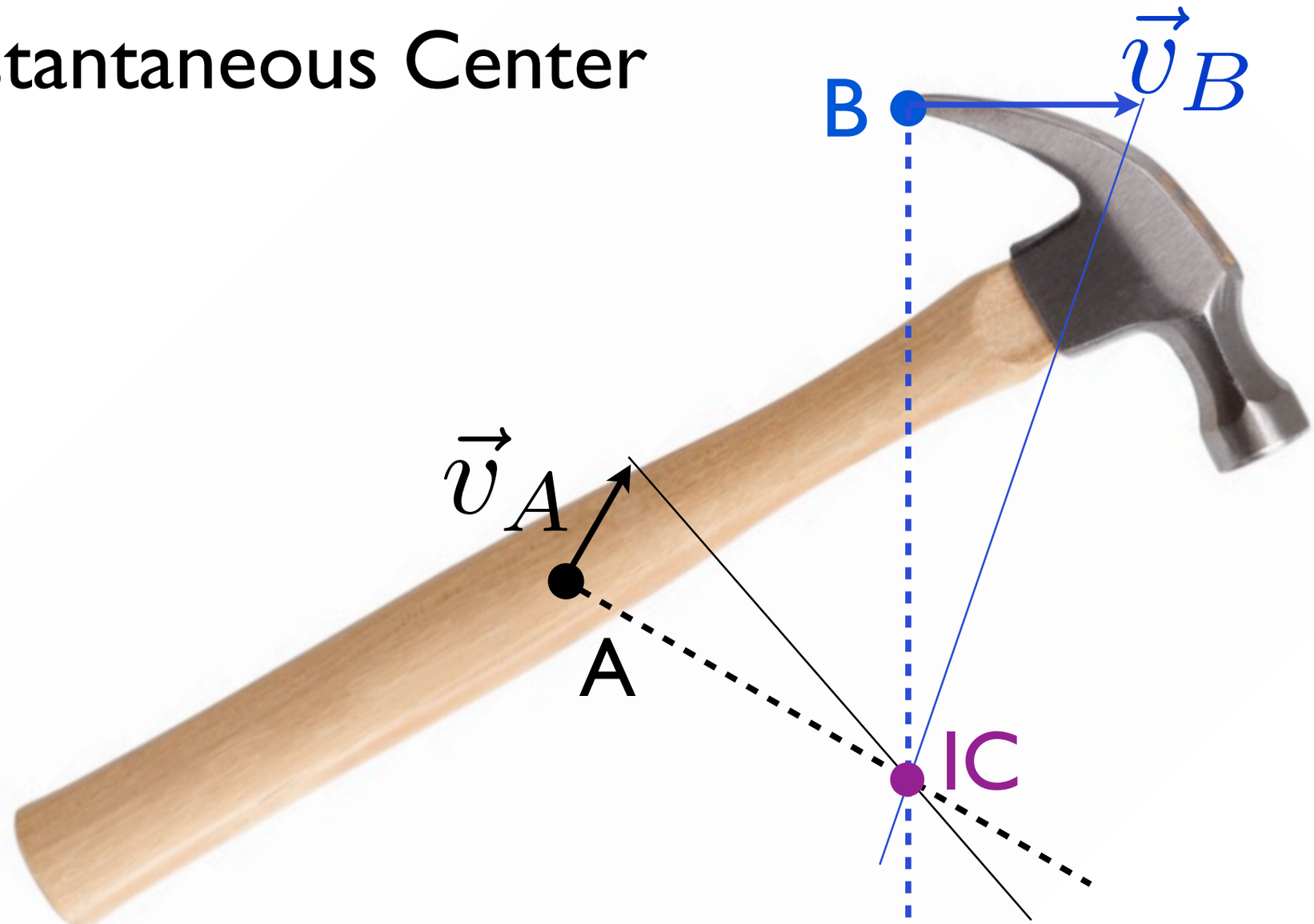


Rigid Body Kinematics



$$\vec{v}_B = \vec{v}_A + \vec{\omega} \times \vec{r}_{B/A}$$

Planar : Instantaneous Center



$$\vec{v}_B = \vec{v}_A + \vec{\omega} \times \vec{r}_{B/A}$$

Differential Steering : Forward Kinematics

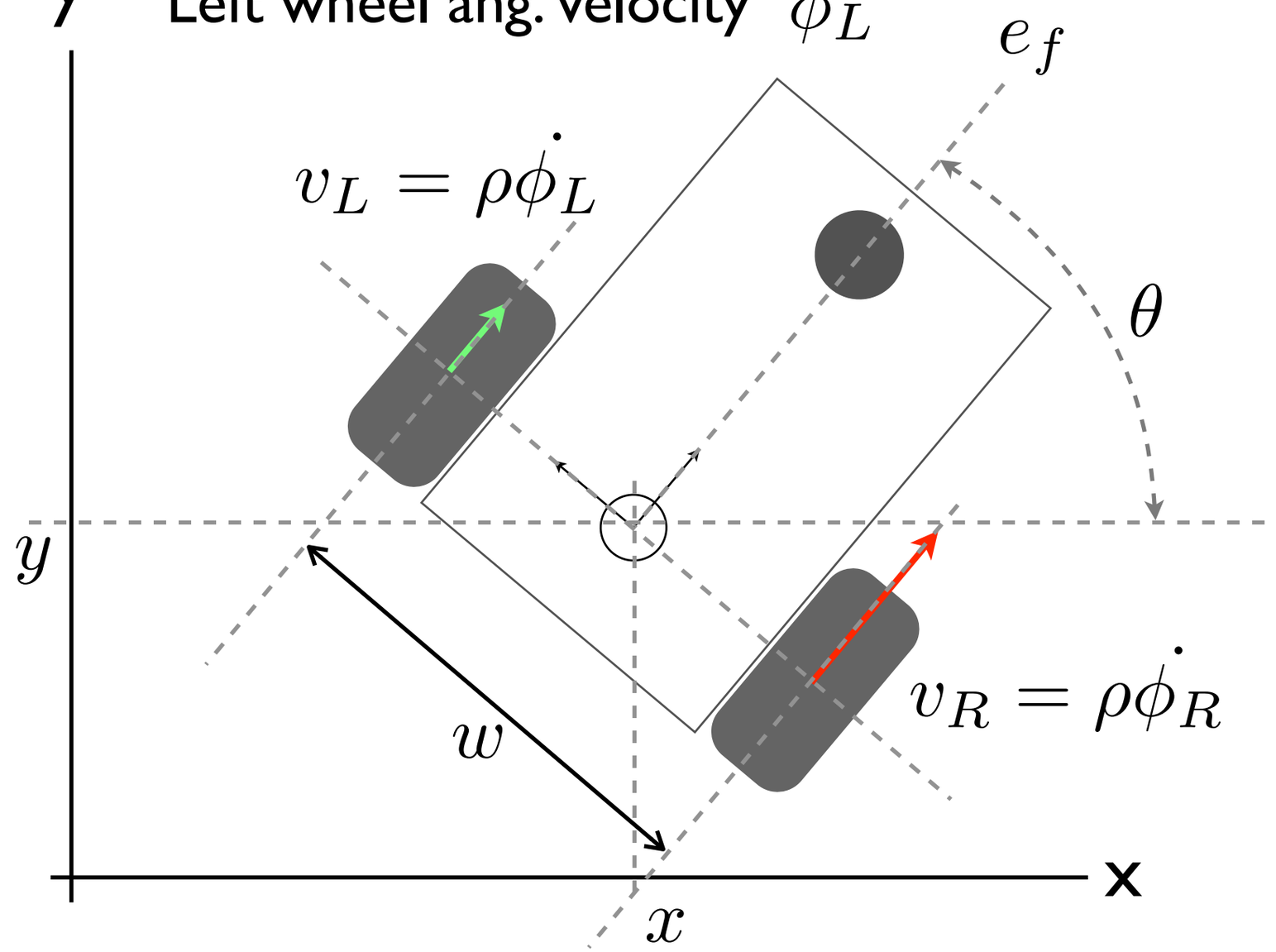
Given the robot geometry and wheel speeds, what is the robot's velocity?

