EML 6531-Adaptive Control Project 6

Submitted to: Dr.Warren Dixon, MAE Dept, University of Florida

Submitted by: Prajwal Gowdru Shanthamurthy

April 15, 2020

1)Simulation Section for the Continuous Neural-Network based Controller

1) Control Gains used:

- K and K_n multiplying r in the control law
- α multiplying e in the definition of the error signal r
- γ_1 and γ_2 are the learning rates in the update laws for the estimates of weight matrices ($\widehat{W}$ and $\widehat{V}$) in the neural network

The values of which were tuned as follows:

$$K = \begin{bmatrix} 20 & 0 \\ 0 & 5 \end{bmatrix},\ K_n = \begin{bmatrix} 20 & 0 \\ 0 & 5 \end{bmatrix},\ \alpha = 3,\ \gamma_1 = 500 \text{ and } \gamma_2 = 1000$$

NOTE: γ_1 and γ_2 can be chosen to be diagonal matrices of appropriate dimensions. If we don't wish to tune the diagonal elements separately, they can be chosen as scalar gains as shown above

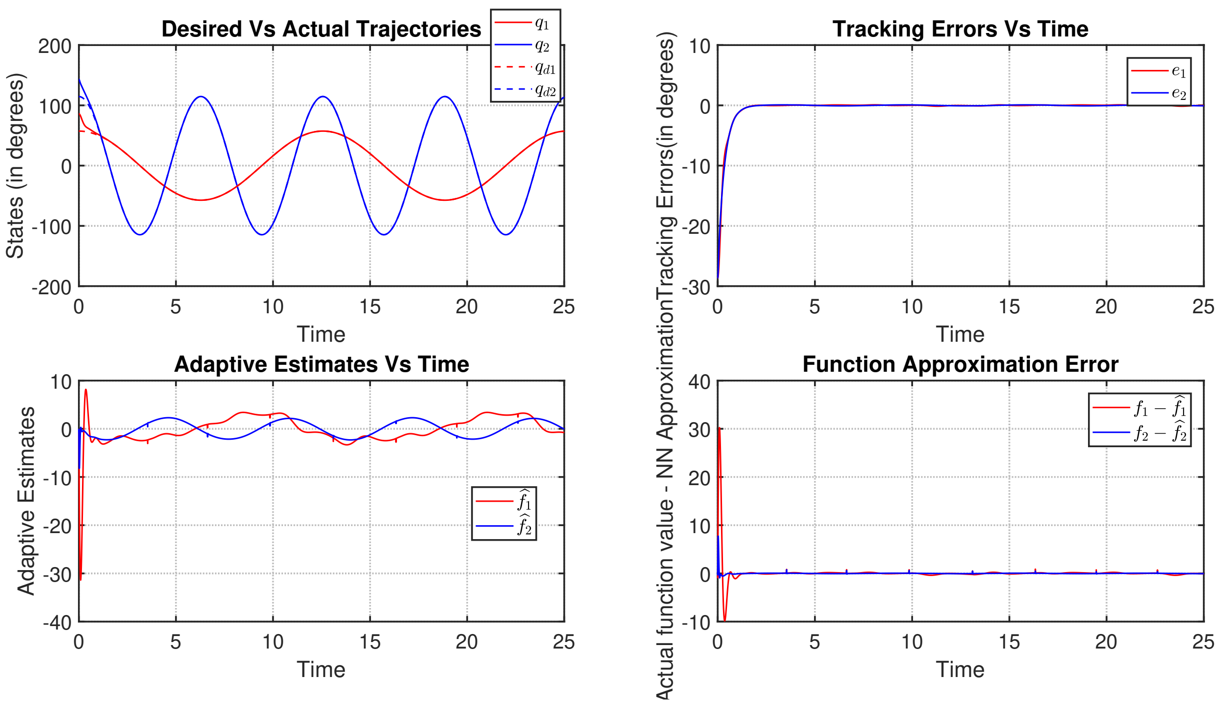

Figure 1: Continuous NN-based Control

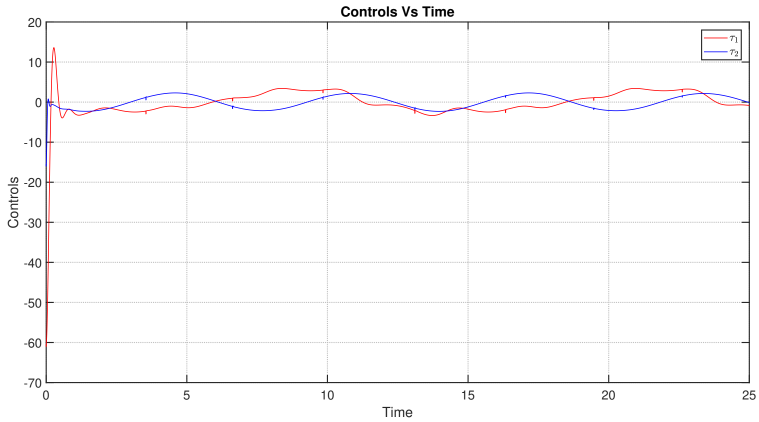

Figure 2: Control Inputs for Continuous NN-based Control

Maximum values of the torques are 60.9860 Nm and 15.9832 Nm respectively.

2)Simulation Section for the Sliding-Mode Neural Network Based Controller

1) Control Gains used:

- K and K_n multiplying r in the control law
- α multiplying e in the definition of the error signal r
- γ_1 and γ_2 are the learning rates in the update laws for the estimates of weight matrices ($\widehat{W}$ and $\widehat{V}$) in the neural network

The values of which were tuned as follows:

$K = \begin{bmatrix} 100 & 0 \\ 0 & 3 \end{bmatrix}$, $K_n = \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix}$, $K_s = 1$, $\alpha = 3$, $\gamma_1 = 500$ and $\gamma_2 = 1000$

NOTE 1: γ_1 and γ_2 can be chosen to be diagonal matrices of appropriate dimensions. If we don't wish to tune the diagonal elements separately, they can be chosen as scalar gains as shown above

NOTE 2: K and K_n have been chosen to be diagonal matrices instead of scalar constants to ensure that the control torques are deliverable by the motors while getting a good performance. (The maximum torque for motor 1 is 250 Nm, while the maximum torque for motor 2 is 30 Nm)

