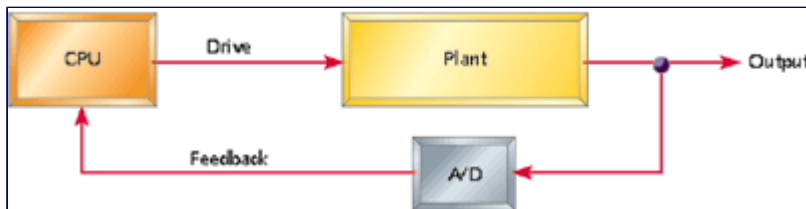


General Segway Control

The basic idea for a two-wheeled dynamically balancing robot is pretty simple: drive the wheels in the direction that the upper part of the robot is falling. If the wheels can be driven in such a way as to stay under the robot's center of gravity, the robot remains balanced. In order to accelerate or decelerate the robot, simply let it fall slightly to one of the sides, and the balancing mechanism will cause the robot to accelerate the same way. The speed and the following position of the robot may then be controlled by carefully adjusting the acceleration.

Although the principle is simple and straight-forward, there are many approaches to this control problem. All the control strategies are based on feedback from the robot in the form of sensors. This feedback must include information about the robot angle and preferably also about the angular velocity and the position of the wheels. The controller compares the measured value from the robot with a reference setpoint value. The difference (or "error" signal) is then processed to calculate a new value for the motor input that brings the robot measured value back to its desired setpoint. Such a system is called a closed loop system and the basic principle is shown in figure 7.1. [1]



Linear Quadratic Control

Pure PID control does not make any assumptions about the system to be controlled and any information about the system characteristics is not strictly necessary. In the case where the characteristics of the system is known or partly known (as in this case), controllers may be developed based on that knowledge. This is what is called model based control. (MBC).

The State Space Model:

Four terms are sufficient to define the motion and position of this "inverted pendulum" and thereby balance the robot. These are 1- the tilt angle and 2 - its first derivative, the angle velocity, 3 - the platform position and 4 - its first derivative, the platform velocity. These four terms are considered as the system states. Classic control theory does not handle nonlinear systems and therefore any nonlinearities must be approximated by a linear function which is similar within a certain area and this process is called linearization. By linearizing the physical equations for the four states of the robot one can derive the state space model:

$$\begin{bmatrix} \ddot{x} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u$$

or

$$\dot{\bar{x}} = A\bar{x} + Bu$$

Where;

l is the height of the center of mass,

m is the total system mass,

M is the mass equivalent of the wheels (mass of cart),

b is the linear motor damping,

I is the robot moment of inertia about the center of mass and

g is the gravity constant.

For the purpose of the design process we assume that we have full-state information (i.e. that we can measure all four states). State feedback control is based on introducing a feedback to the system by calculating the input u as:

$$u = -K\bar{x} = -\begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} = -k_1x - k_2\dot{x} - k_3\phi - k_4\dot{\phi}$$

This type of controller may be implemented directly on a microcontroller, given the 4 states are measured continuously. The controller is simply the implementation of the equation above. The vector of feedback gains (K) may be computed in different ways. A possible way of doing this is by pole-placement [2], but a method that is perhaps even more elegant is to calculate the feedback gains using an optimization problem. The resulting controller is called a Linear Quadratic Controller and is the kind used in the barbie segway project.

Since the robot in question does not have wheel feedback (encoders), the two last states (speed and position) had to be estimated based on the known input force and motor damping. This method works well as long as surface is plane and smooth but experiences some problems when the surface is not. This could have been avoided using high resolution encoder feedback.

References

- [1] Michael Barr's Embedded Systems Glossary - Introduction to Closed-Loop Control
<http://www.netrino.com/Publications/Glossary/PID.html>
- [2] Ruhr-Universität Bochum - Design of state-feedback controllers by pole placement
<http://www.atp.ruhr-uni-bochum.de/rt1/syscontrol/node103.html>