Wheel radius ρ

Body width w

Right wheel ang. velocity $\dot{\phi}_R$

Left wheel ang. velocity $\dot{\phi}_L$



Forward velocity

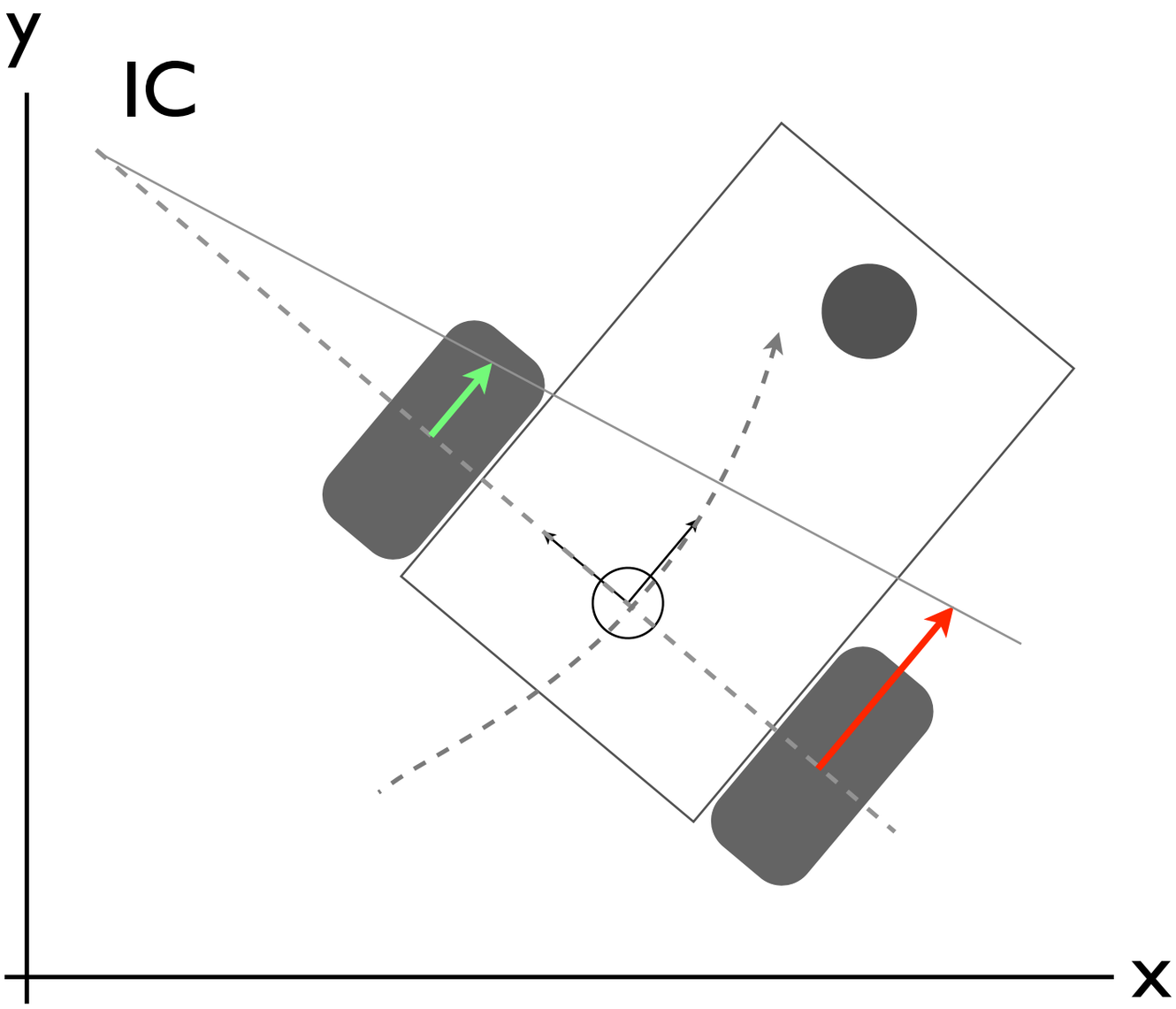
$$v_f = \frac{v_R + v_L}{2}$$

Angular velocity

$$\dot{\theta} = \frac{v_R - v_L}{w}$$

Differential Steering : Instantaneous Center of Curvature

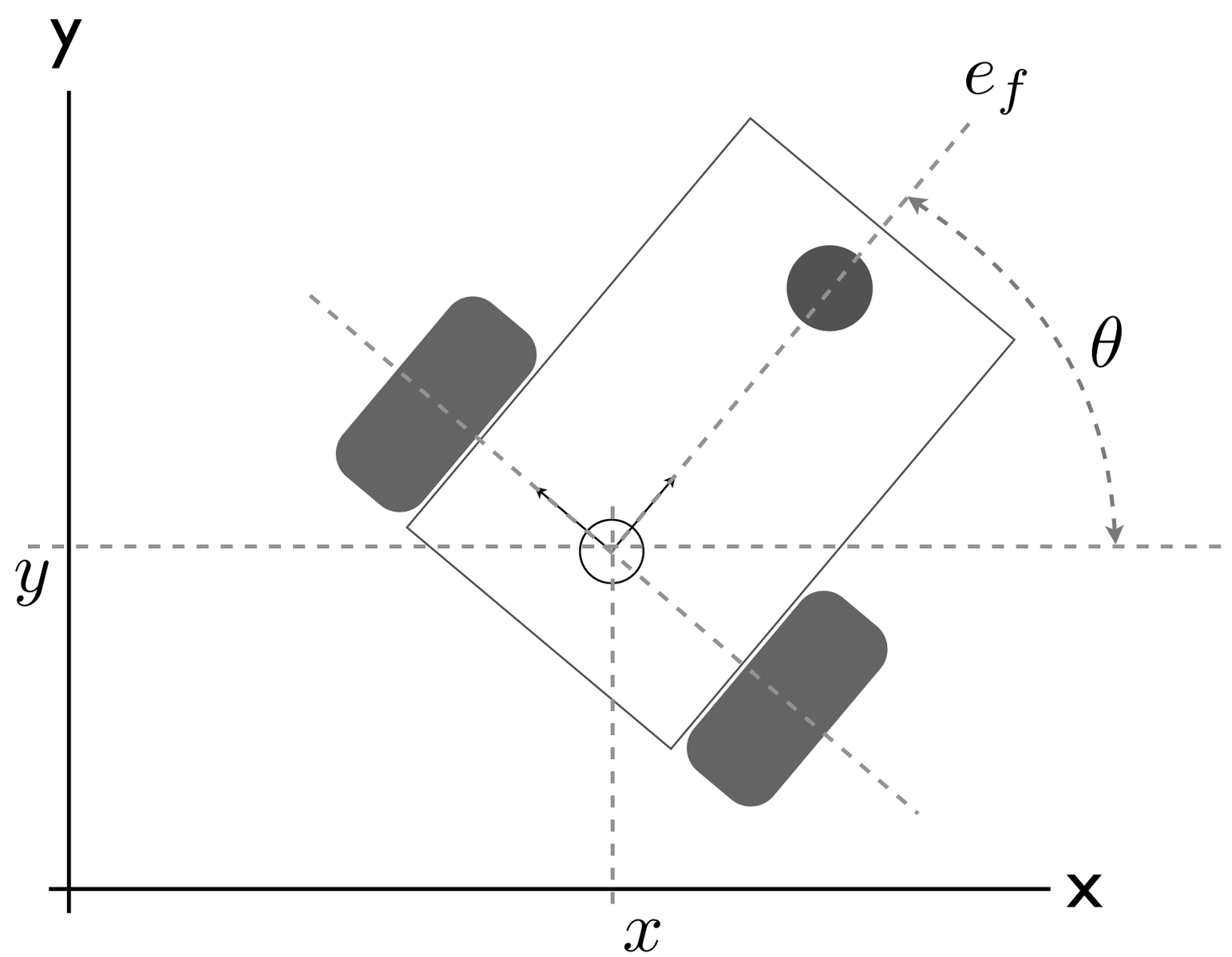
$$R = ?$$



$$R = \frac{w}{2}\frac{v_R + v_L}{v_R - v_L}$$

Differential Steering : Inverse Kinematics

What wheel speeds are necessary to produce a desired robot velocity?



