**Homework 6**:

The state space model of a brushless DC motor is:

where:

(angular velocity) (input voltage)

Parameters:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  | (motor viscous friction constant) |  |

**Solution:**

1. **Deriving Transfer Function of plant**

I derived transfer function to control plant with fuzzy controller:

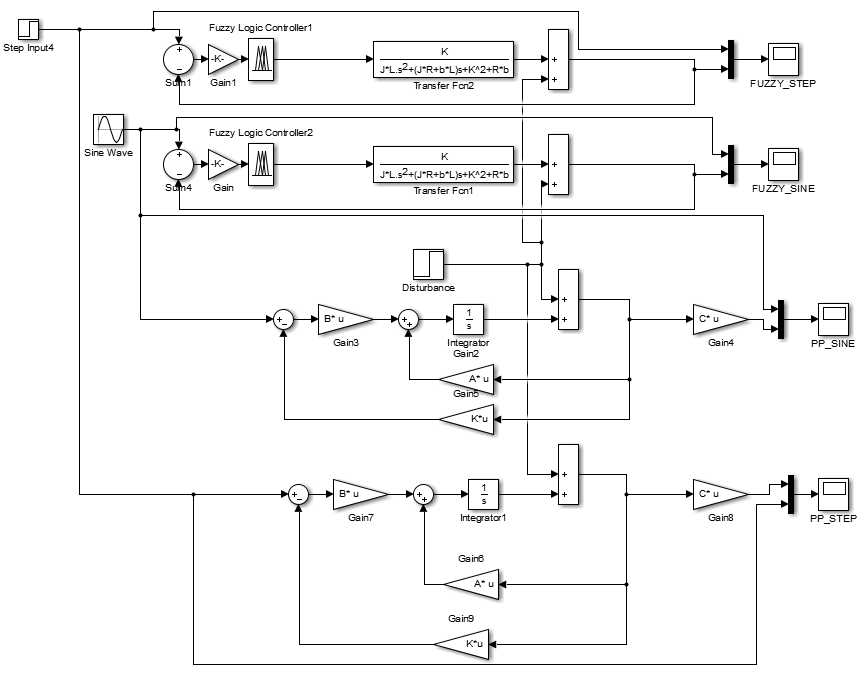
Following governing equations are given:

Since , that can be rewritten

Applying Laplace transform:

Open-loop transfer function:

1. **Simulink Model**



To compare controllers I designed model to test models with step reference and sine wave reference signals. I also used step signal block to model disturbance.

1. **Pole-Placement – choosing poles**

I experimented with different poles to derive optimal control with pole placement method. In the end I used LQR regulator to see optimal poles according to LQR, those were:

p1 = -3.1315 + 0.8256i;

p2 = -3.1315 - 0.8256i;

Gain K can be calculated in Matlab using:

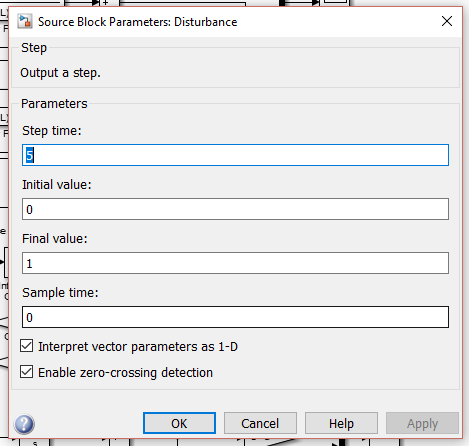
K = lqr(A,B,[1 0; 0 1],1);

Or

K = place(A,B,[p1 p2]);

Depending on choice of either using LQR or simply choosing poles.