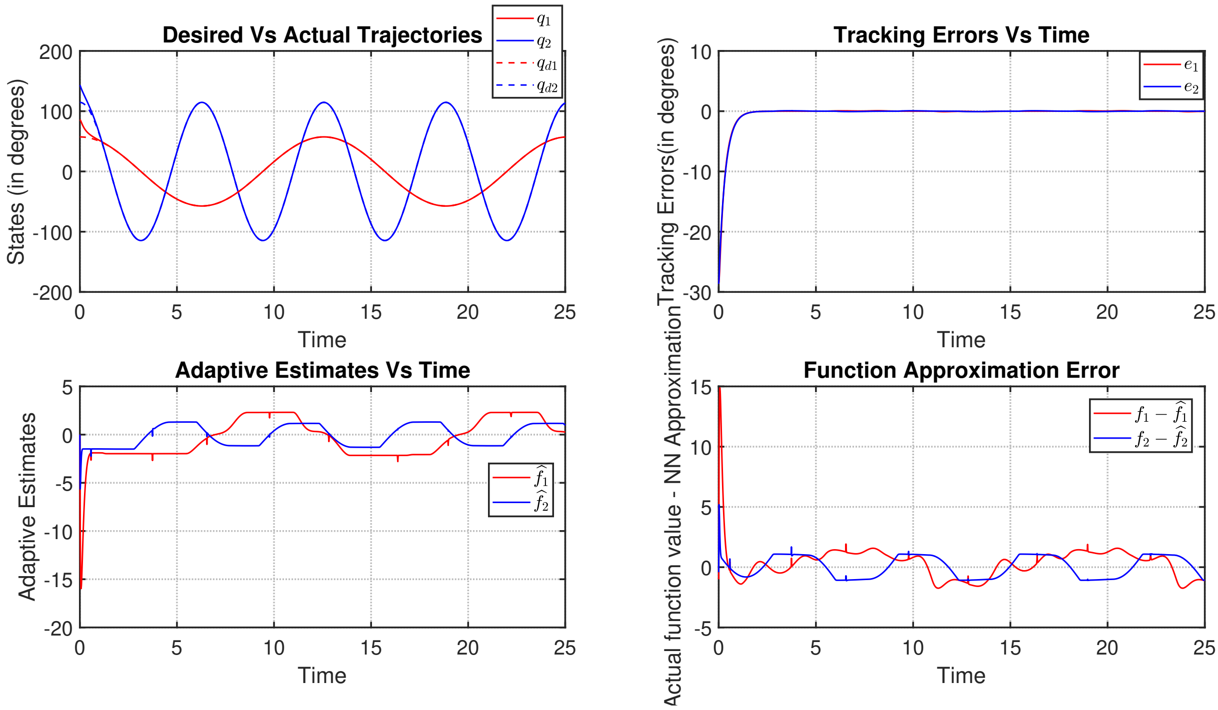

Figure 3: Sliding-Mode NN-based Control

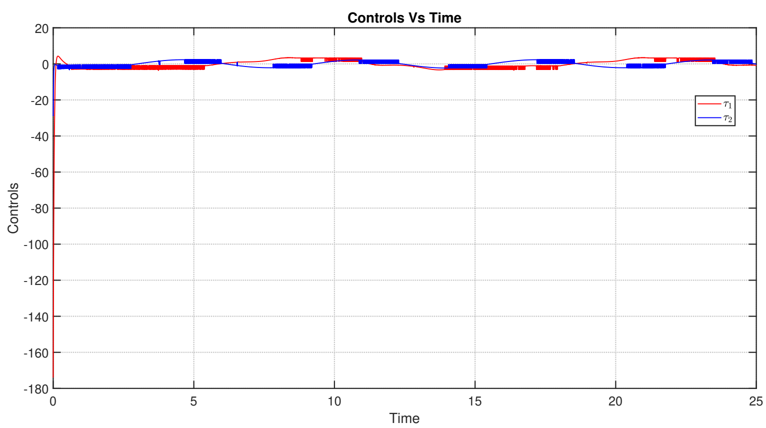

Figure 4: Control Inputs for Sliding-Mode NN-based Control

Maximum values of the torques are 174 Nm and 28.7207 Nm respectively.

3)Simulation Section for the RISE Neural Network Based Controller

1) Control Gains used:

- K_s multiplying e_2 in the RISE part of the control law
- α_1 multiplying e_1 in the definition of the error signal e_2
- α_2 multiplying e_2 in the definition of the error signal r
- β_1 multiplying $sgn(e_2)$ in the RISE part of the control law
- γ_1 and γ_2 are the learning rates in the update laws for the estimates of weight matrices ($\widehat{W}$ and $\widehat{V}$) in the neural network

The values of which were tuned as follows:

$K_s = \begin{bmatrix} 80 & 0 \\ 0 & 9 \end{bmatrix}$, $\alpha_1 = 3$,$\alpha_2 = 2$, $\beta_1 = 0.5$ $\gamma_1 = 500$ and $\gamma_2 = 1000$

NOTE 1: γ_1 and γ_2 can be chosen to be diagonal matrices of appropriate dimensions. If we don't wish to tune the diagonal elements separately, they can be chosen as scalar gains as shown above

NOTE 2: K_s been chosen to be diagonal matrix instead of a scalar constant to ensure that the control torques are deliverable by the motors while getting a good performance. (The maximum torque for motor 1 is 250 Nm, while the maximum torque for motor 2 is 30 Nm)