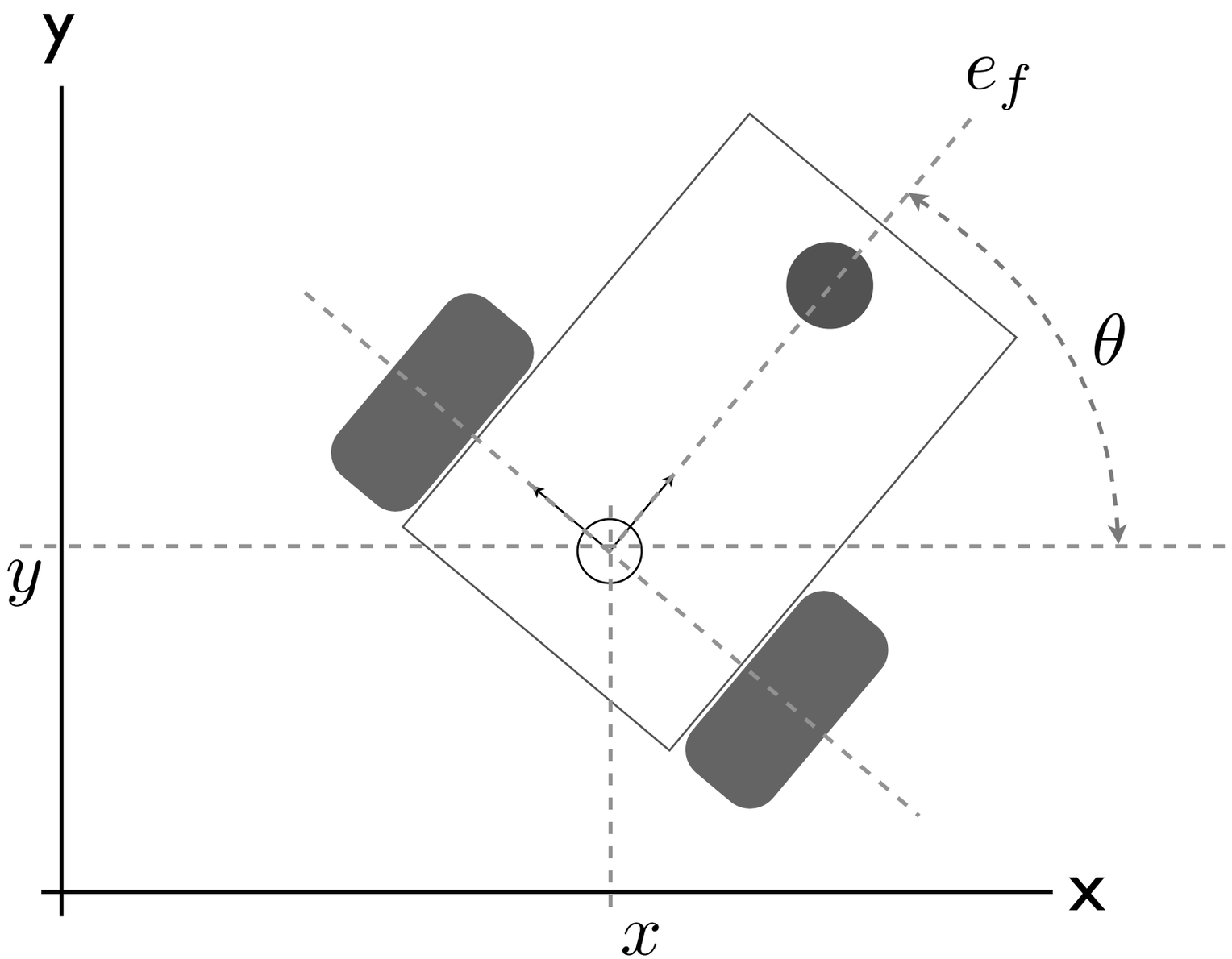
$$\dot{\phi}_L = \frac{1}{\rho}\left(v_f - \frac{w}{2}\dot{\theta}\right)$$