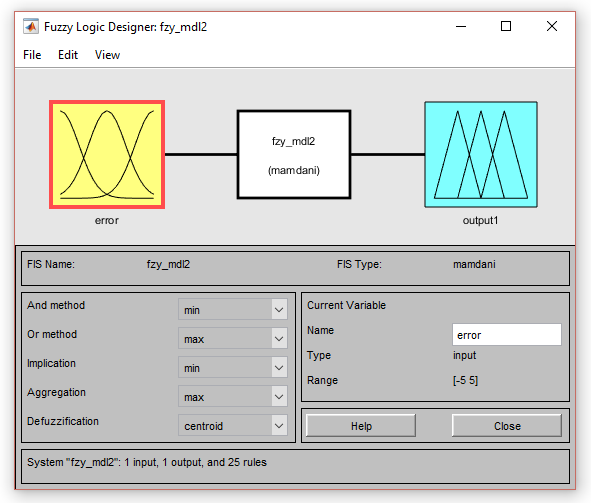
1. **Disturbance**



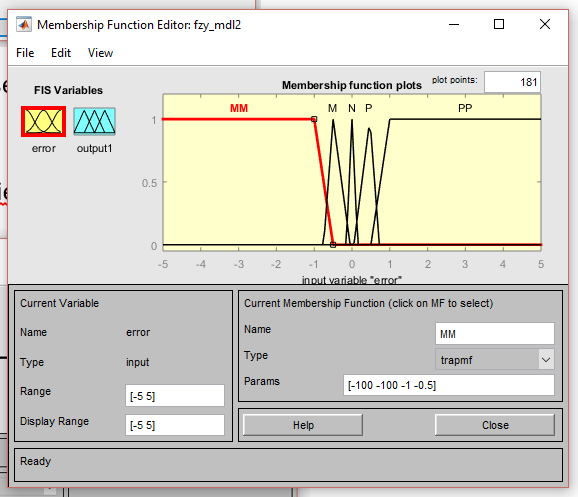
For external disturbance model I simply used step signal at 5 s with final value 1.

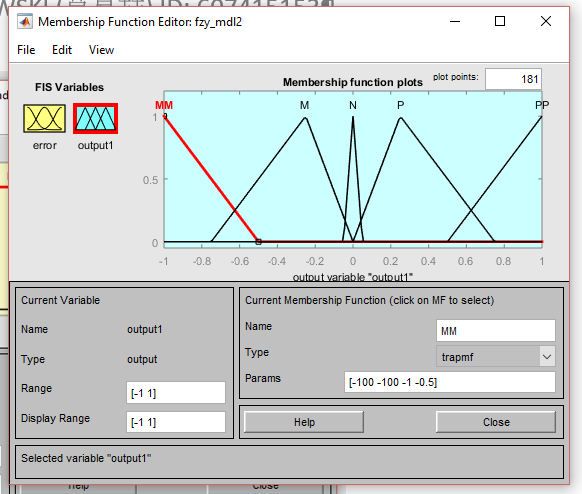
1. **Fuzzy rules**

I configured fuzzy controller using >>ruleview mam21 to create fuzzy logics file .fis



I relied on single input – error.





|  |  |  |
| --- | --- | --- |
| Error |  | Output |
| MM |  | MM |
| M |  | M |
| N |  | N |
| P |  | P |
| PP |  | PP |

(Rules)

1. **Matlab routine**

I wrote a Matlab routine to initialize parameters for Simulink model:

%parameters

R=2.0;

Km=0.1;

J=0.02;

L=0.5;

Kb=0.1;

Kf = 0.01;

K=Kb;

b=Kf;

%gain for fuzzy controler

gain = 10;

%state-space model

A=[-R/L -Kb/L; Km/J -Kf/J];

B=[1/L; 0];

C=[0 1];

D=[0];

%poles for pole-placement

p1 = -3.1315 + 0.8256i;

p2 = -3.1315 - 0.8256i;

%calculate gain for pole-placement

K\_PP = place(A,B,[p1 p2]);

%read fuzzy logics model for controller

fzy\_mdl = readfis('fzy\_mdl');

1. **Adding model uncertainty**

After initialization I overwrote parameters when analyzing the robustness in case of model uncertainty.

%ADD UNCERTAINITY TO PARAMETERS

R=R+(rand()-0.5)\*R;

Km=Km+(rand()-0.5)\*Km;

J=J+(rand()-0.5)\*J;

L=L+(rand()-0.5)\*L;

Kb=Kb+(rand()-0.5)\*Kb;

Kf = Kf+(rand()-0.5)\*Kf;

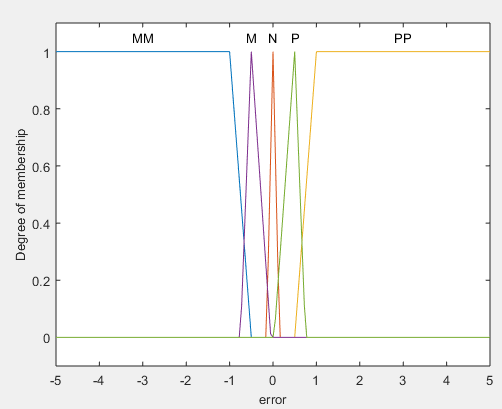
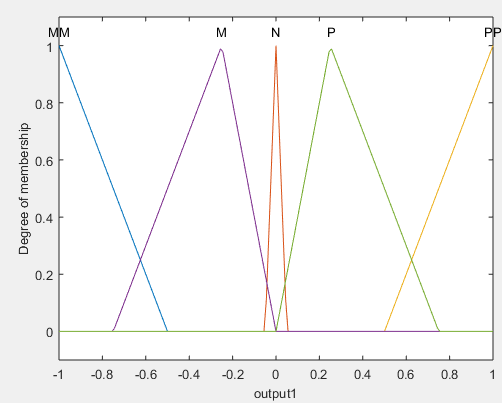
Kb = Kb+(rand()-0.5)\*Kb;

