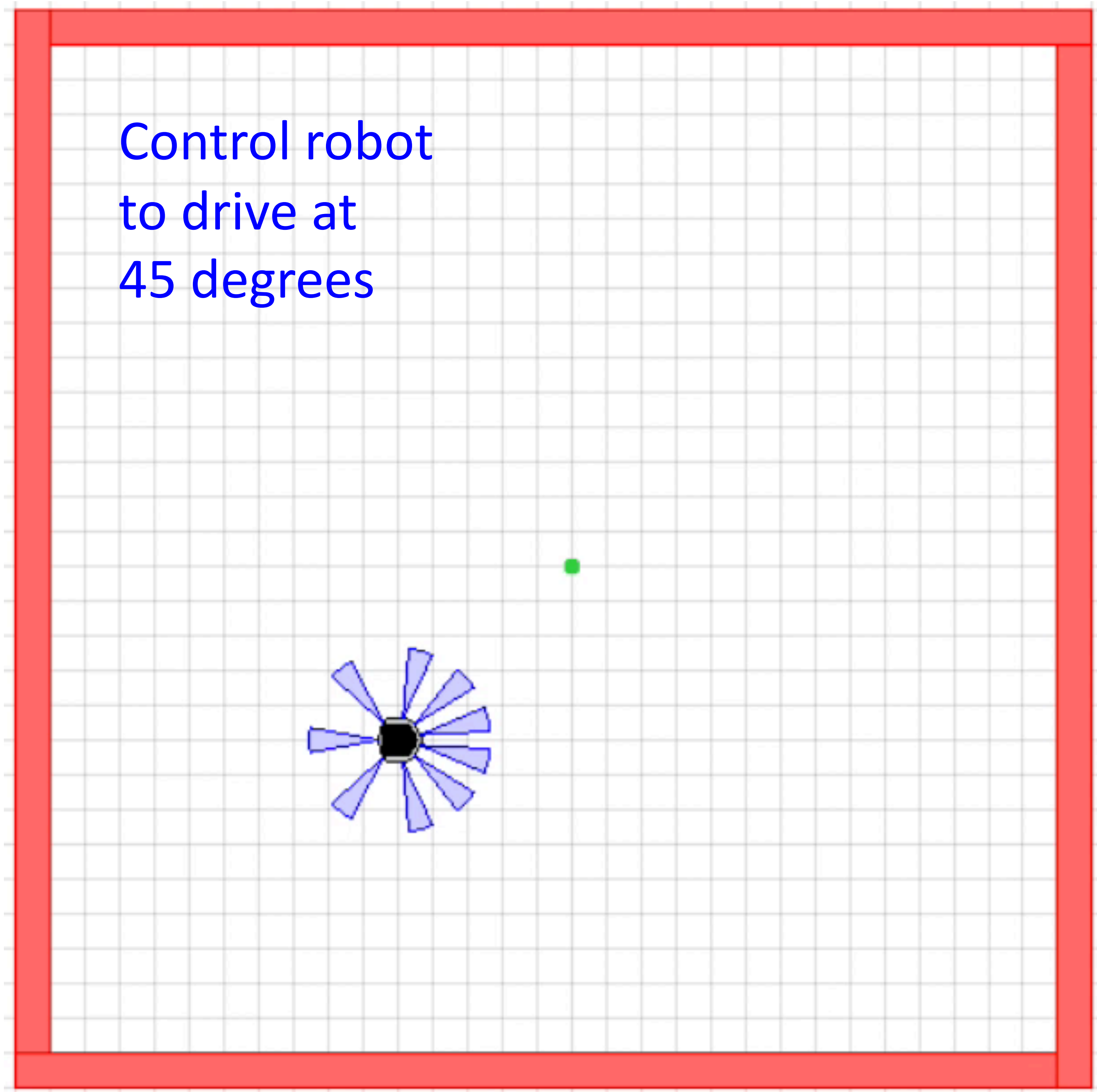
電気情報工学基礎演習B

Simulation of Controlling Mobile Robot
Lecture 2

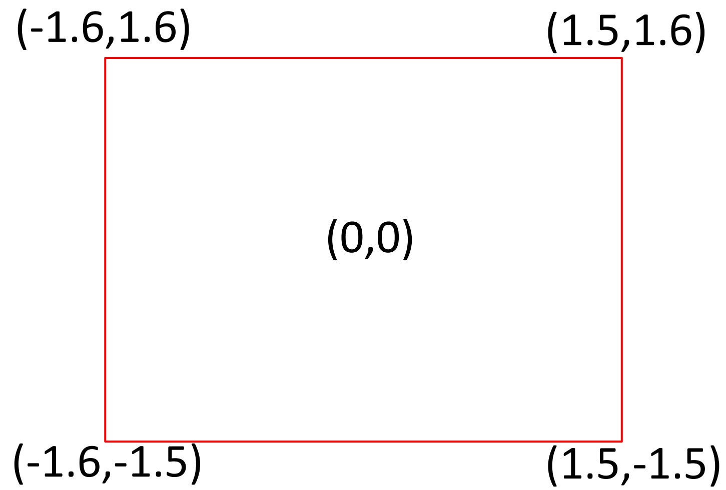
Instructor:　蔡　凱
TA:　笠原　萌人


Control robot
to drive at
45 degrees

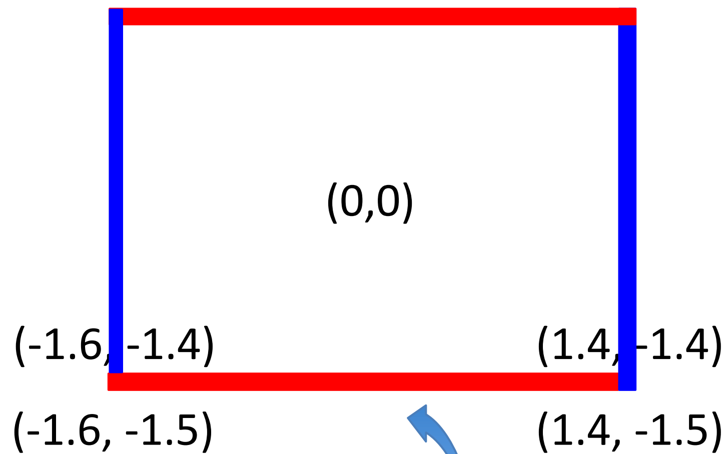
Lecture 2

- Simulator configuration
- Robot dynamics
- GoToAngle control

Simulator Configuration



Simulator Configuration



- settings.xml:
  - <obstacle>

    <pose x=”-1.6" y=”-1.5" theta="0" />

    <geometry>