$$\dot{\phi}_R = \frac{1}{\rho}\left(v_f + \frac{w}{2}\dot{\theta}\right)$$

Differential Steering : Benefits

Simple construction

Zero minimum turning radius

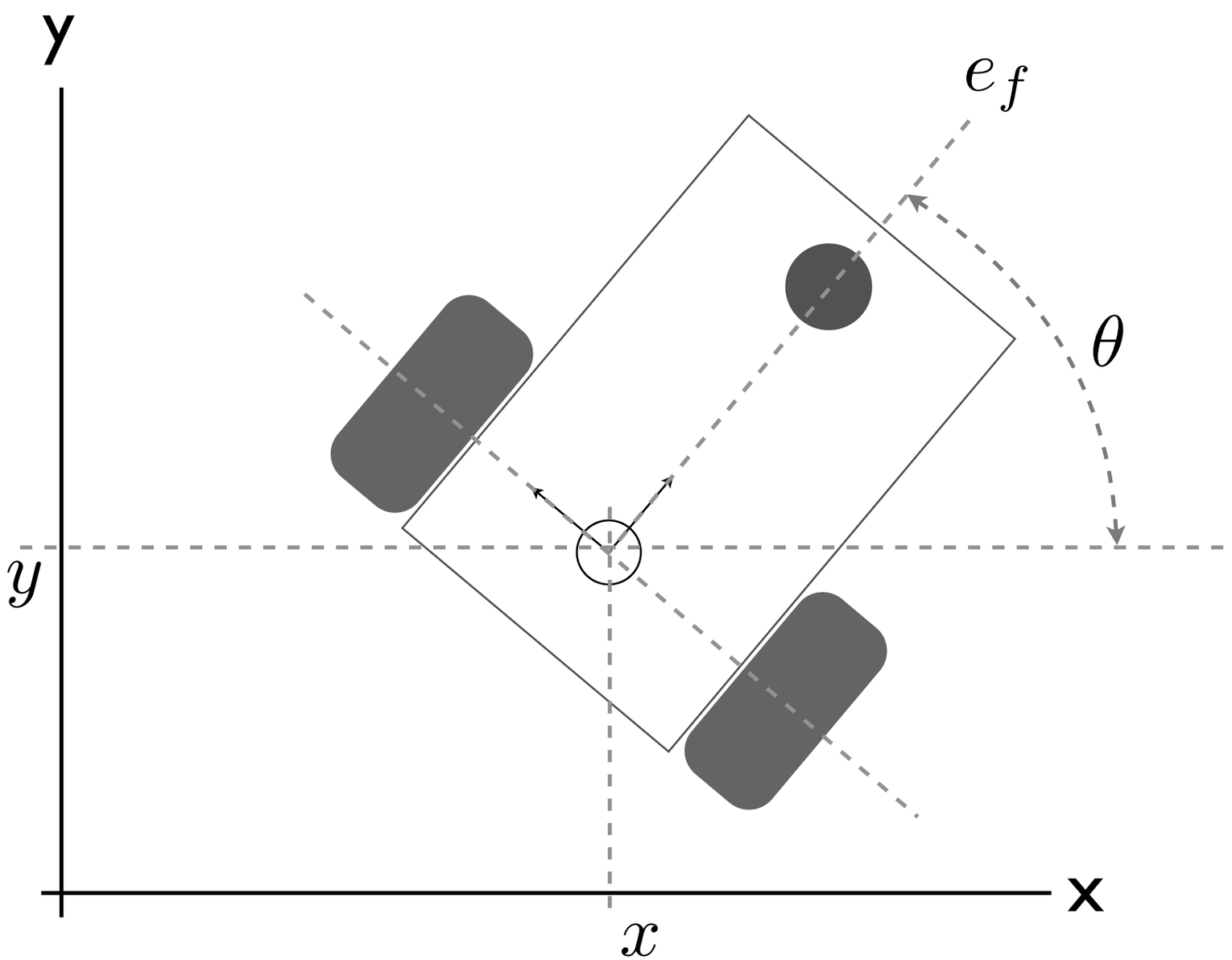


Differential Steering : Drawbacks

Small error in wheel speeds translates to large position errors

Requires two drive motors

Wheels-first is dynamically unstable



Tricycle : Forward Kinematics

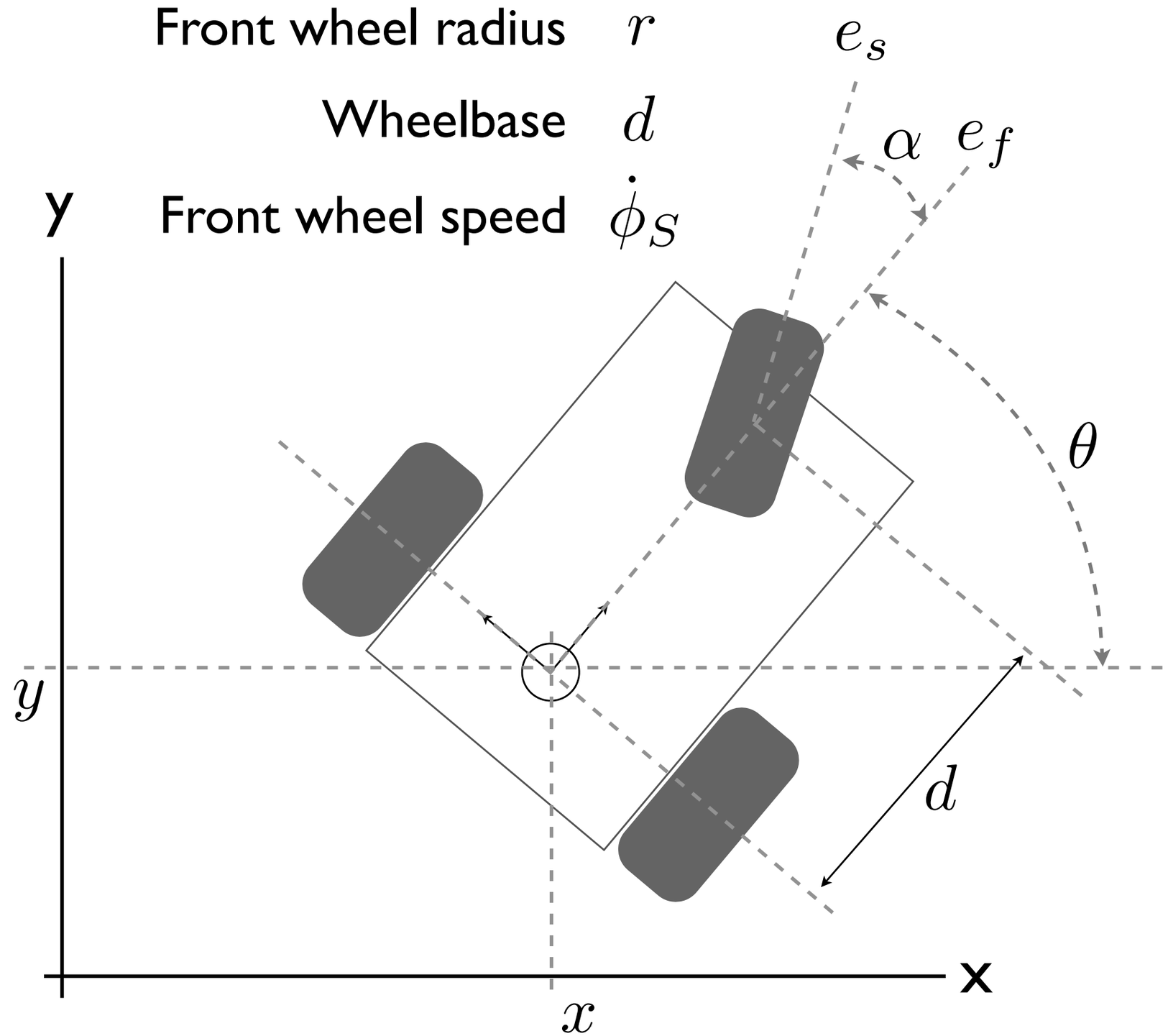
Steerable powered front wheel

Free-spinning rear wheels

