Data Driven Model Learning and Convergence of Actor-Critic

Shahzeb Aamir

Data-Driven Robust Approximate Optimal Tracking Control for Unknown General Nonlinear Systems Using Adaptive Dynamic Programming Method, Zhang et. al

Content:

Main Contributions

The main contributions of this paper include the following.

1) It is the first time that the optimal tracking problem of the unknown general nonlinear systems based on ADP method is investigated.
2) A novel data-driven model is established based on an RNN which guarantees the modeling error to asymptotically converge to zero.
3) A robust approximate optimal tracking controller is developed to ensure that the tracking error converges to zero asymptotically. Moreover, the proposed controller can ensure that the obtained control input is close to the optimal control input within a small bound.
4) In the design of the optimal feedback controller based on ADP, the critic NN and the action NN are updated simultaneously.

Problem Formulation

1) It is the first time that the optimal tracking problem of the unknown general nonlinear systems based on ADP method is investigated.

Continuous Non-linear System:

$$\dot{x}(t) = h(x(t), u(t))$$

Objective:

Optimal Controller such that state vector x(t) follows a specified trajectory $x_d(t)$ and minimizes some Performance Index:

$$V(e(t)) = \int_t^{\infty} r(e(\tau), u(e(\tau)))d\tau$$

$r(e(t), u(t)) = e^T(t)Qe(t)+u^T(t)Ru(t)$ is the utility function.

Usually solved by ADP, but dynamics must be known. Here we don't know the dynamics, so we learn by Input-Output data (Data-Driven Model Learning)

Model Learning by RNN

2) A novel data-driven model is established based on an RNN which guarantees the modeling error to asymptotically converge to zero.

The RNN model:

$$\dot{x}(t) = C^{*T}x(t) + A^{*T}f(x(t)) + C_u^{*T}u(t) + A_u^{*T} + \varepsilon(t)$$

Data-Driven Model

$$\dot{\hat{x}}(t) = \hat{C}^T(t)\hat{x}(t) + \hat{A}^T(t)f(\hat{x}(t)) + \hat{C}_u^T(t)u(t) + \hat{A}_u^T(t) - v(t) \qquad (5)$$

$$v(t) = Se_m(t) + \frac{\hat{\lambda}(t)e_m(t)}{e_m^T(t)e_m(t) + \eta}$$

Assumption 1 [39]: The term $\varepsilon(t)$ is assumed to be upper bounded by a function of modeling error such that

$$\varepsilon^T(t)\varepsilon(t) \le \varepsilon_M(t) = \lambda^* e_m^T(t)e_m(t) \qquad (7)$$

The modelling error will converge to zero, if we choose the weight matrices in a certain way. For details, see the original paper. But the idea is, we can drop the error term. Which as a results gives us this -> $\dot{x}(t) = C^T x(t) + A^T f(x(t)) + C_u^T u(t) + A_u^T.$

Optimal Tracking Control Problem

3) A robust approximate optimal tracking controller is developed to ensure that the tracking error converges to zero asymptotically. Moreover, the proposed controller can ensure that the obtained control input is close to the optimal control input within a small bound.

$$\dot{x}(t) = C^T x(t) + A^T f(x(t)) + C_u^T u(t) + A_u^T.$$

$$\dot{x}_d(t) = C^T x_d(t) + A^T f(x_d(t)) + C_u^T u_d(t) + A_u^T$$

$$\dot{e}(t) = C^T e(t) + A^T f_e(t) + C_u^T u_e(t)$$

Controller Consists of two parts:

Steady State Controller

$$u_d(t) = C_u^{-T}\left(\dot{x}_d(t) - C^T x_d(t) - A^T f(x_d(t)) - A_u^T\right)$$

FeedBack Controller (After some derivation)

$$u_e^* = -\frac{1}{2} R^{-1} C_u V_e^*$$

Optimal Tracking Control Problem

$$u_e^* = -\frac{1}{2}R^{-1}C_u V_e^*$$

$$V^*(e) = \min_{u_e \in \psi(\Omega)} \left(\int_t^{\infty} r(e(\tau), u_e(e(\tau)))d\tau \right)$$

$$V_e^* = \partial V^*(e)/\partial e.$$

Critic NN Design

$$V(e) = W_1^T \phi_1(e) + \varepsilon_1$$

$$\hat{V}(e) = \hat{W}_1^T \phi_1(e).$$

$$\begin{aligned} H\left(e, u_e, \hat{W}_1\right) &= \hat{W}_1^T \nabla\phi_1 \left(C^T e + A^T f_e + C_u^T u_e\right) + e^T Q e + u_e^T R u_e \\ &= e_1. \end{aligned} \tag{29}$$

$$E_1(\hat{W}_1) = \frac{1}{2}e_1^T e_1.$$

$$\dot{\hat{W}}_1 = -a_1\sigma_1 \left(\sigma_1^T \hat{W}_1 + e^T Q e + u_e^T R u_e\right)$$