    <point x=“0" y="0" />

    <point x=”3" y=”0" />

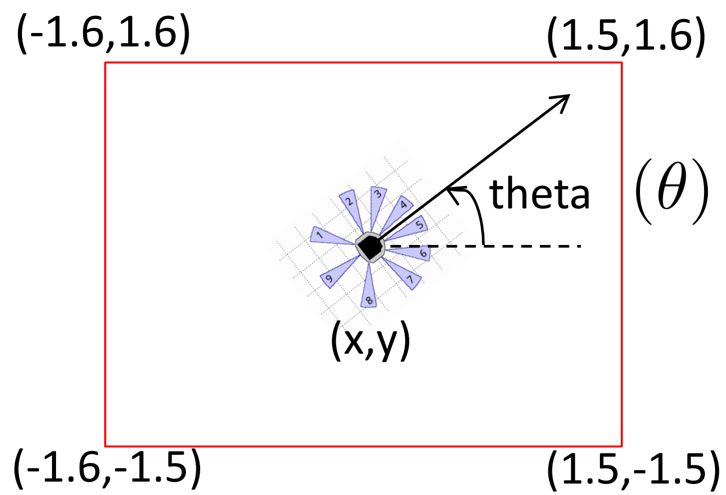
    <point x=”3" y=”0.1" />

    <point x=“0" y=“0.1" />

    </geometry>

    </obstacle>

Robot Location and Orientation



- settings.xml:
  - <robot type="Khepera3">

    <supervisor type="khepera3.K3Supervisor" />

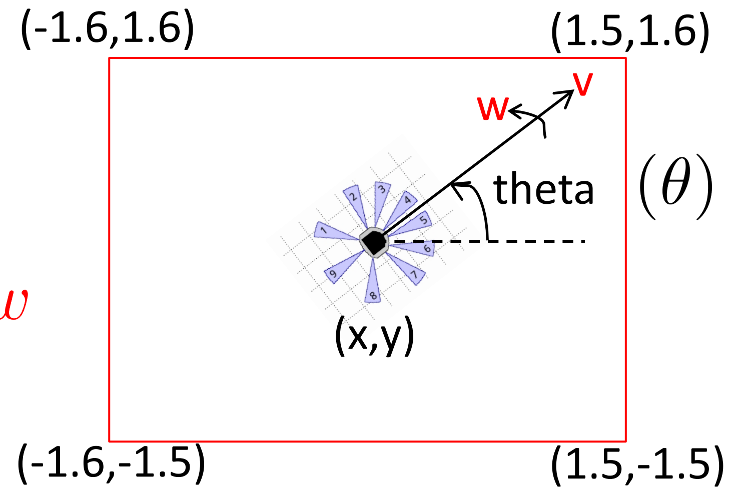
    <pose x=“0" y=“0" theta=“0.785" />

    </robot>

Robot Dynamics

linear velocity: v

angular velocity: w



(θ)