Front wheel radius r

Wheelbase d

Front wheel speed $\dot{\phi}_S$



Forward velocity

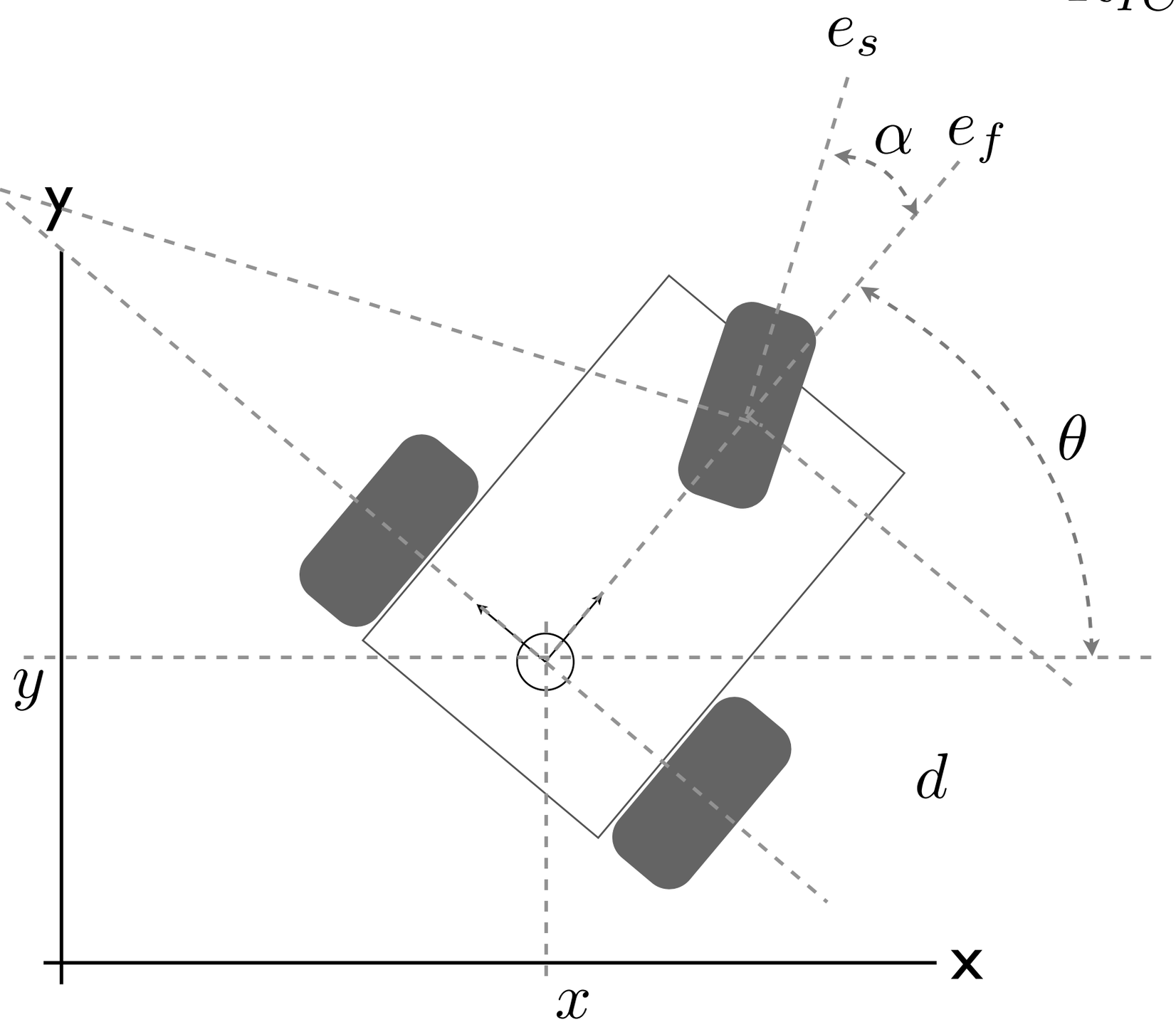
$$v_f = r\dot{\phi}_S \cos\alpha$$

Angular velocity

$$\dot{\theta} = \frac{r}{d}\dot{\phi}_S \sin\alpha$$

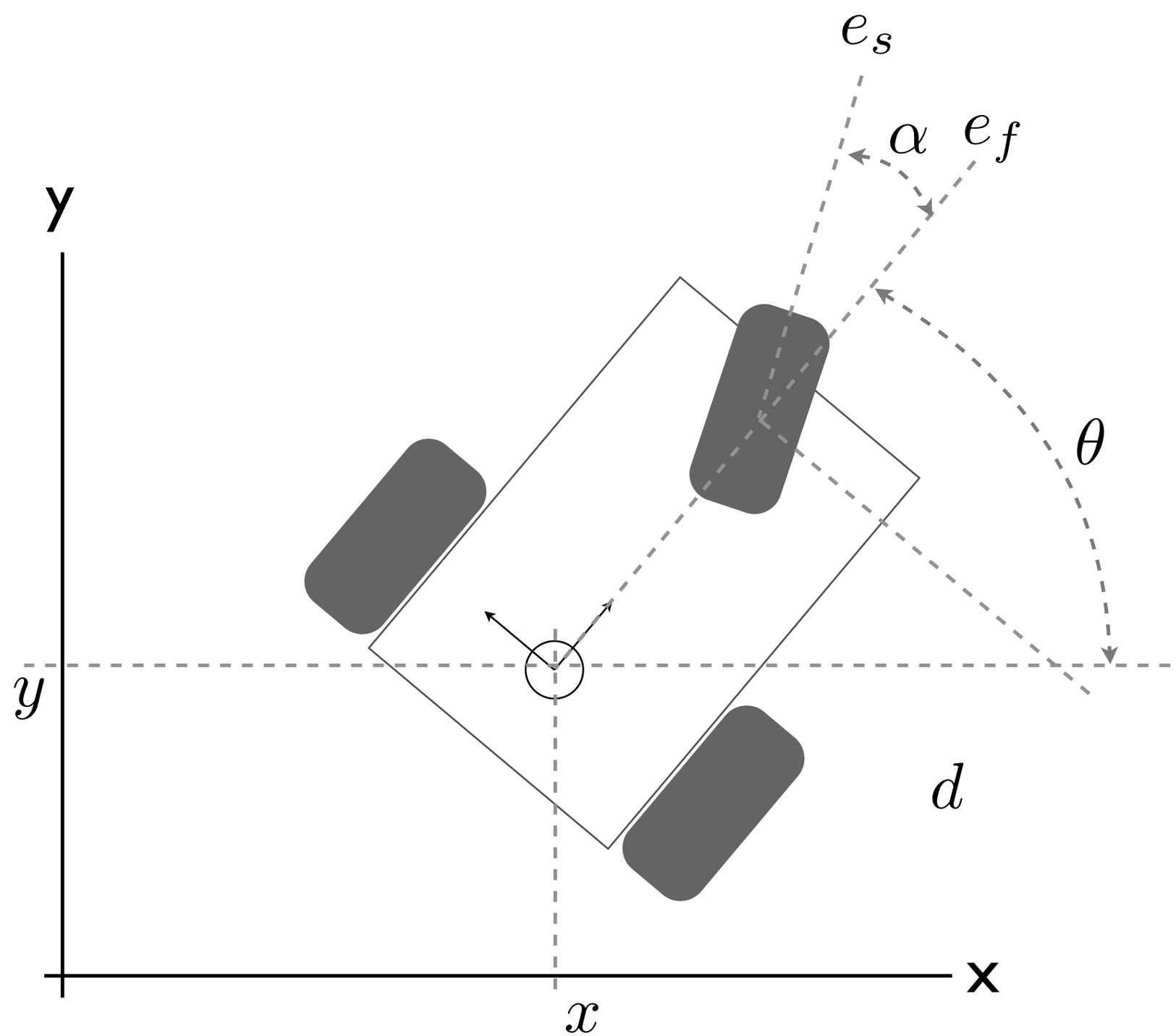
Tricycle : Instantaneous Center of Curvature

$$R_{ICC} = \frac{d}{\tan\alpha} = \frac{v_f}{\dot{\theta}}$$



Tricycle : Inverse Kinematics

What wheel speed and angle are necessary to produce a desired robot velocity?

