Stabilizing the inverted pendulum with PID control

1 PID control

The PID controller is one of the simplest heuristics for approaching the inverted pendulum problem. A PID controller manages error in a dynamic system by driving a process variable towards a setpoint. It does this by applying a control signal which depends on three error-related terms: a proportional, integral, and derivative term.

The error e_i at a given time step i is defined as the difference between the process variable and the setpoint:

$$e_i = process\ variable_i - setpoint \tag{1}$$

The proportional term of the controller is the product of the proportional gain constant k_p and the error:

$$P = k_p e_i \tag{2}$$

This term makes the controller immediately responsive to the present error.

The integral term of the controller is the product of the integral gain constant k_i and the sum over error at all past time steps:

$$I = k_i \sum_{t=0}^{i} e_t \tag{3}$$

This term makes the controller sensitive to accumulation of error over time; it helps in situations where the proportional term is not enough to avoid steady-state error.

The derivative term of the controller is the produce of the derivative gain constant k_d and the difference between error at the present time step and the past time step:

$$D = k_d(e_i - e_{i-1}) \tag{4}$$

This term makes the controller sensitive to rapid changes in the error; it can help provide an extra "boost" to the controller when error is increasingly rapidly, or reduce the output when error is rapidly decreasing to avoid overshooting.

The output of the controller is the sum of these three terms:

$$\begin{aligned} output &= P + I + D \qquad (5) \\ &= k_p e_i + k_i \sum_{t=0}^{i} e_t + k_d(e_i - e_{i-1}) \qquad (6) \end{aligned}$$

The control constants k_p, k_i, and k_d are selected by heuristic tuning.

2 A behavioral hierarchy

The PID controller is readily applied to the problem of stabilizing an inverted pendulum: the process variable is the angle of the rod, the set point is the desired angle of the rod, and the control signal (or output of the PID controller) is a force presumably applied via an internal motor, on the cart.

The above controller, when well-tuned, will properly stabilize the inverted pendulum. However, it is not enough: it places no constraints on the range of motion of the cart, so the cart may easily fly off the track while moving to keep the pendulum stable. To solve this problem, we construct another (higher-level) PID controller to keep the cart close to the center of the track. In this controller, the process variable is the position of the cart, the set point is the center of the track, and the control signal is the desired angle of the rod.

Note that the output of this high-level controller is the input to our low-level controller which stabilizes the pendulum. This is called cascading PID control. When the high-level controller requires that the cart go left, the low-level controller aims to lean the pendulum slightly left of center; when the cart must go right, the low-level controller leans the pendulum right. We have created our first behavioral hierarchy: the low-level motor program (balancing the pendulum) experiences top-down modulations to act in service of a high-level goal (staying centered on the track).