Stopping Smoothly

Benjamin Kuipers

December 3, 1999

An important basic skill is to stop smoothly and firmly at a desired point. This is particularly difficult for the wheelchair, since it has such poor motor control at low speeds, but the same problem exists for any robot.

Suppose that the robot starts at \( x = 1 \), moving toward the origin, and wants to stop at \( x = 0 \). It is at rest at \( t = t_s \).

\[
\begin{align*}
x(0) &= 1 & x(t_s) &= 0 \\
\dot{x}(0) &= 0 & \dot{x}(t_s) &= 0
\end{align*}
\]

Velocity Control

If velocity is directly controllable, then the first-order control law

\[
\dot{x} = -f(x) \quad \text{where} \quad f \in M_0^+
\]

will bring the system to rest at \( x = 0 \) for any \( f \in M_0^+ \) (that is, for any monotonically increasing \( f \) such that \( f(0) = 0 \)). Since any function \( f \) will work, we can pick the \( f \) that has the performance properties we want.

The linear controller \( f(x) = kx \) is the typical choice. We can solve the equation of motion explicitly, to get \( x(t) = e^{-kt} \), which approaches its rest position asymptotically at \( t_s = \infty \), which is poorly suited to the wheelchair.

Instead, we will use

\[
f(x) = k\sqrt{x}
\]

Solving this analytically, with the initial condition \( x(0) = 1 \), we get the predicted behavior

\[
x(t) = (1 - kt/2)^2
\]

which is a parabolic (rather than exponential) drop from \( x(0) = 1 \) to \( x(t_s) = 0 \), where \( t_s = 2/k \). That is, finite rather than infinite time to stop.

We can also solve explicitly for velocity:

\[
v(t) = k^2t/2 - k.
\]
The initial velocity of the system determines $k$: $v(0) = -k$. Acceleration is thus constant, at $k^2/2$. This can be compared with the maximum permissible acceleration of the robot, to see whether the stopping goal is feasible.

**Feedback Control**

The straight-forward implementation of the control law (1) is to sense $x(t)$, and update $\dot{x}(t)$ appropriately, at each time-step $t$.

This feedback control law is robust against sensor and motor errors, but has the cost, both in computation and in sensory delay, of sensing the world at each time step.

**Open-Loop (= Feed-forward?) Control**

Since we can explicitly solve for the time-course of the control signal $v(t)$, given observations of $x(0)$ and $\dot{x}(0)$, we can simply follow equation (4) as a fixed policy, independent of further observations.

The open loop formulation also avoids any problems that might arise because $f'(x) \to \infty$ as $x \to 0$.

While the open loop control law clearly minimizes sensor effort and delay, it leaves the robot without feedback to compensate for sensor and motor errors.

**Model-Predictive Control**

We can get the best of both worlds by repeating the derivation of equation (4) every time we get new observations of $x(t)$ and $\dot{x}(t)$.

If sensing is slow compared with the control loop, the feed-forward policy provides an output signal to send whenever the control loop needs one. But if new sensory information becomes available, the model can be updated, compensating for sensory or motor errors, or for inaccuracies in the model.

**Caveats**

We have assumed that velocity is directly and instantaneously controllable, which is only an approximation. But we don’t need arbitrary controllability. All we need is the ability to achieve a constant deceleration of $k^2/2$.

It may be necessary to have a special method to handle the final moment of stopping, to prevent velocity from overshooting zero.
This controller should be very easy to implement and test, using the forward range sensor to test how close to the desired point the wheelchair actually stopped. (Aim for 1.00 m from the wall.)