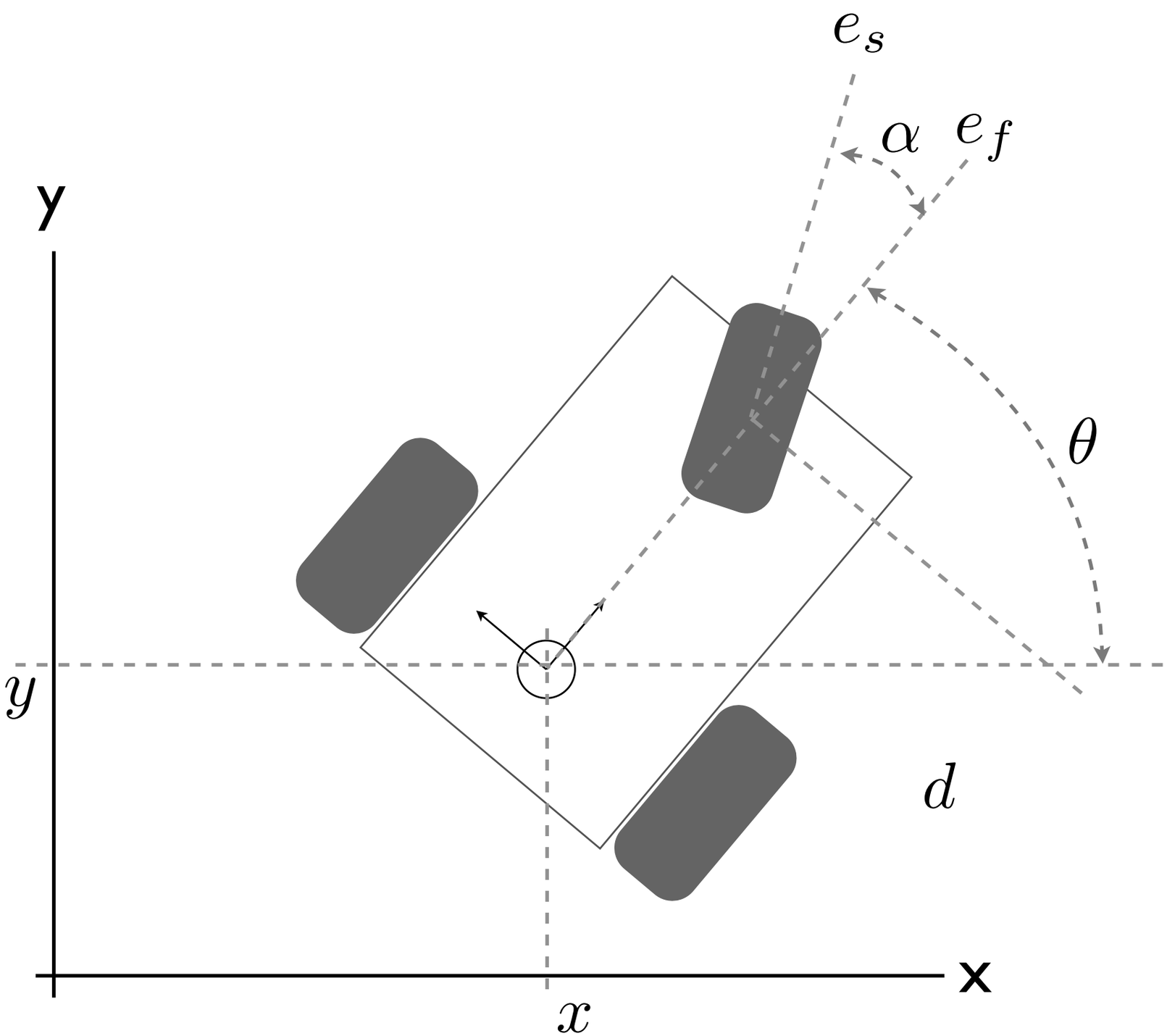


$$\alpha = \tan^{-1}\left(\frac{\dot{\theta}d}{v_f}\right)$$

$$\dot{\phi}_S = \frac{1}{r}\sqrt{v_f^2 + \dot{\theta}^2 d^2}$$

Tricycle : Benefits

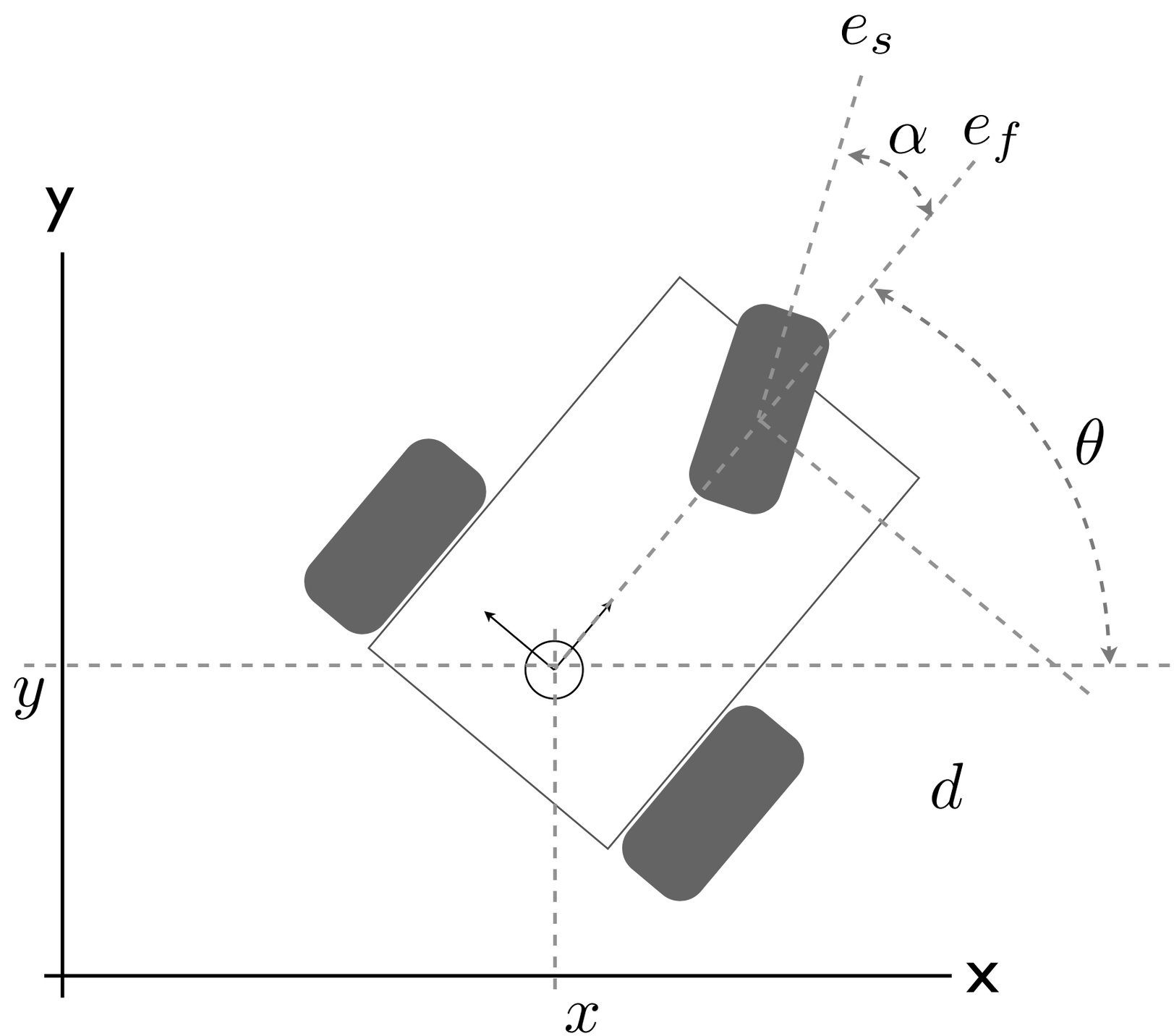
Doesn't require accurate speed matching



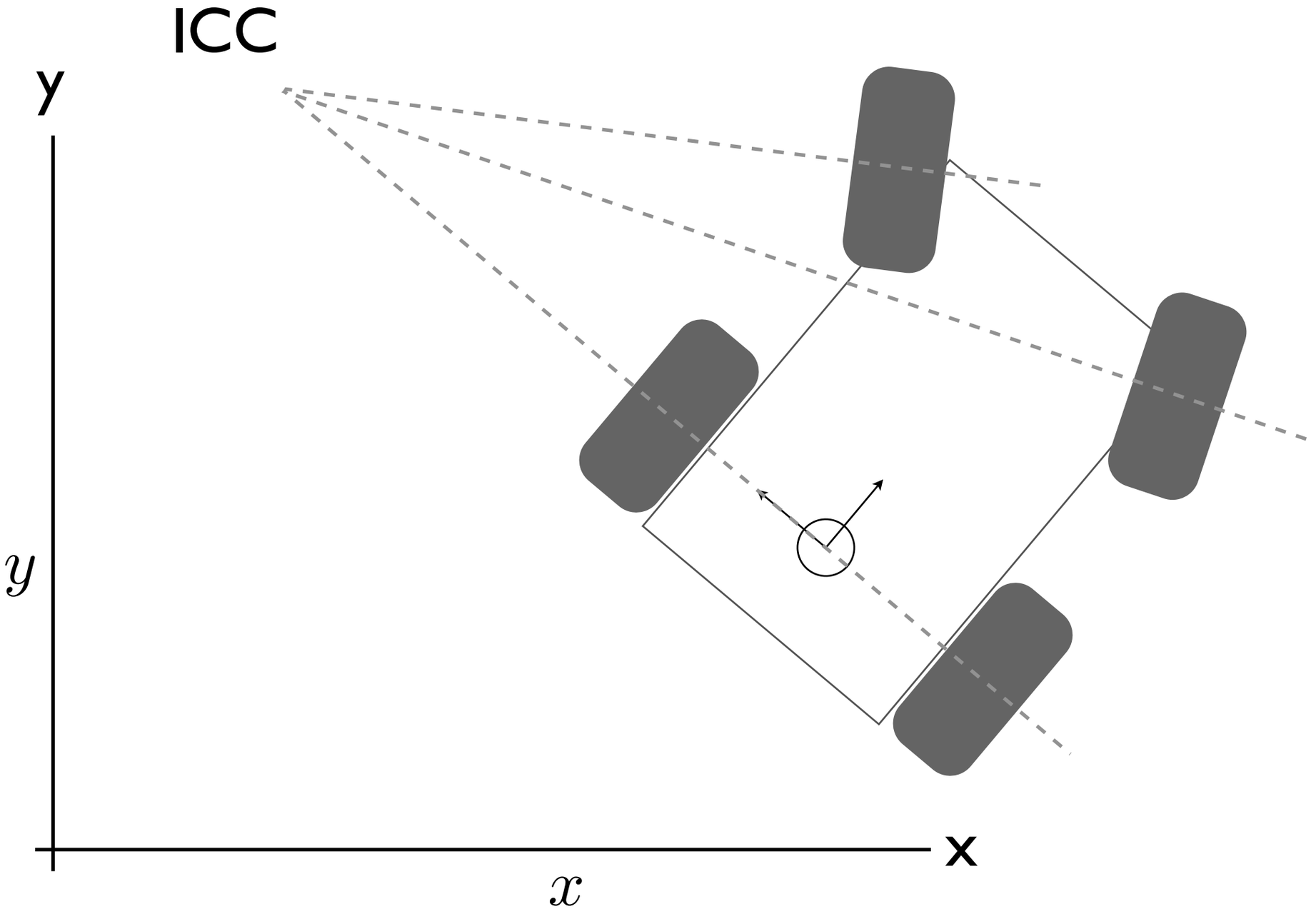
Tricycle : Drawbacks

Zero turning radius requires steering to 90 degrees

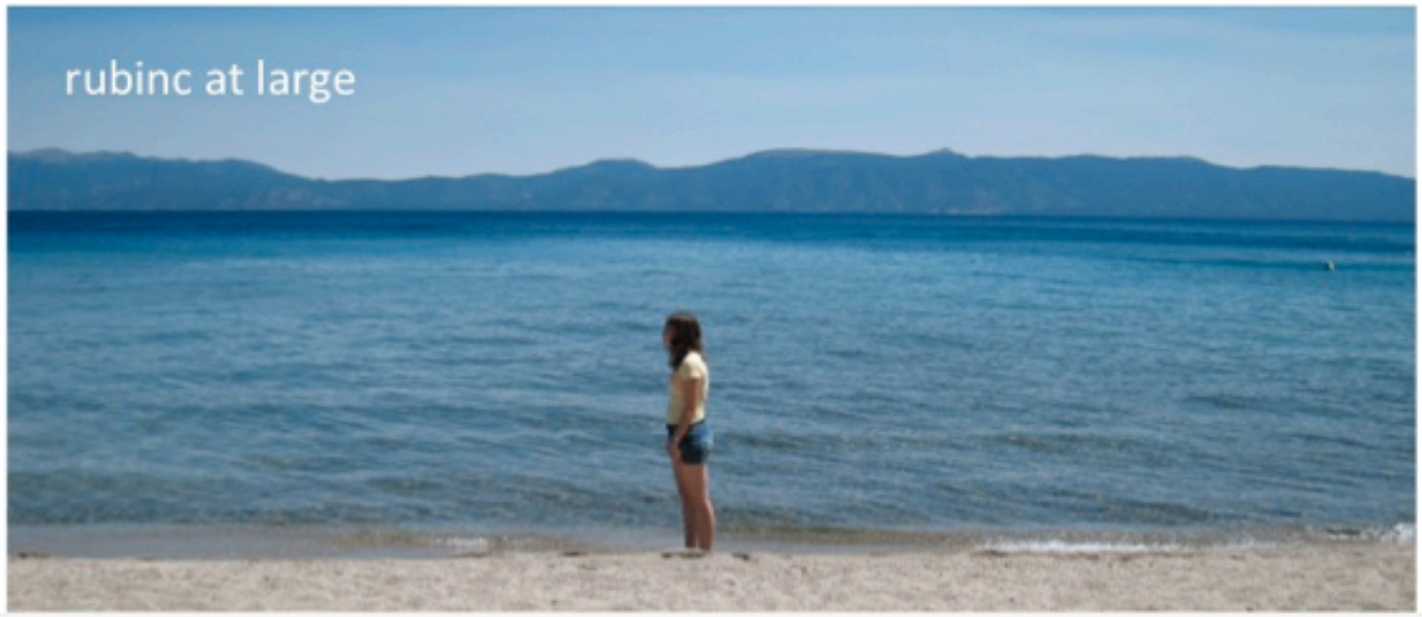
More complicated drive train



Ackerman Steering



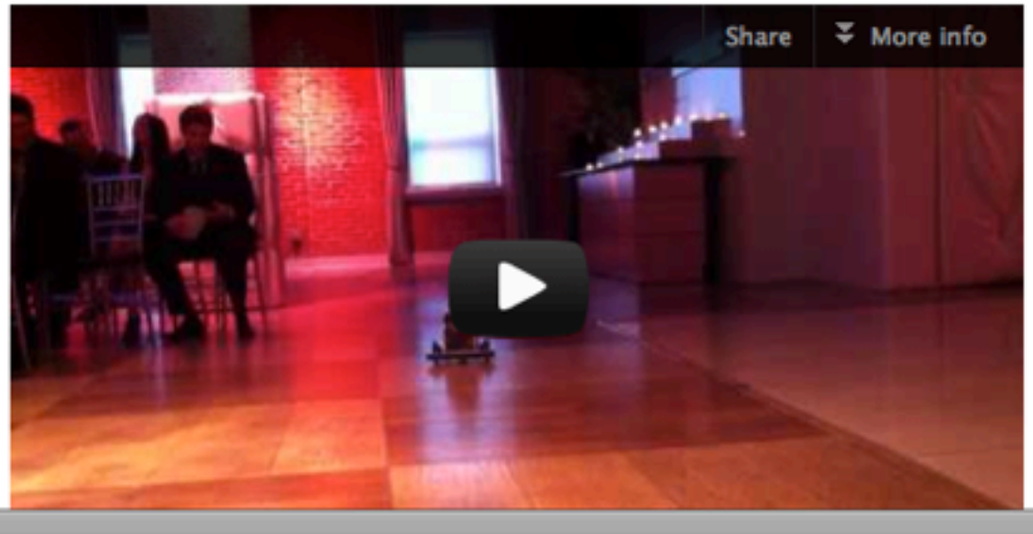
Questions ?



MONDAY, OCTOBER 15, 2012

We're married!

