CSCE 790: Neural Networks and Their Applications

AIISC and Dept. Computer Science and Engineering
Email: vignar@sc.edu

© Vignesh Narayanan

October 31, 2023

UNIVERSITY OF
South Carolina

Example Control Problem

- Given
  - The desired or the reference trajectory for the robotic system to track
  - Measurements from the sensor informing the actual path/trajectory of the robotic system
- To Do
  - Design control inputs or policies that steers the actual path traced by the robotic system is close to the reference trajectory

Physics-based Model

- Robotic arm
$$M(q)\ddot{q}(t) + V_m(q,\dot{q}) + G(q) + F(q,\dot{q}) = \tau(t) + \tau_d(t)$$
- Dynamic Equations - Newton-Euler method or Lagrangian Dynamics
- $q(t)$ Joint variable
- $M(q)$ Models of inertial mass
- $V_m(q,\dot{q})$ Models of coriolis/centripetal force
- $F(q,\dot{q})$ Models of friction
- $G(q)$ models of Gravity
- $\tau(t)$ Control torque
- $\tau_d(t)$ models of disturbance

Tracking Control Problem

- Let the desired trajectory for the robot manipulator be $q_d(t)$
- Now, we can define the tracking error as

$$e(t) = q_d(t) - q(t)$$

- Define the filtered tracking error as

$$r(t) = \dot{e}(t) + \lambda e(t)$$

- Filtered tracking error dynamics

$$\dot{r}(t) = \ddot{e}(t) + \lambda \dot{e}(t)$$

Tracking Control Problem

- Filtered tracking error dynamics are: $\dot{r}(t) = \ddot{e}(t) + \lambda\dot{e}(t)$
- Recall the robot dynamics: $M(q)\ddot{q}(t) + V_m(q,\dot{q}) + G(q) + F(q,\dot{q}) = \tau(t) + \tau_d(t)$

$$M\dot{r}(t) = -V_m r(t) - \tau(t) + h + \tau_d(t)$$
$$h = M(q)(\ddot{q}_d + \lambda\dot{e}) + V_m(q,\dot{q})(\dot{q}_d + \lambda e) + F(\dot{q}) + G(q)$$

Control Torque

$$\tau(t) = \hat{h} + K_v r(t)$$

with λ, K_v being a positive design parameter

- The closed-loop dynamics is obtained as

$$M\dot{r}(t) = -V_m r(t) - \hat{h} - K_v r(t) + h + \tau_d(t)$$

NN Control - Function Approximator

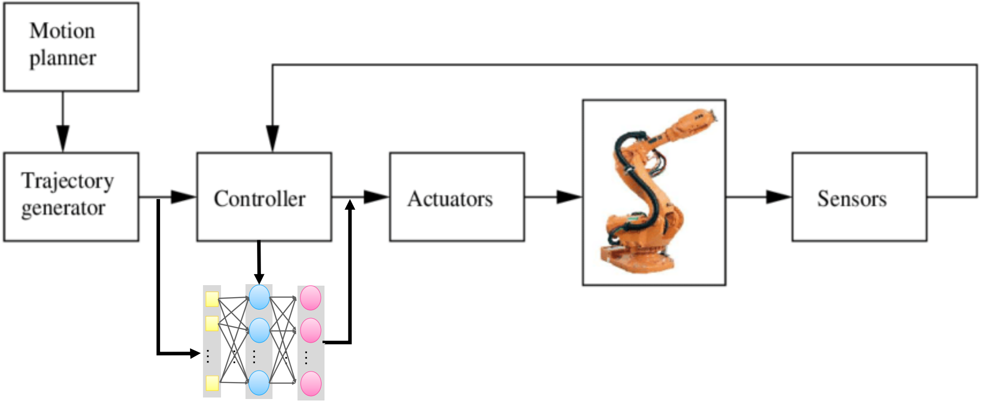

Figure: Feedback NN control

Steady-State Analysis of Feedback Control System

- Filtered tracking error dynamics

$$\dot{r}(t) = -\frac{V_m - K_v}{M} r(t) + \frac{h - \hat{h}}{M} + \frac{\tau_d(t)}{M}$$

$$\downarrow$$

$$\dot{r}(t) = -Kr(t) + N_\varepsilon + d(t)$$

- What does the Lyapunov approach reveal?