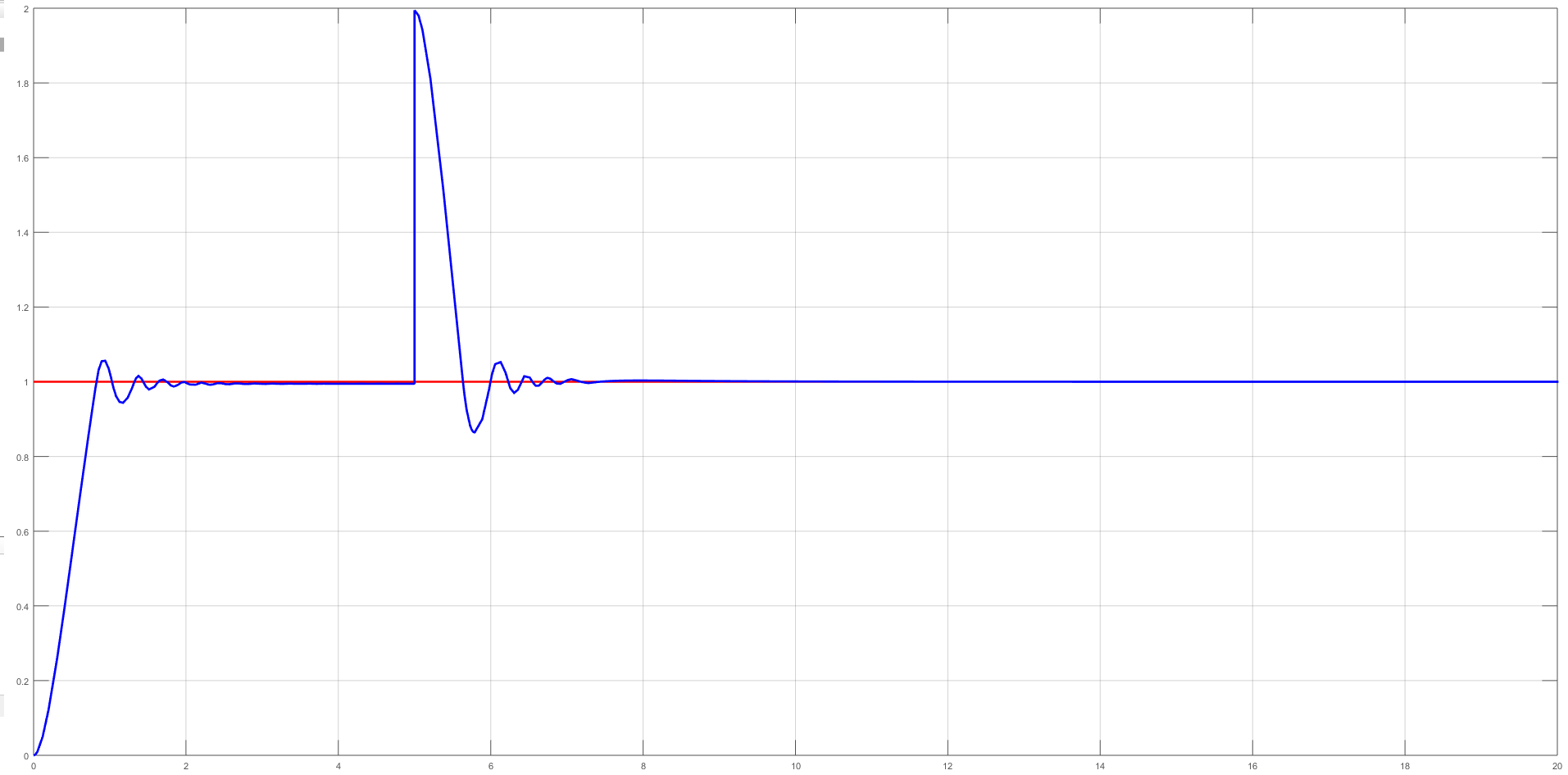
K=Kb;

Kf=Kf+(rand()-0.5)\*Kf;