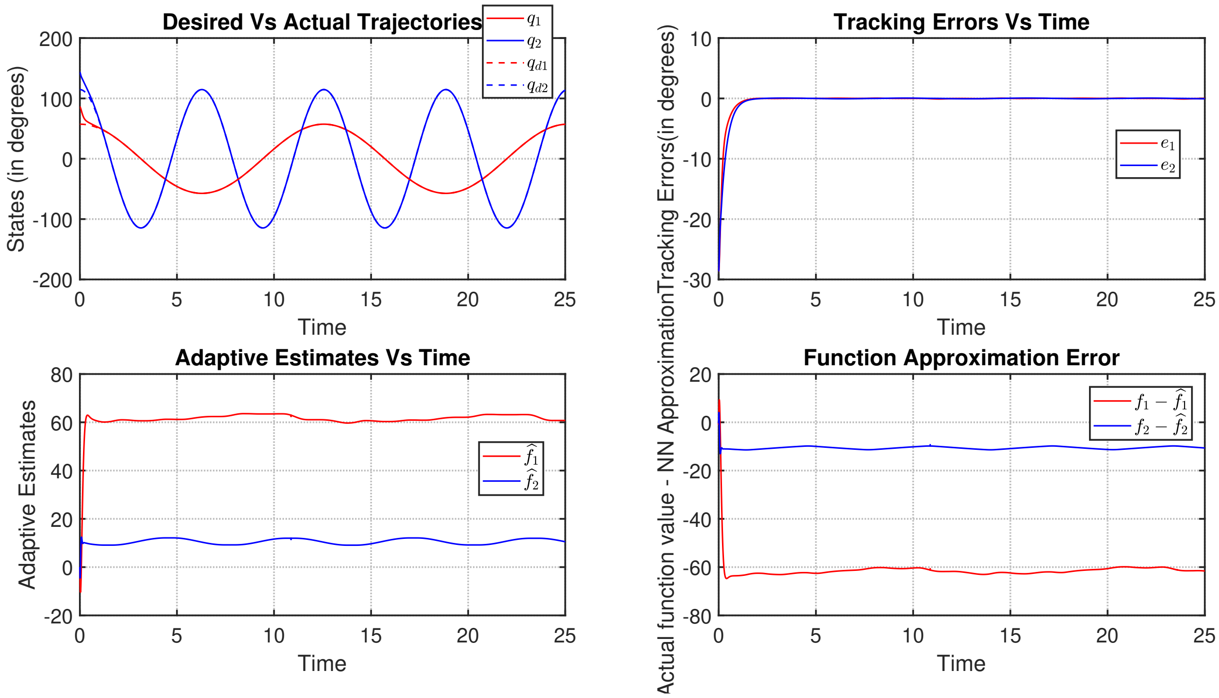

Figure 5: RISE NN-based Control

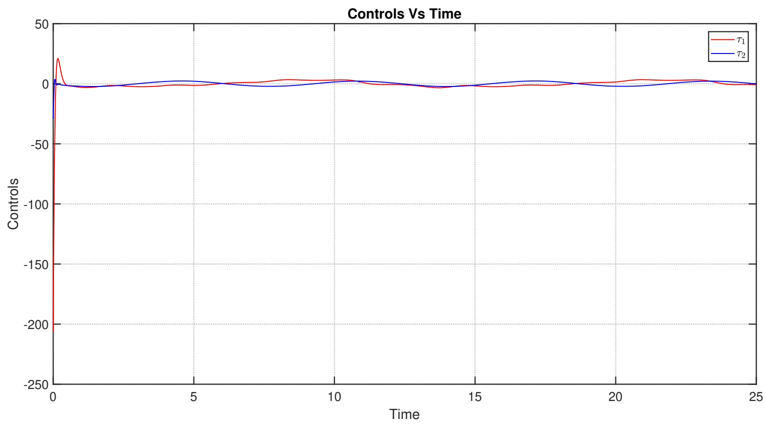

Figure 6: Control Inputs for RISE NN-based Control

Maximum values of the torques are 206 Nm and 28.8722 Nm respectively.

4)Discussion Section

- Increasing the gains K, K_n or K_s in any of the controllers increases the speed of convergence of the tracking error to zero. However, this comes at the cost of a bigger control effort.The gains are therefore fixed based on the maximum control torques that the motors are capable of delivering.

- A higher value of K_s in the sliding mode based controller (multiplies the $sgn(.)$ function) would mean a control signal which is more discontinuous (higher jumps).

- However, a higher value of β_1 in the RISE controller, although multiplying a $sgn(.)$ function, does not contribute to control discontinuity since it's integral appears in the control law.

- ***Tracking-Error Convergence:*** We have a Uniformly Ultimately Bounded result with the Continuous NN-based controller. The tracking error comes close to zero and keeps oscillating between some bounds. How close to zero and how small the oscillations are, is determined by our choice of gains.
  We have an asymptotic convergence of the tracking error with sliding mode based controller. However the control input is high-frequency (discontinuous).
  With the RISE based controller, we have an asymptotic convergence achieved with a continuous control input.
  Speeds of convergence in all three are higher with bigger gains.

- ***Convergence of the function approximation*** By the universal function approximation property, the NN function approximation must converge to the actual function that is being approximated within some error margin. While, this is seen to happen in the above simulations for the continuous and sliding mode controllers, for the RISE based controller the approximation error remains bounded, but not very close to zero. I believe the gains need more tuning to ensure a better convergence.

It can be seen from the update law for $\widehat{W}$ and $\widehat{V}$ that once the tracking error goes very close to zero, these estimates become almost constant. So , a downside of choosing high K gains to drive the tracking error to zero very quickly is that the Neural Network does not get enough time to learn. In other words, the function approximation does not come very close to the actual function value.

Difficulty : These controllers are easier to code compared to model-based controllers because of absence of regression matrices. However, they take more time to run since the controller has to learn more about the system as compared to model-based controllers.