

Sistemas Autônomos

através do Robot Operating System (ROS)

Prof. André Schneider de Oliveira
Prof. João Alberto Fabro

Universidade Tecnológica Federal do Paraná (UTFPR)

ROS-Matlab

1. Interface ROS-Matlab

- **inicializar** – rosinit
- **finalizar** - roshutdown
- **ver nós ativos** - rosnode list

1. Tópicos

- **listar tópicos ativos** - rostopic list
- **Informações sobre tópicos** - rostopic info /turtle1/pose

2. Mensagens

- **Listar tipos de mensagens** - rosmg list
- **Informações sobre mensagens** - rosmg show geometry_msgs/Twist

Matlab Publisher

%cria a mensagem

```
twist = rosmessagem('geometry_msgs/Twist')
```

%cria o publisher

```
pub = rospublisher('/turtle1/cmd_vel')
```

%visualiza a mensagem

```
showdetails(twist)
```

%atribui valores para a mensagem

```
twist.Linear.X = 0.5;
```

%visualiza a mensagem

```
showdetails(twist)
```

%envia a mensagem

```
send(pub,twist)
```

Matlab Subscriber

%cria o subscriber

```
sub = rossubscriber('/turtle1/pose')
```

%recebe a mensagem

```
posedata = receive(sub, 10)
```

%ve os detalhes da mensagem

```
showdetails(posedata)
```

%captura os campos da mensagem

```
posedata.X
```

```
posedata.Y
```

TurtleSim

Ver os tópicos ativos

>> rostopic list

Topicos do TurtleSim

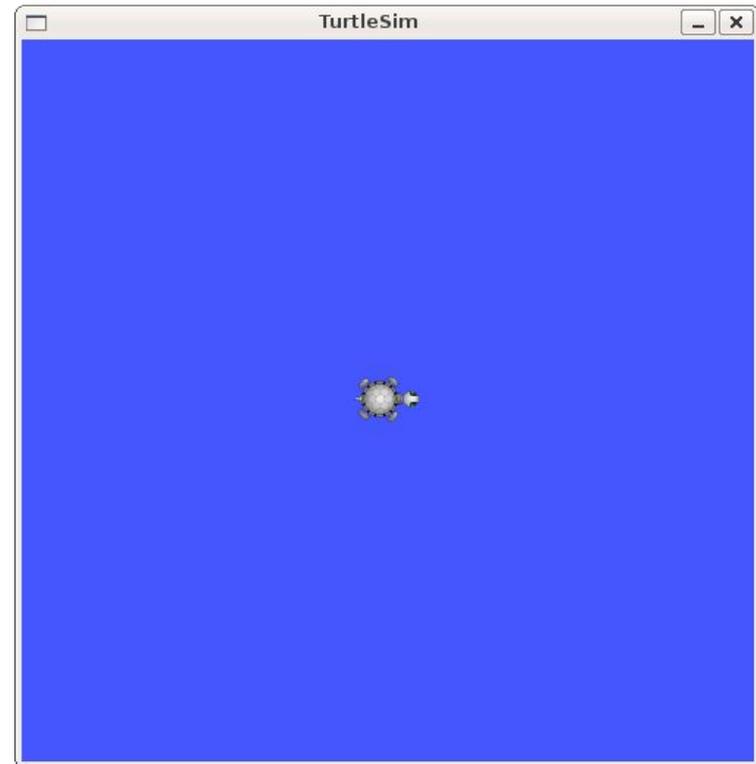
/rosout

/rosout_agg

/turtle1/cmd_vel

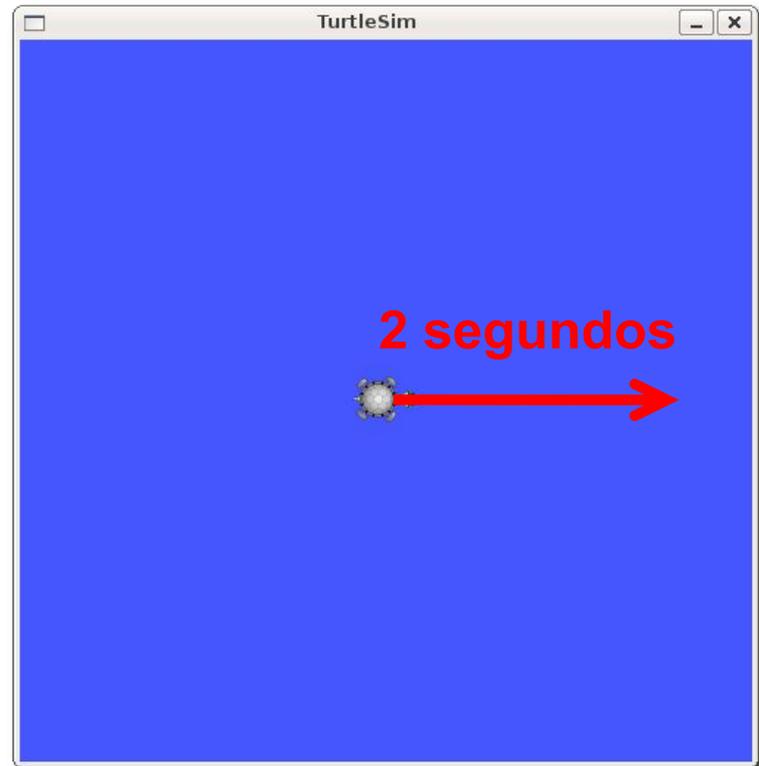
/turtle1/color_sensor

/turtle1/pose



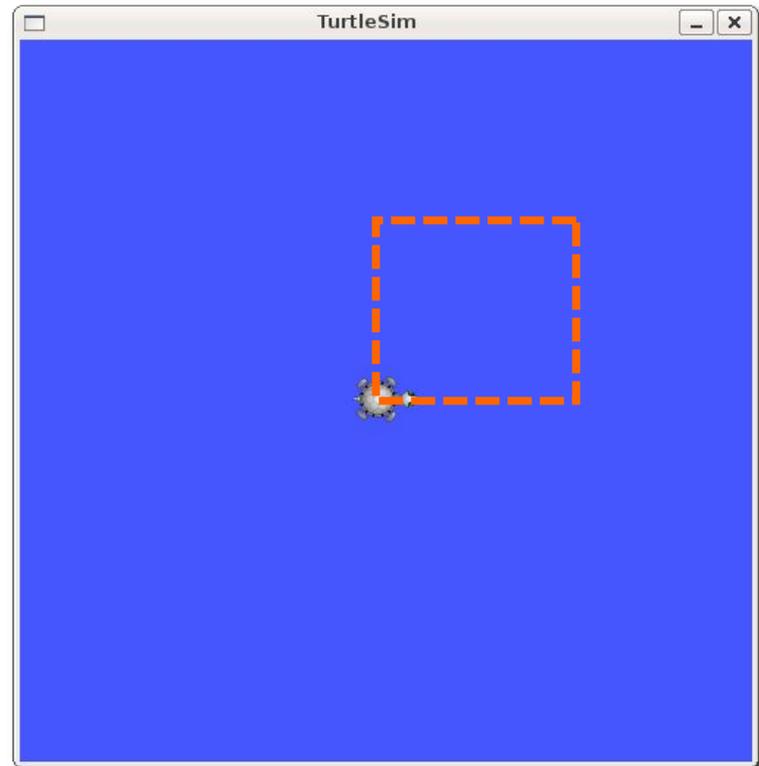
Atividade 1: Movimento simples

- **Objetivo:** realizar uma sequencia de movimentos com o robô turtle para execução de uma trajetória quadrada



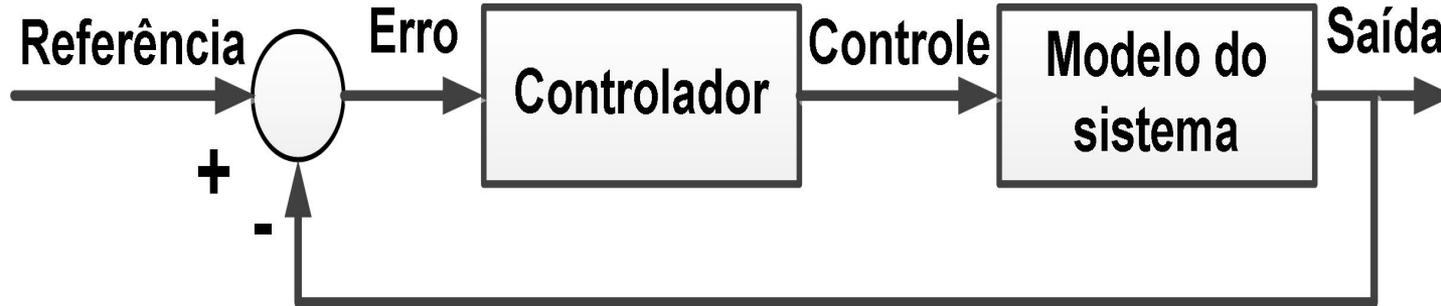
Atividade 2: Sequencia de movimentos

- **Objetivo:** realizar uma sequencia de movimentos com o robô turtle para execução de uma trajetória quadrada