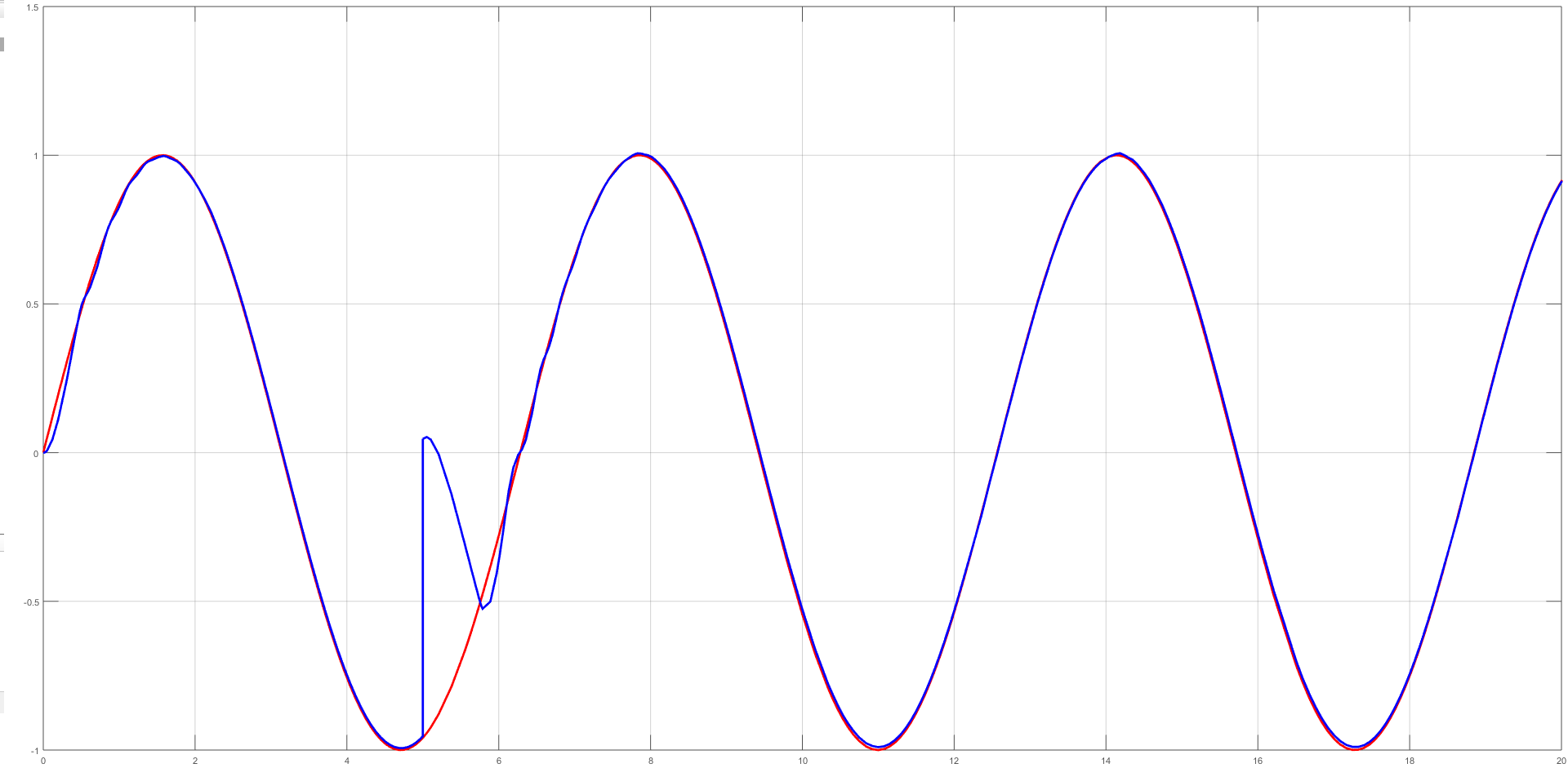
b=Kf;