Optimal Tracking Control Problem

$$u_e^* = -\frac{1}{2}R^{-1}C_u V_e^*$$

$$V^*(e) = \min_{u_e \in \psi(\Omega)} \left(\int_t^{\infty} r(e(\tau), u_e(e(\tau)))d\tau \right)$$

$$V_e^* = \partial V^*(e)/\partial e.$$

Actor NN Design

$$u_e = W_2^T \phi_2(e) + \varepsilon_2$$

$$\hat{u}_e = \hat{W}_2^T \phi_2(e).$$

feedback control input applied to the error system and the control input minimizing $\hat{V}(e) = \hat{W}_1^T \phi_1(e).$

$$e_2 = \hat{W}_2^T \phi_2 + \frac{1}{2}R^{-1}C_u \nabla\phi_1^T \hat{W}_1.$$

$$E_2(\hat{W}_2) = \frac{1}{2}e_2^T e_2.$$

$$\dot{\hat{W}}_2 = -a_2\phi_2 \left(\hat{W}_2^T \phi_2 + \frac{1}{2}R^{-1}C_u \nabla\phi_1^T \hat{W}_1 \right)^T$$

Stability Result

Then the tracking error e and the weight estimate errors W1 and W2 are UUB

Moreover, the obtained control input u is close to the optimal control input u* within a small bound ε_u

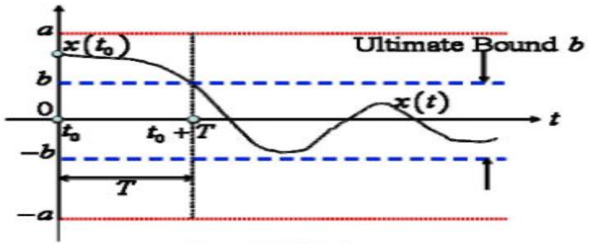


Figure 5.1: UUB Concept

Remark 4: If the NN approximation errors ε_1 and ε_2 are considered to be negligible, then from (49) we have $D_M = 0$, with $u \to u^*$. Otherwise, the obtained control input u is close to the optimal input u^* within a small bound ε_u.

Assumption 2:

1) The unknown ideal constant weights for the critic NN and the action NN, i.e., W_1 and W_2, are upper bounded so that $\|W_1\| \le W_{1M}$, $\|W_2\| \le W_{2M}$, respectively.
2) The NN approximation errors ε_1 and ε_2 are upper bounded so that $\|\varepsilon_1\| \le \varepsilon_{1M}$, $\|\varepsilon_2\| \le \varepsilon_{2M}$, respectively.
3) The vectors of the activation functions of the critic NN and the action NN, i.e., ϕ_1 and ϕ_2, are upper bounded so that $\|\phi_1(\cdot)\| \le \phi_{1M}$, $\|\phi_2(\cdot)\| \le \phi_{2M}$, respectively.
4) The gradients of the critic NN approximation error and the activation function vector are upper bounded so that $\|\nabla\varepsilon_1\| \le \varepsilon'_{1M}$, $\|\nabla\phi_1\| \le \phi_{dM}$. And the residual error is upper bounded so that $\|\varepsilon_{HJB}\| \le \varepsilon_{HJBM}$.

Additional Robustifying term can be added to attenuate for

NN approximation errors

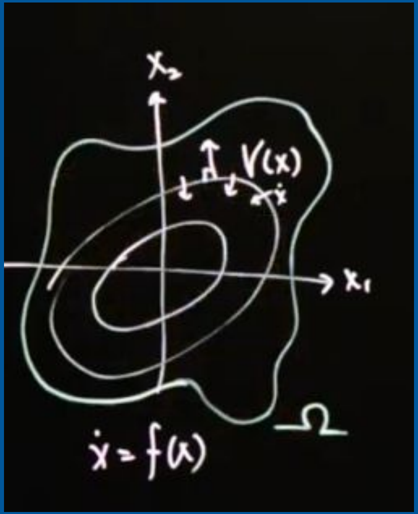
Stability Analysis Mechanism used here

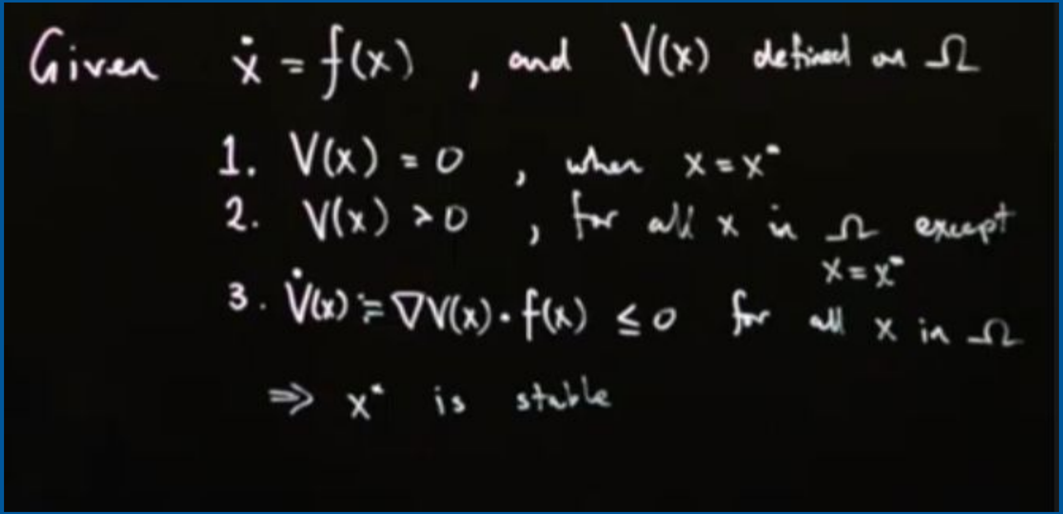
https://www.youtube.com/watch?v=td-d4Yi-81c

