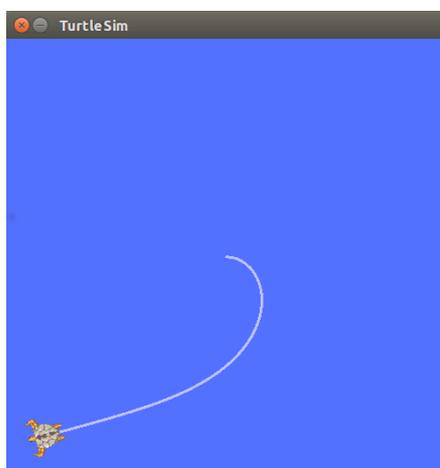


Universidade Tecnológica Federal do Paraná (UTFPR)
Disciplina: CPGEI/PPGCA - Robótica Móvel

Prof. André Schneider de Oliveira
Prof. João Alberto Fabro

Exercício 2 - Movimentação até um ponto desejado

Descrição: Desenvolva um pacote compatível com o Robot Operating System (ROS) para o controle de movimentação do *TurtleSim* até um destino (ponto desejado), considerando sua orientação. O controlador deve ser um Proporcional-Integral-Derivativo (PID) que realize a gestão dos movimentos lineares e angulares simultaneamente, conforme demonstrado na figura abaixo.



A entrada de referência do controlador pode ser realizada pela leitura de teclado com travamento (*scanf* ou *cin*). A realimentação considerará o tópico `/turtle1/pose` do tipo `turtlesim/Pose` e a saída da estratégia de controle será o sinal de atuação do robô para as velocidades linear em X e angular em Z, no tópico `/turtle1/cmd_vel` do tipo `geometry_msgs/Twist`.

O pacote ROS deve ser criado com as dependências: **roscpp**, **std_msgs**, **geometry_msgs** e **turtlesim**, com o nome de **controle_posicao**. No código fonte principal **pid.cpp** devem ser declaradas constantes/variáveis com os valores dos ganhos P, I e D.

**** Dicas importantes ****

- Para que a movimentação seja simultânea em ambos os movimentos (linear e angular) pode-se empregar duas abordagens distintas:

1. Implementar duas estratégias de controle independentes, mas publicar ao `cmd_vel` apenas após processar ambas;

2. Empregar uma estratégia de controle multivariável (ou seja, matricial), onde cada grandeza fique em uma linha. Para isso, é sugerido o uso da biblioteca *Eigen* (eigen.tuxfamily.org/)

- O *TurtleSim* pode ser reinicializado pelo nó ROS através da chamada do seu serviço de *reset*, pelo comando

```
system("rosservice call reset");
```

- O cálculo do tempo pode ser realizado pelo trecho de código abaixo.

```
ros::Time last_time, actual_time;  
...  
last_time = actual_time;  
actual_time = ros::Time::now();  
double dt = (actual_time - last_time).toSec();
```

- Experimente adicionar o comando abaixo para visualizar a variação de posição na forma gráfica e em tempo de execução.

```
system("rqt_plot /turtle1/pose/x:y &");
```