1. **Results**

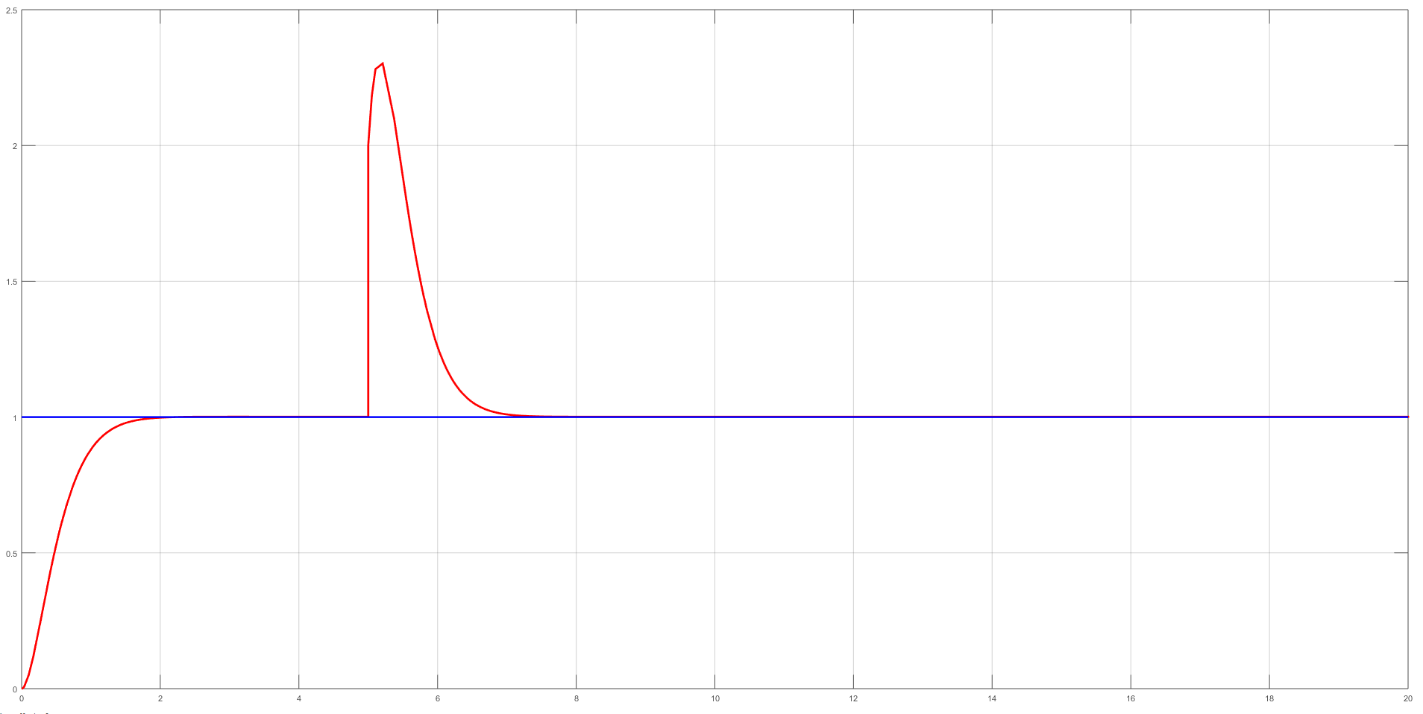
** **

****

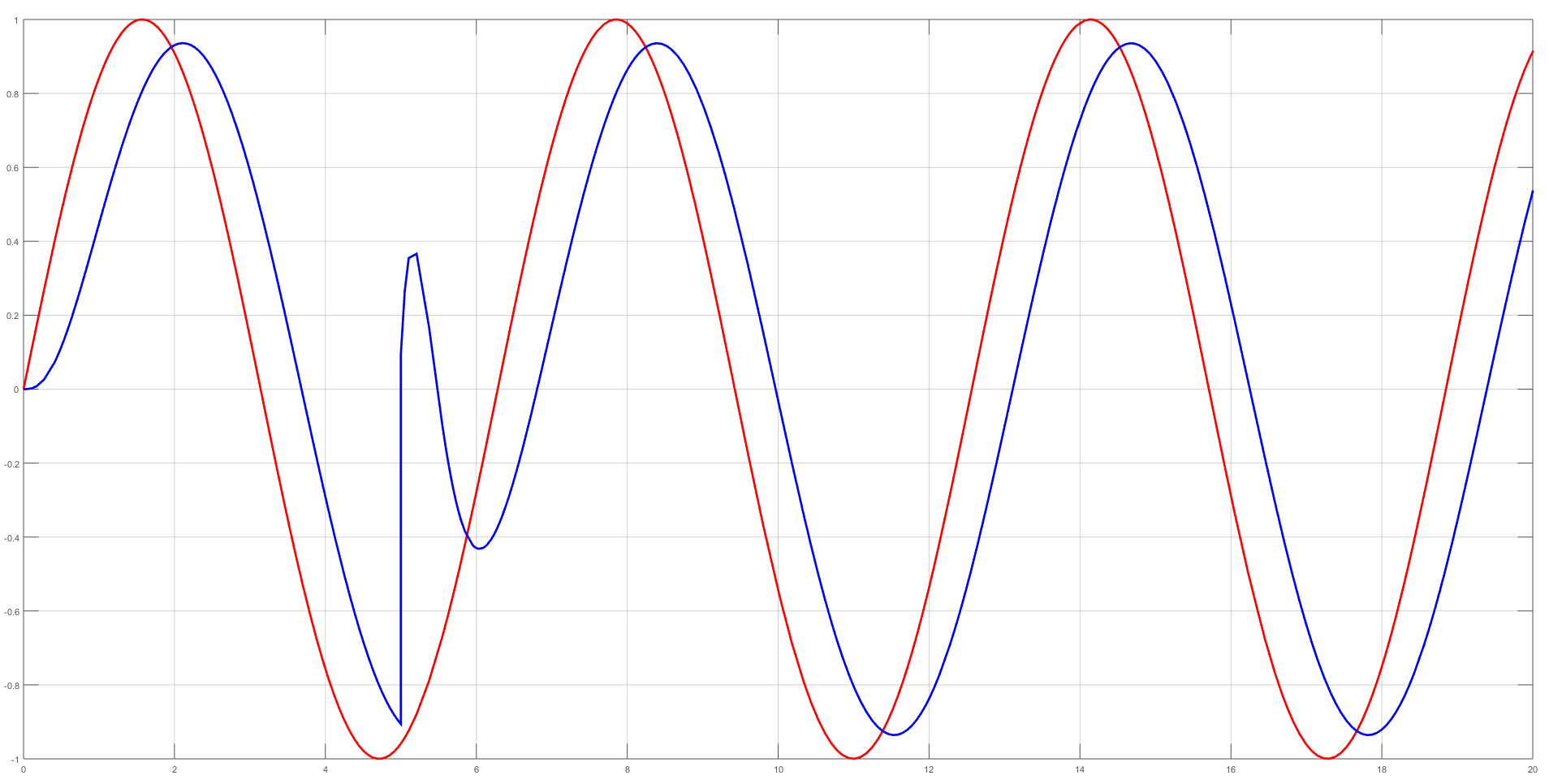