Choose candidate Lyapunov Function

Compute gradients of Lyapunov Function along trajectories of error system

Then Gradients must be negative





Example:

$$\dot{x}_1 = -x_1 + x_2$$
$$\dot{x}_2 = -0.5x_1 - 0.5x_2\left(1-(\cos(2x_1)+2)^2\right) + (\cos(2x_1)+2)u.$$

$$x_{1d} = \sin(t).$$

Ground Truth Model!

Discover dynamics from Input-Output data by RNN as described:

$$\dot{x}(t) = C^T x(t) + A^T f(x(t)) + C_u^T u(t) + A_u^T.$$

Choose all the required design parameters

Design the Actor-Critic Network to perform the tracking control problem

Example:

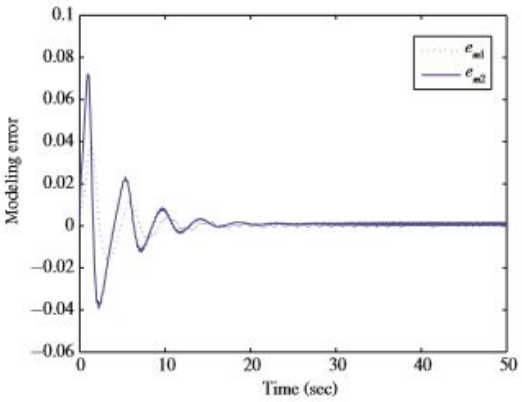


Fig. 1. Modeling error for the affine nonlinear system.

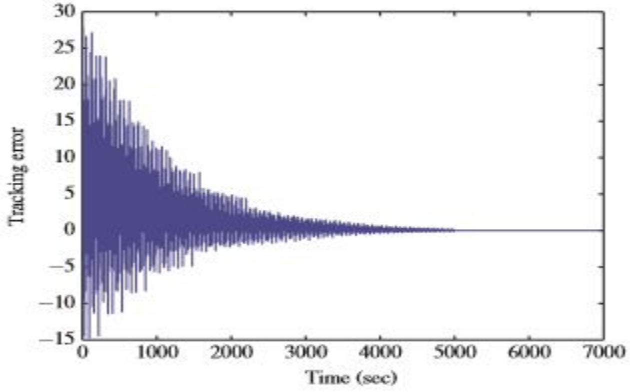


Fig. 2. Tracking error for the affine nonlinear system.

Example:

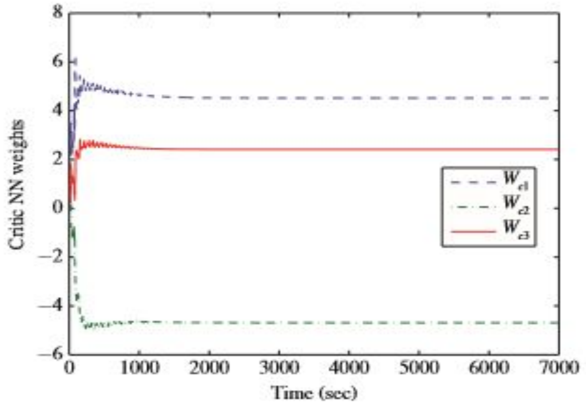


Fig. 3. Critic NN weights.

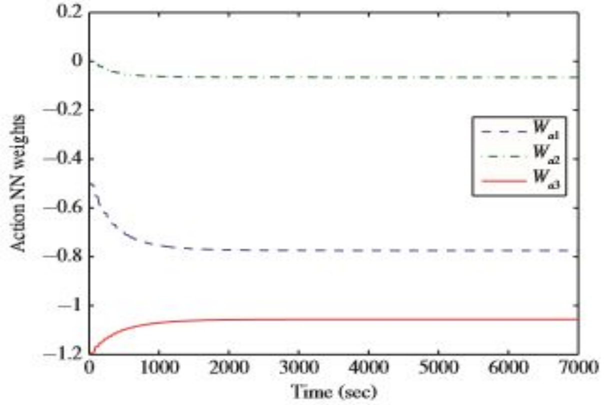


Fig. 4. Action NN weights.

Skoltech

Skolkovo Institute of Science and Technology

Thank You!