Introdução

- As estratégias de controle automático são algoritmos para o controle de uma determinada grandeza em um processo
- O controlador é o elemento do sistema que aplica o sinal de erro para corrigir a atuação do sistema



Controle Proporcional

- O controle proporcional utiliza a saída do controlador para realizar uma realimentação proporcional em sua entrada
- O sinal de erro é utilizado para corrigir a saída do controlador onde um ganho proporcional é empregado para “agilizar” a correção do sistema, assim:

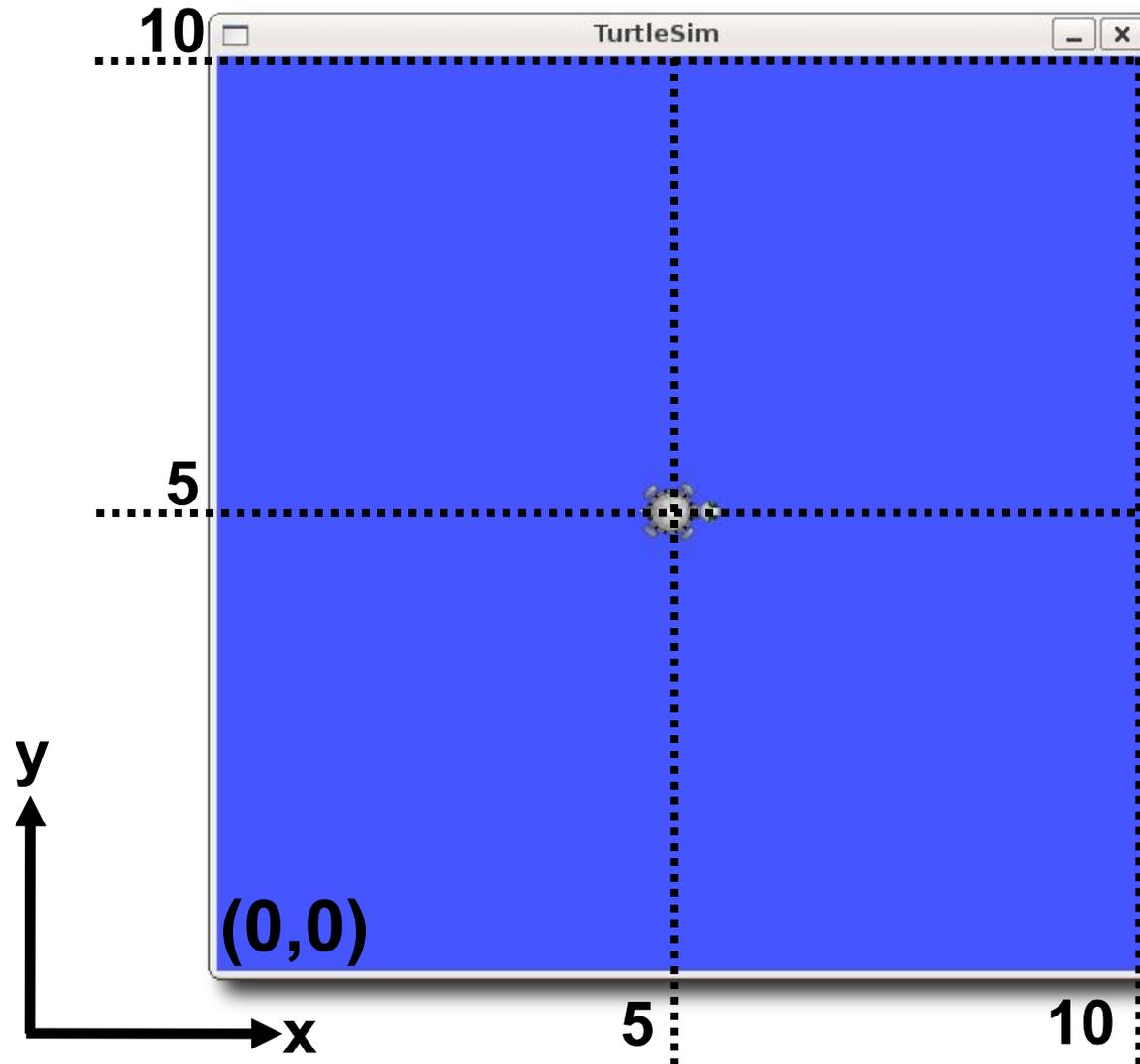
$$Saída = K_p \cdot erro$$

- Onde K_p é o ganho proporcional e a saída do controlador depende apenas da amplitude (tamanho) do erro

Controle Proporcional

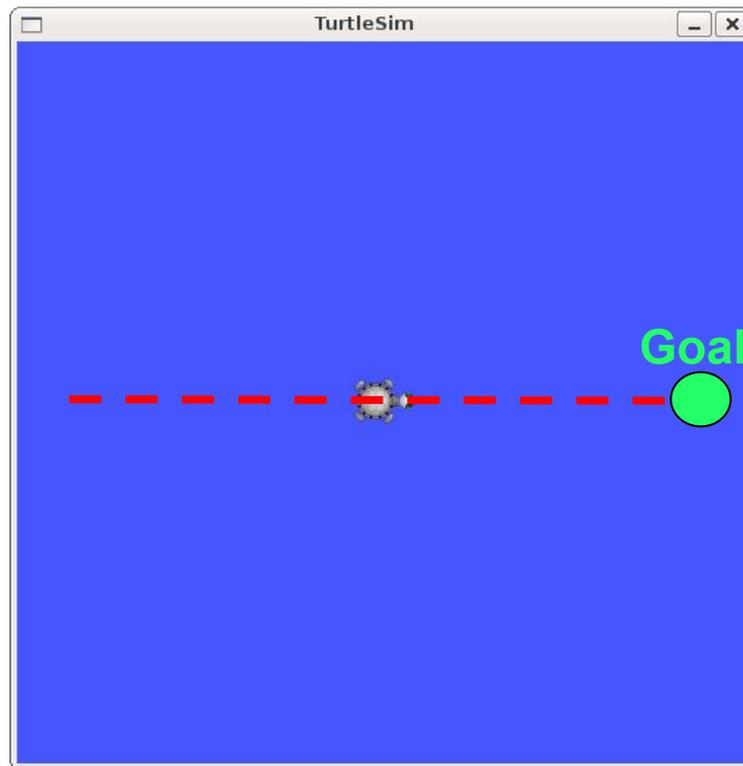
- O controlador é apenas um amplificador com um ganho constante
- Um grande erro em algum instante de tempo pode levar a um alto valor na saída do controlador
- Pode ocorrer a sobrepassagem (overshoot), por isso o ganho proporcional deve ser ajustado com cautela

Ambiente *TurtleSim*



Controle de posição em X

- **Objetivo:** realizar o controle do movimento do robô turtle ao longo do eixo X para atingir uma determinada posição (Goal)



Solução

```
% posicao desejada
desired_x = 5;

%tolerancia de correção
tolerance = 0.1;

%ganho proporcional
Kp = 10;

%cria o subscriber
sub = rossubscriber('/turtle1/pose')

%cria o publisher
pub = rospublisher('/turtle1/cmd_vel')

%define a mensagem
twist =
    rosmesssage('geometry_msgs/Twist')
```

```
while abs(error)> tolerance

    %recebe a mensagem
    posedata = receive(sub, 10)

    %captura os campos da mensagem
    error = desired_x - posedata.X;

    % acao corretiva
    twist.Linear.X = Kp*error;

    %envia a mensagem
    send(pub,twist)

    %visualiza a posicao
    posedata.X
end
```

Controle de posição em X e Y

- **Objetivo:** realizar o controle do movimento do robô turtle ao longo dos eixos X e Y para atingir uma determinada posição (Goal)



Referências

- Instalação do ROS Indigo

<http://wiki.ros.org/indigo/Installation/Ubuntu>

- Tutoriais do ROS-Matlab

<https://www.mathworks.com/help/robotics/robot-operating-system-ros.html>