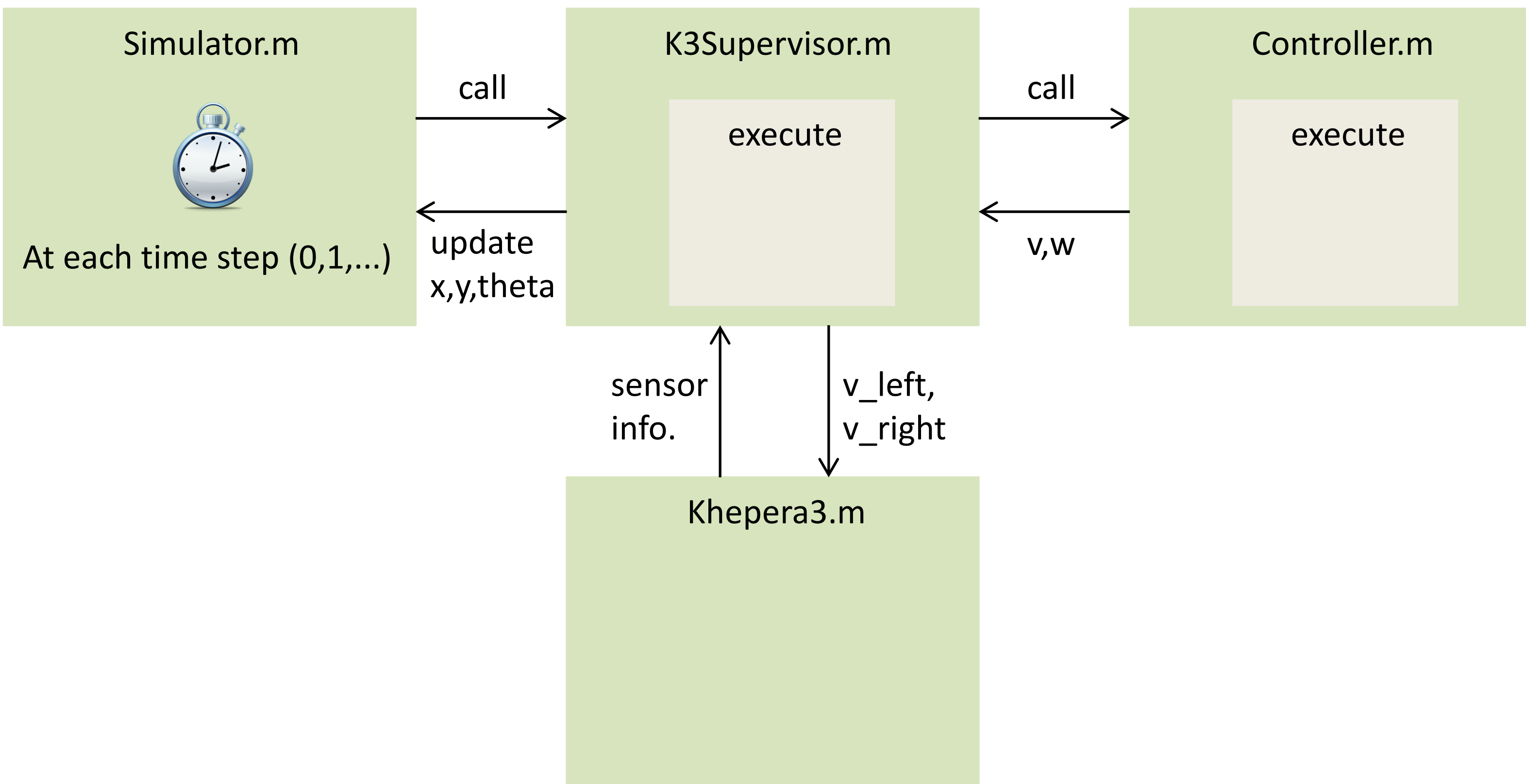
$$\frac{dx}{dt} = v\cos(\theta)$$

$$\frac{dy}{dt} = v\sin(\theta)$$

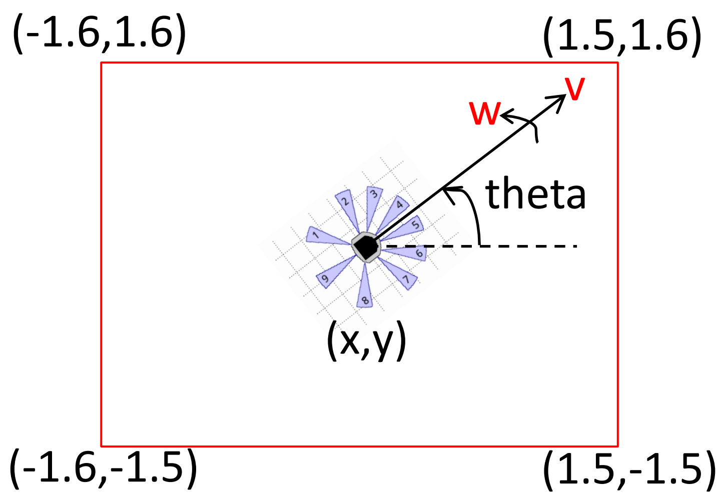
$$\frac{d\theta}{dt} = w$$

- In this course, we set v=constant and only control w.
- Our robot has two-wheel differential drive:
  [v_left   v_right] = f(v,w)

How Simulation Works



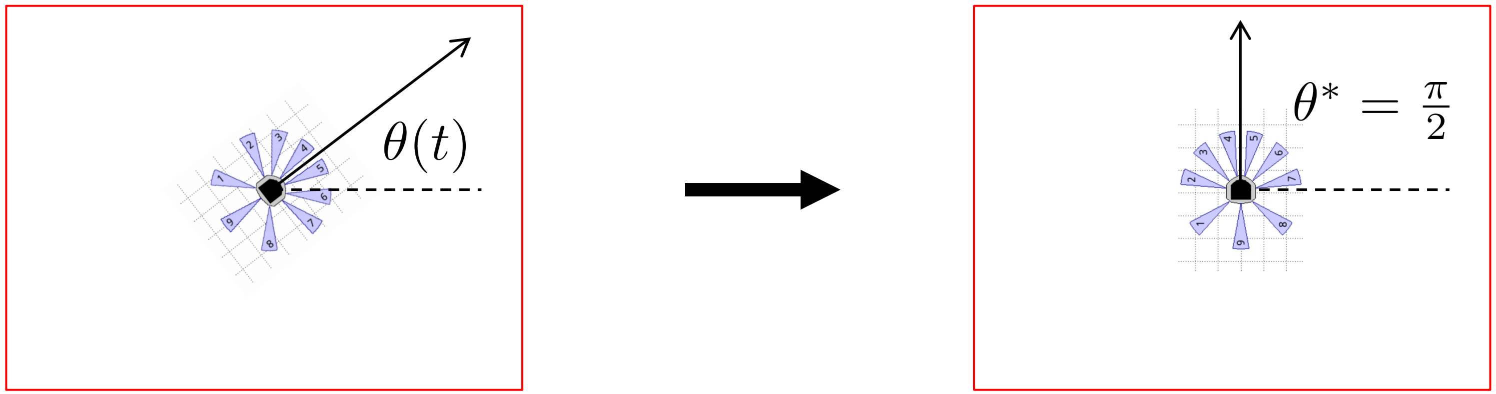
Robot Dynamics



- Initial robot pose: x(0), y(0), theta(0)   [in settings.xml]
- At each time t=0,1,2…, robot pose: x(t), y(t), theta(t)
- +simiam/+controller/+khepera3/K3Supervisor.m
  - function execute(obj, dt)
    - fprintf('current_pose: (%0.3f, %0.3f, %0.3f)\n)', x, y, theta);

Go To Angle

- Objective: steer the robot to a specified angle



Assume robot is moving at linear velocity v=constant.

We only control robot's angular velocity w: $\frac{d\theta}{dt} = u(t)$, control input

Objective: $\theta(t) \to \theta^*$ as $t \to \infty$ $\quad (e(t) := \theta^* - \theta(t) \to 0)$

