

Simulation Project 6

EML 6351

by

Aahlad Vanam

13681791

**Dynamic model:**

This is in the form of -----------------------(1)

(b) Problem definition and open-loop error system development:

Open Loop error system:

**SIMULATION:**

Neural Network-based controller with a typical CONTINUOUS FEEDBACK control law for the dynamics:

**Gains:**

k = 50;

kn = 10;

ks = 1;

alpha = 10;

gamma1 = 5;

gamma2 = 10;

proj1 = 4.5; %vHat Saturation

proj2 = 6.7; %wHat Saturation

**Controller:**









**Controller 2:**

**Neural network-based controller with a discontinuous sliding mode feedback control law**

Gains:

k = 50;

kn = 10;

ks = 1;

proj1 = 4.5; %vHat Saturation

proj2 = 6.7; %wHat Saturation

alpha = 10;

gamma1 = 5;

gamma2 = 10;

Controller:









Controller 3:

**Neural network-based controller with a continuous RISE feedback control law for the dynamics:**

**Gains:**

k = 50;

kn = 10;

ks = 1;

kr = 3;

alpha = 10;

gamma1 = 5;

gamma2 = 10;

beta1 = 2;

proj1 = 4.5; %vHat Saturation

proj2 = 6.7; %wHat Saturation

**Controller:**









**DISCUSSION:**

1. Projection gains had no impact on first two controllers as they were already bounded. However, for RISE controller we see that having lower gains (limits) tightens the graph.
2. Tuning adaptation gains had the most effect on the convergence of errors, control gains were tuned in order to improve the performance of the controllers.
3. Position errors convergence performance is highest in RISE Neural Network controller and least in Standard one. However, the performance doesn’t vary significantly.

Continuous high gain NN controller



NN controller with sliding mode control law



NN controller with RISE control law 

1. Position errors in Link 1 converge fastest in RISE NN controller at 2.1s, whereas it is 2.4s for Sliding mode NN controller and 4s for standard NN controller
2. Position errors in Link 2 converge at almost same time in all controllers, however, the standard NN bounces around zero for a while where as RISE NN controller converges and remains there.
3. Computational effort for RISE is slightly more than standard one but the errors perform much better, so it is a worthy trade-off.
4. Controllers remain bounded within the given limits.
5. To achieve this performance, all three controllers use 0.4% of Max. Torque for Link 1 and 17% for Link 2
6. In addition to that, we can see the blocks in Link 2 controller for Sliding mode controller, due to the addition of a discontinuous controller. However, these discontinuities don’t cause spikes or blow out of bounds, so our performance is not affected.
7. graphs are similar for first two controllers, but we can notice the difference in the third controller. This role is played by the projection function which saturated the estimates and we see many straight lines
8. We use a fixed step ode3 solver. To saturate with projection function we needed the time step ‘dt’ which needed a fixed value instead of a variable one.

In addition to that, ode3 finished the simulations faster than ode45

**Ease of implementation:** Compared to previous simulations where the regression matrices were manually typed in, this one doesn’t need one. However, Design of Neural network takes the similar amount of time.

**Tuning:** As we are saturating our adaptive update laws with projection functions they don’t blow up and we can tune our controllers faster and with higher precision.