**FUZZY CONTROLLER, STEP RESPONSE, DISTURBANCE AT 5s**

****

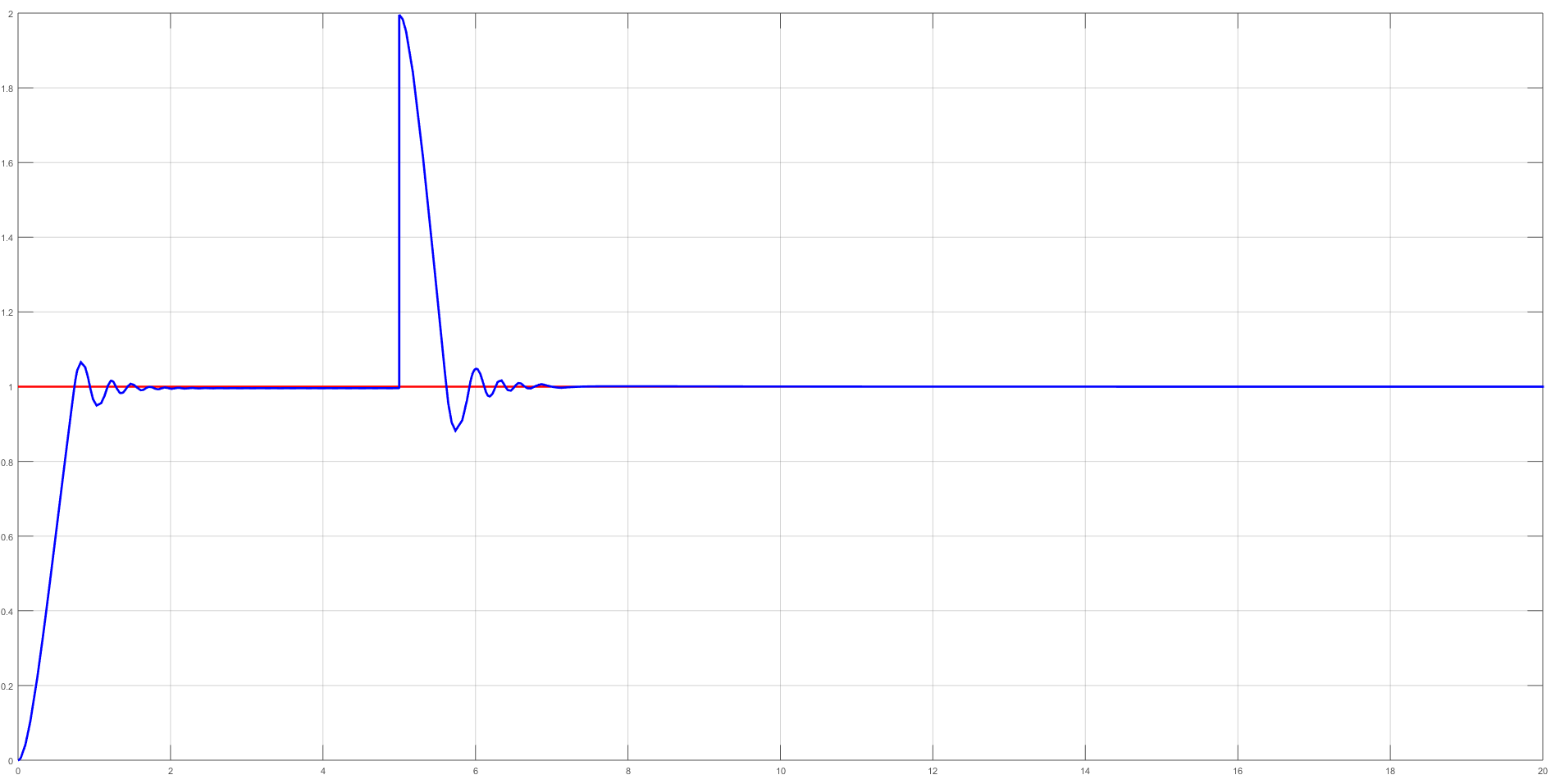
**FUZZY CONTROLLER, SINE WAVE