Readme

Notes on Reinforcement Learning Java Source for Dynamic Systems
Bernhard Hengst 12th March 2012

Aim

This experimental code is designed to return the optimal control action given the current state to achieve a goal state. In the example code the goal is to move a cart from any given position and velocity to the zero position and stop. The code can be adapted to stabilize a Nao.

Outline

The dynamics of the model (transitions) are constructed by using a simple process update based on know physics of the cart (or inverted pendulum). Then RL is used to solve the shortest path problem to the goal.

Classes

RLwrapper – constructs the dynamic model, sets the goal and states that are out of bounds, and provides an interface between the robot and the reinforcement learner.

RLModel – stores transition model cases (to approximate the continuous transition function), calculates the optimal policy, and returns the optimal action for any query state.

Transition – stores individual transition cases

Assistance

This is experimental code at the time of writing. It has been tested on a simple cart with two state variables (position and velocity). Anyone intending to implement this code should consult Bernhard Hengst for assistance.