GoToAngle Controller

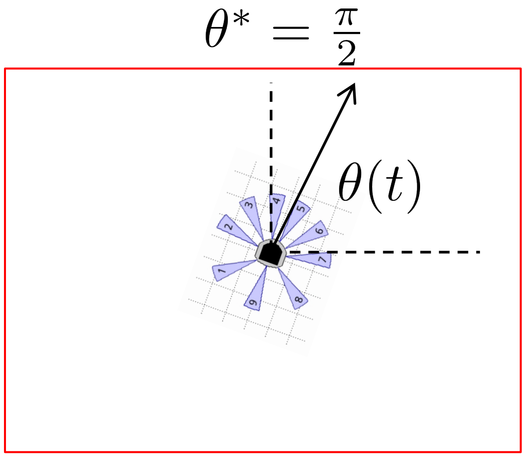
- Objective: steer the robot to a specified angle

$$\frac{d\theta}{dt} = u(t), \text{ control input}$$

$$\text{Objective: } \theta(t) \to \theta^* \text{ as } t \to \infty \qquad (e(t) := \theta^* - \theta(t) \to 0)$$

<u>P(roportional)-controller:</u> $u(t) = K_p e(t)$

– if e(t)>0, u(t)>0 $\quad (K_p > 0)$



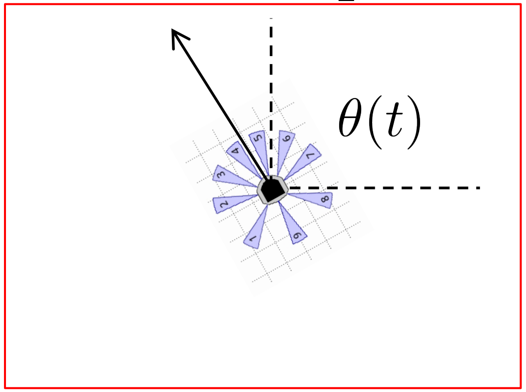
GoToAngle Controller

- Objective: steer the robot to a specified angle

$$\frac{d\theta}{dt} = u(t), \text{ control input}$$

$$\text{Objective: } \theta(t) \to \theta^* \text{ as } t \to \infty \qquad (e(t) := \theta^* - \theta(t) \to 0)$$

$\theta^* = \frac{\pi}{2}$



P(roportional)-controller: $u(t) = K_p e(t)$

$(K_p > 0)$

- if e(t)>0, u(t)>0
- if e(t)<0, u(t)<0

GoToAngle Controller

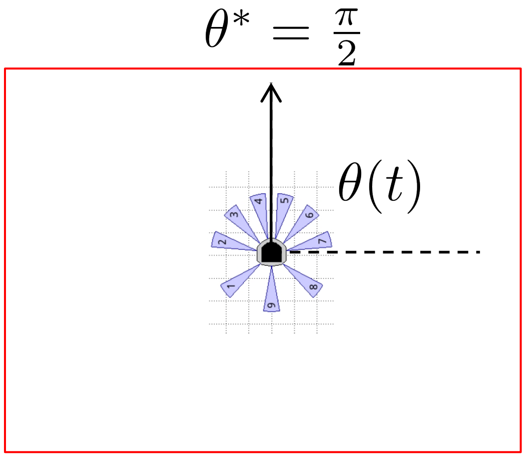
- Objective: steer the robot to a specified angle

$$\frac{d\theta}{dt} = u(t), \text{ control input}$$

$$\text{Objective: } \theta(t) \to \theta^* \text{ as } t \to \infty \qquad (e(t) := \theta^* - \theta(t) \to 0)$$

<u>P(roportional)-controller:</u> $u(t) = K_p e(t)$ $\quad (K_p > 0)$

- if e(t)>0, u(t)>0
- if e(t)<0, u(t)<0
- if e(t)=0, u(t)=0
- big/small e(t) yields big/small u(t)



Code

- +simiam/+controller/+khepera3/K3Supervisor.m
  - function obj=K3Supervisor()

```
% Input your code below %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Specified angle
obj.theta_d = 0; (change this to see what happens)
%Specified (constant) speed
obj.v = 0; (change this to see what happens)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Code

- +simiam/+controller/GoToAngle.m
  - function obj=GoToAngle()

```
% Input your code below %
%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Propositional control gain %
obj.Kp = 0; (change this to see what happens)
%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Exercises

- Use package: simiam_lecture2.zip
- Change robot's pose in settings.xml
- Set robot's linear speed and desired angle in K3Supervisor.m
- Adjust control gain parameter in GoToAngle.m

Tasks

Task 1

- Set robot's pose (-1,-1,-1.57) in settings.xml
- Set robot's linear speed 0.1 and desired angle 0.785 in K3Supervisor.m
- Find the largest control gain parameter in GoToAngle.m that works "smoothly"

Task 2

- Set robot's pose (-1,-1,-1.57) in settings.xml
- Set robot's desired angle 0.785 in K3Supervisor.m and control gain parameter 10 in GoToAngle.m
- Find the largest linear speed in K3Supervisor.m that works