We won't get the fancy pictures back from the photographer for awhile, but in the meantime here's a video of our ring-bearer (thanks to Alana for taking the video, and Phil for teleoperating the ring-bearer):



PAST LIVES

- Ms. Rubin
- Queen of CTY

FOLLOWING

- dr. emsque
- the dinner takes it all
- martini
- la vida es una tombola
- colossus of rhode

BLOG ARCHIVE

▼ 2012 (42)
▶ November (1)
▼ October (7)
New camera
Like them apples

ROBOCKEY

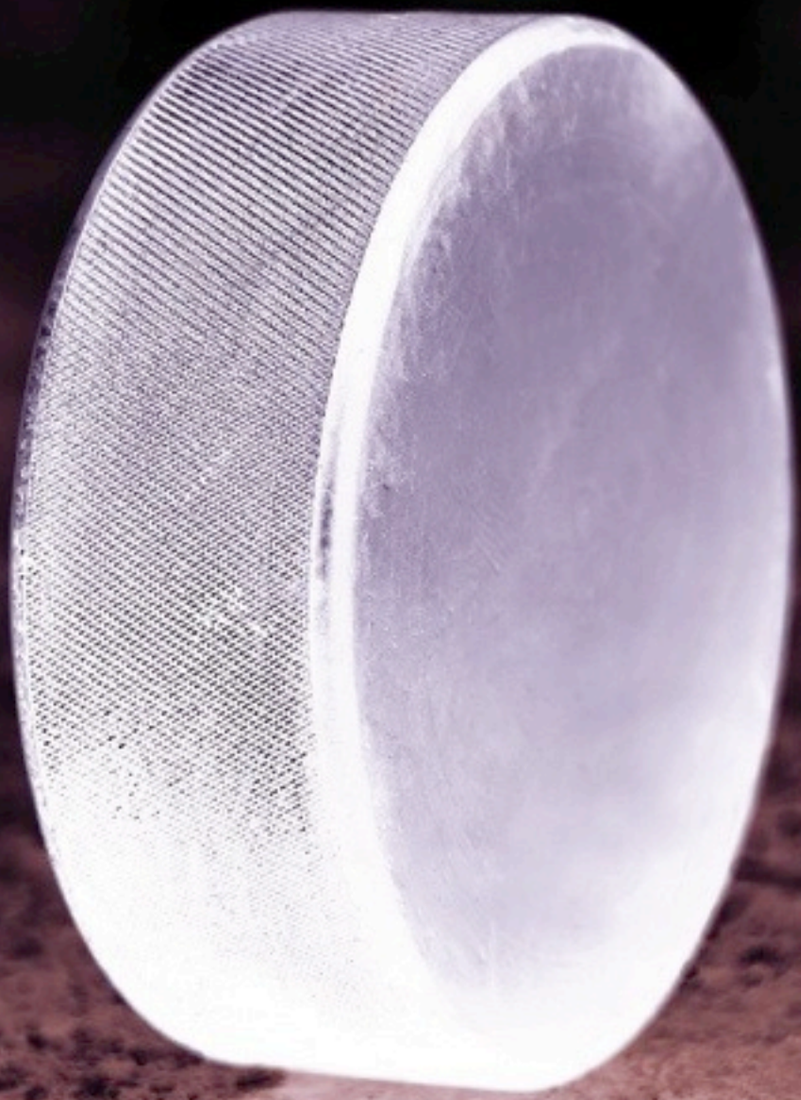
2012

WU + CHEN

6:00 P. M. DECEMBER 11

ROBOCKEY

2012



WU + CHEN

6:00 P. M. DECEMBER 11

Final Exam

Wednesday, December 19

Noon to 2:00 p.m.

Location to be announced

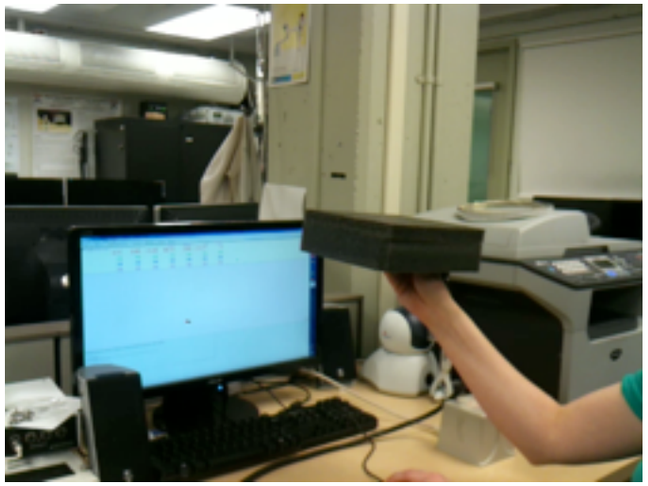
Comprehensive, covering everything through Tuesday's lecture

