Low Level Control for Robot Soccer Utilities

Randal W. Beard
Department of Electrical & Computer Engineering
Brigham Young University, Provo, Utah 84602

January 23, 2002

1 Control

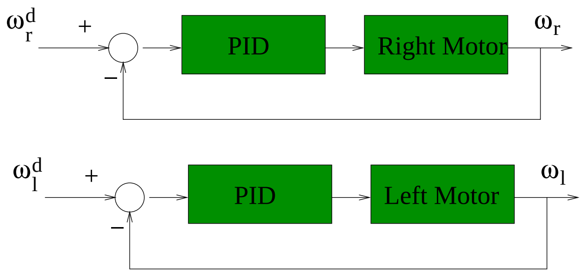
The objective is to write a few control function encapsulated as utilities for skills, and then to write all of the other skills in terms of fundamental building blocks. As a first example of a skill, lets write a skill called

`sklSpin(mag)`

that causes the robot to spin at magnitude `mag`. Example code is included in `sklSpin.c`. Notice that the last thing that happens in this code is to set `X.dr[0]` and `X.dl[0]`. The next skill that we would like you to write is

`void sklMoveWheels(double wrd, double wrl),`

where `wrd` is the desired angular speed of the right wheel and `wrl` is the desired angular speed of the left wheel. The PID structure shown in the figure below can be used to implement this skill.



Example code that shows how this skill might be implemented is shown in `sklMoveWheels.c`

The other skills required by this lab can be written in terms of `sklMoveWheels()`. For example, consider the task of writing a skill called `sklMoveAtVel(v,w)`, where v is the desired linear speed of the robot and ω is the desired angular speed of the robot. As explained in the modeling section, the relationship between v and ω and ω_r and ω_l is given by

$$\begin{pmatrix} \omega_r \\ \omega_l \end{pmatrix} = \begin{pmatrix} \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & \frac{-b}{r} \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix},$$

therefore sample code which could be used to implement `sklMoveAtVel(v,w)` is sklMoveAtVel.c.

As another example, consider writing a skill called `sklTurnToAng(ang)`, where `ang` is the desired angle. If the robot is stationary then the robot kinematics become

$$\begin{aligned} \dot{r}_x &= 0 \\ \dot{r}_y &= 0 \\ \dot{\psi} &= \omega. \end{aligned}$$

Therefore only the third equation is relevant. Let $\tilde{\psi} = \psi - \psi^d$ where ψ^d is the desired angle. Then

$$\dot{\tilde{\psi}} = \dot{\psi} - \dot{\psi}^d = \omega.$$

Letting $\omega = -k_p \tilde{\psi}$ results in the differential equation

$$\dot{\tilde{\psi}} = -k_p \tilde{\psi},$$

which implies that

$$\tilde{\psi}(t) = \tilde{\psi}(0) e^{-k_p t} \to 0.$$

Sample code that impliments `sklTurntoAng()` is sklTurnToAng.c.