TRACKING, DISTURBANCE AT 5s**

****

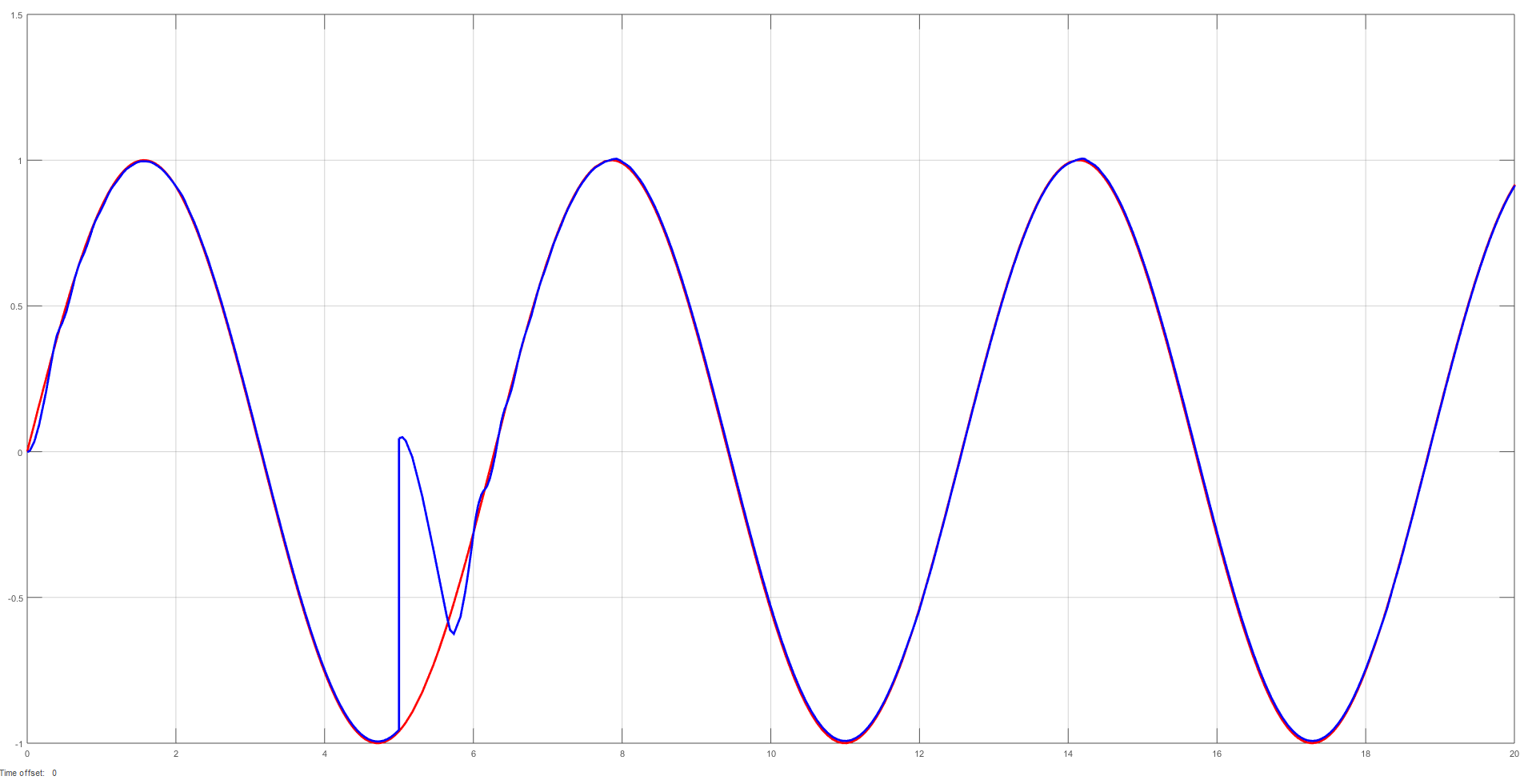
**POLE-PLACEMENT, STEP RESPONSE, DISTURBANCE