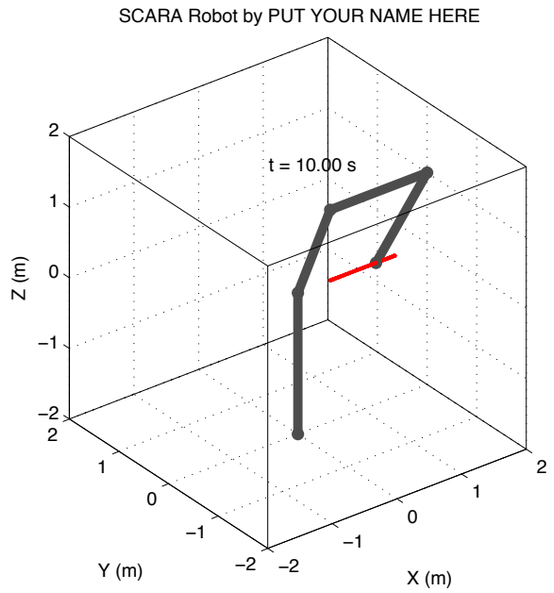
Closed book

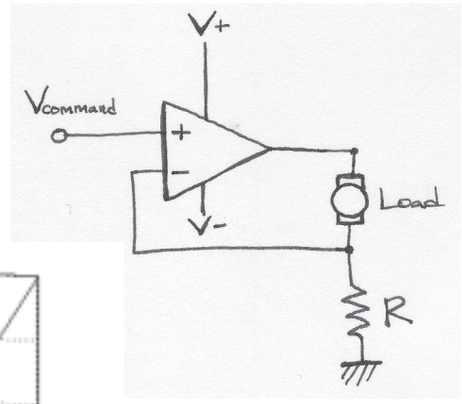
Four single-sided pages of notes

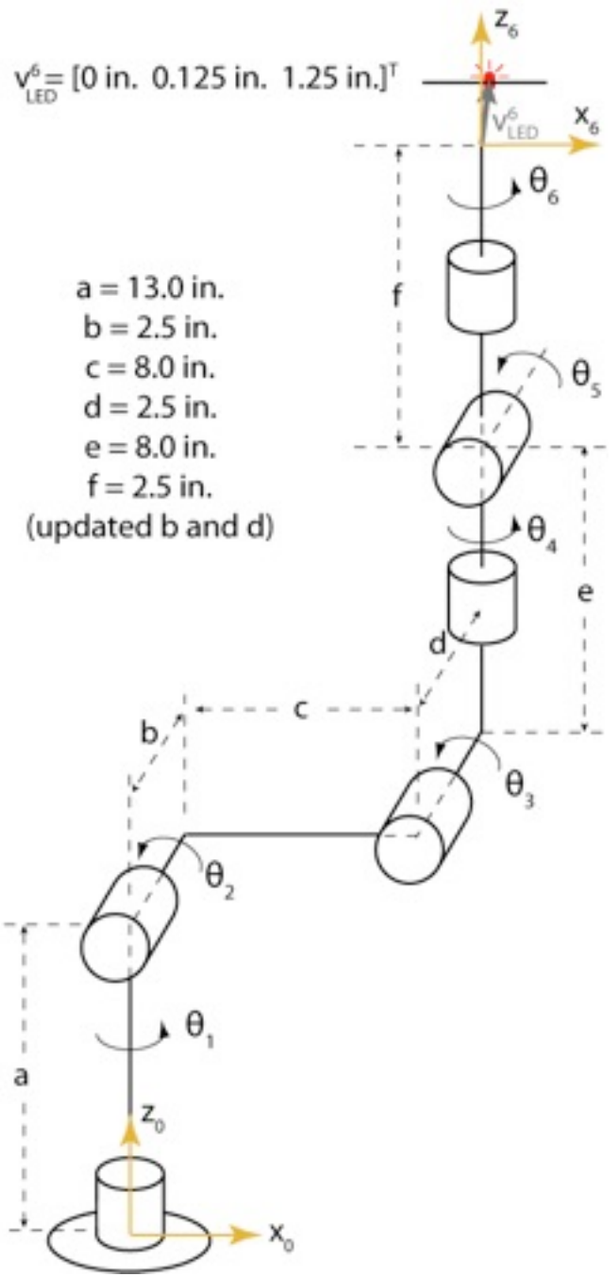
Calculator allowed

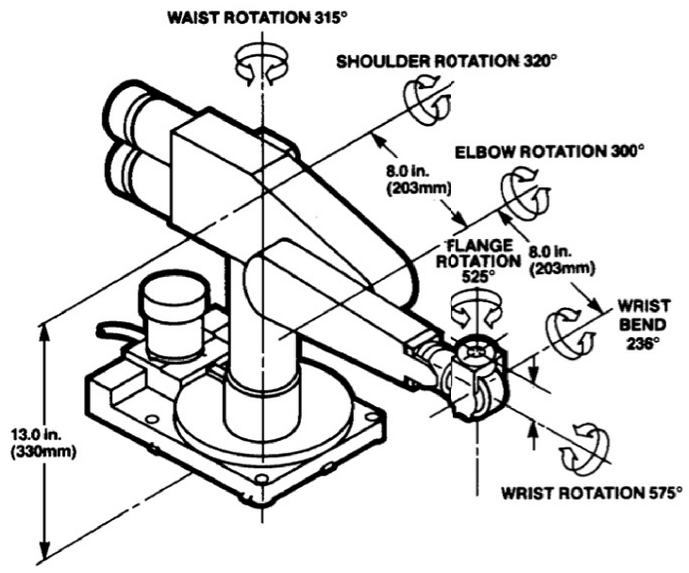

Philip A-G
Ryan H-Q
Denise R-Z

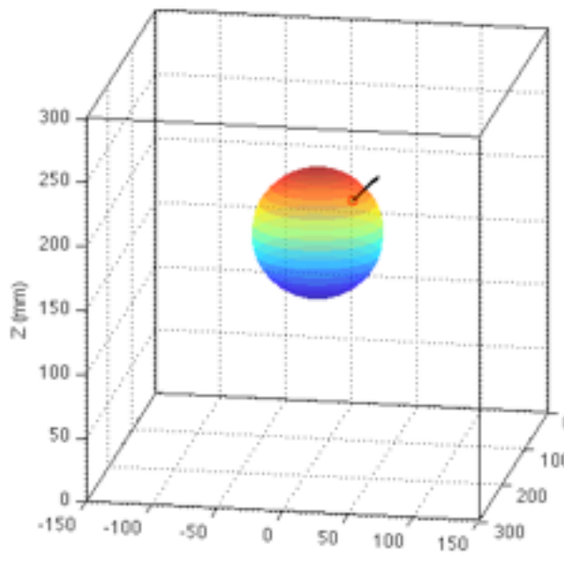


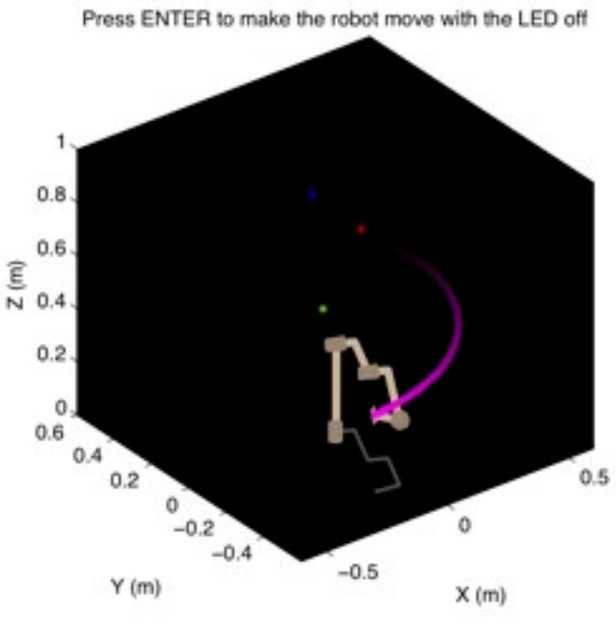

SCARA Robot by PUT YOUR NAME HERE
t = 10.00 s
Z (m)
Y (m)
X (m)

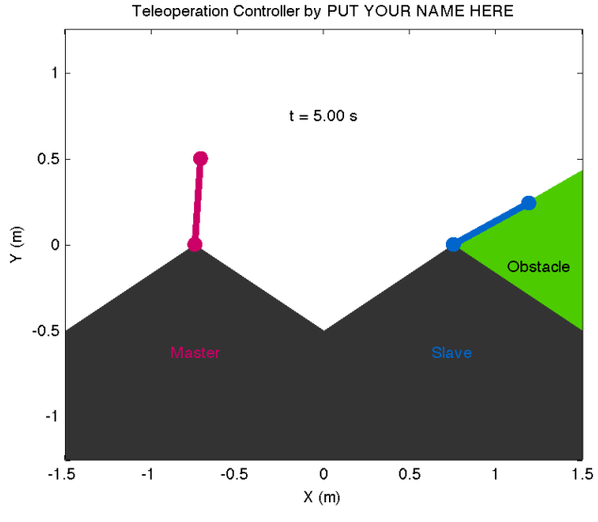

V+
Vcommand
V-
Load
R


V^6_{LED} = [0 in. 0.125 in. 1.25 in.]^T
a = 13.0 in.
b = 2.5 in.
c = 8.0 in.
d = 2.5 in.
e = 8.0 in.
f = 2.5 in.
(updated b and d)


WAIST ROTATION 315°
SHOULDER ROTATION 320°
ELBOW ROTATION 300°
8.0 in. (203mm)
FLANGE ROTATION 525°
8.0 in. (203mm)
WRIST BEND 236°
13.0 in. (330mm)
WRIST ROTATION 575°


Z (mm)
X (mm)


Press ENTER to make the robot move with the LED off
Z (m)
Y (m)
X (m)


Teleoperation Controller by PUT YOUR NAME HERE
t = 5.00 s
Y (m)
X (m)
Obstacle
Master
Slave

Thank you for a great semester!