Example

Example (Given)

- Adaptive control – origins of data-integrated control
- System description: $\dot{y}(t) = ay(t) + b[u(t) - f(y(t))]$, $y(0) \in \mathbb{R}$
- Reference model: $\dot{y}_m(t) = a_m y_m(t) + b_m r(t)$

Example (Control Design)

- $f(y(t)) \approx \hat{f}(y(t)) = W\phi(Vy(t))$, W, V are unknown parameters, $\phi(\cdot)$ are user-defined basis functions
- Assumption: $\exists k_y, k_r : a_m = a + bk_y,\ b_m = bk_r$
- Pick control: $u(t) = \hat{k}_y y(t) + \hat{k}_r r(t) + \hat{f}(y(t))$

Example

Example (Goal)

- Find $\dot{\hat{k}}_y(t)$, $\dot{\hat{k}}_r(t)$, $\dot{W}(t)$, $\dot{V}(t)$ such that as $t \to \infty$, $\hat{k}_y \to k_y$, $\hat{k}_r \to k_r$, $\hat{f} \to f$, $y(t) \to y_m(t)$

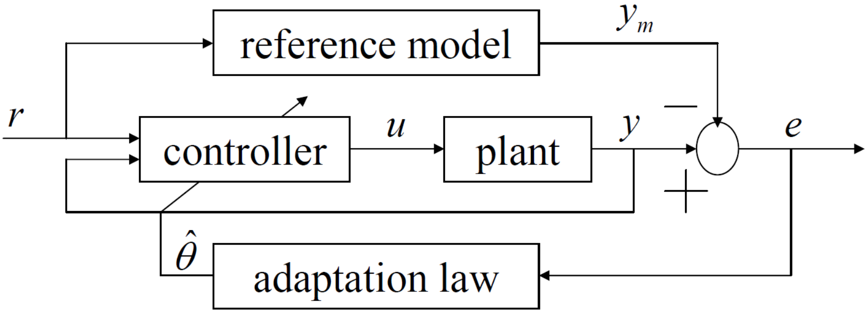

Figure: Adaptive Control-MRAC

Check These for More on NN Control

- NN for Control: Tutorial
- NN for Control Using RL Algorithm
- NN Control of Robot Manipulator

Reinforcement Learning

- An 'agent' is only provided with a grade, or score, which indicates performance, and the objective is to maximize the reward over a long-time interval.

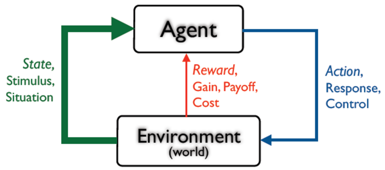

Figure: Reinforcement Learning

- Input data and a 'reward' output is given
- RL problems: Decision making problems
- Environment is unknown, nonlinear, and potentially stochastic, complex
- Policy: A mapping from states to actions
- RL seeks to learn a policy that maximizes the agent's reward in the long run
- RL is important because it is a very general formulation of the AI problem and its objective with autonomy (no knowledgeable supervision)

Formulation as Markov Decision Process

- Physics-based models are useful
  - Models are not always available
  - Models are almost always not perfect
- Markov Decision Process (MDP)
  - Can also be viewed in a feedback control paradigm
  - Common (modeling) in Reinforcement Learning

Agent and Environment

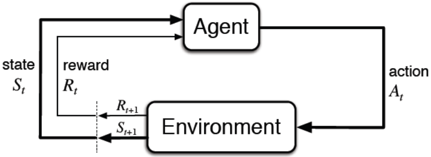

Figure: Agent-Environment Interaction

- Agent observes state at step t, $S_t \in \mathcal{S}$
- Produces action at step t, $A_t \in \mathcal{A}$
- Gets a rewards as a consequence, $R_{t+1} \in \mathcal{R}$, a scalar
- Environment transits to a new state $S_{t+1} \sim T(S_t, A_t)$

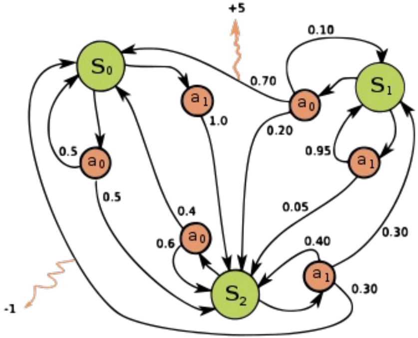

Figure: Finite Markov decision process

Terminologies/Analogies

- Environment - unknown, nonlinear, and potentially stochastic, complex $\rightarrow$ Dynamical System
- Policy - mapping from states to actions $\rightarrow$ Feedback Control law
- Agent – Decision maker $\rightarrow$ Controller
- Action – Decision $\rightarrow$ Control input

Popular NN Architectures

- Autoencoder and Variational Autoencoder
- Generative Adversarial Network
- Long Short Term Memory Network
- Echo State Networks (Reservoir Computing Network)
- Convolutional Neural Network
- Graph Neural Network