AT 5s**

****

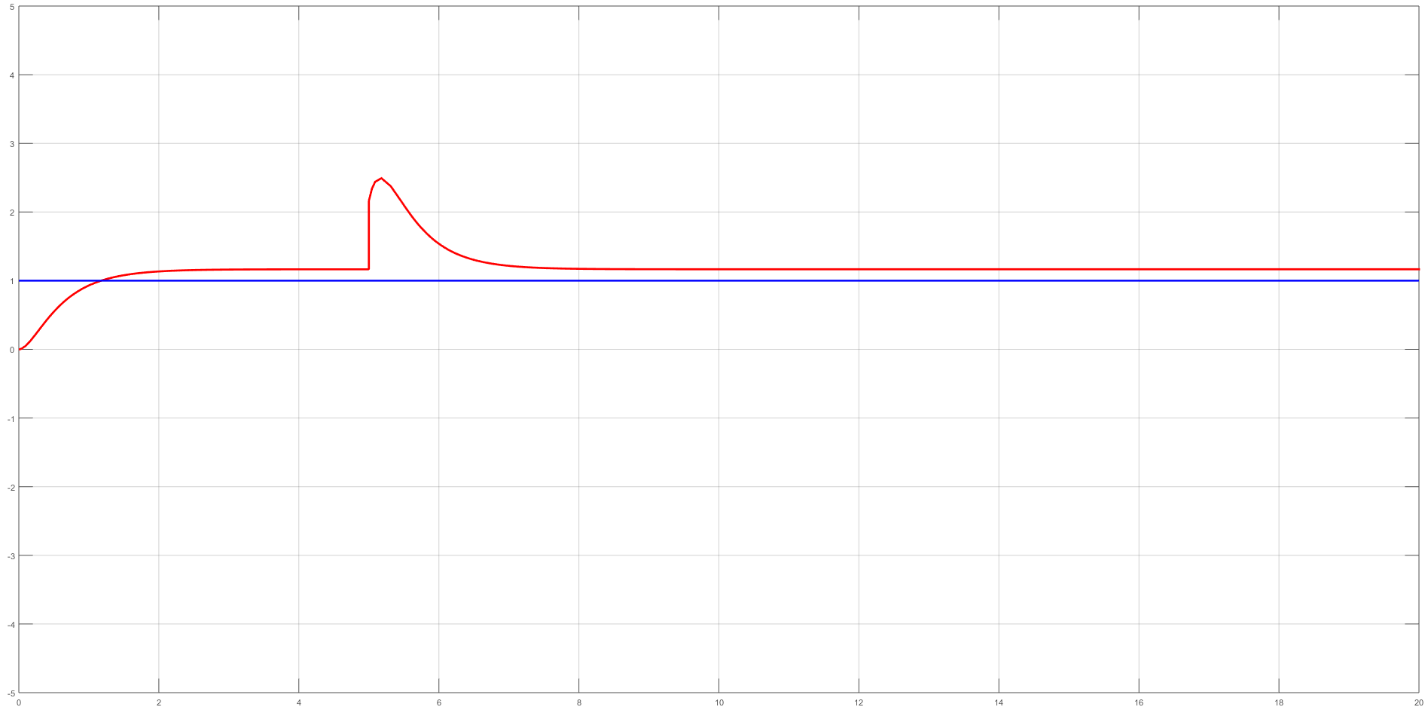
**POLE-PLACEMENT, SINE WAVE TRACKING, DISTURBANCE AT 5s**

****

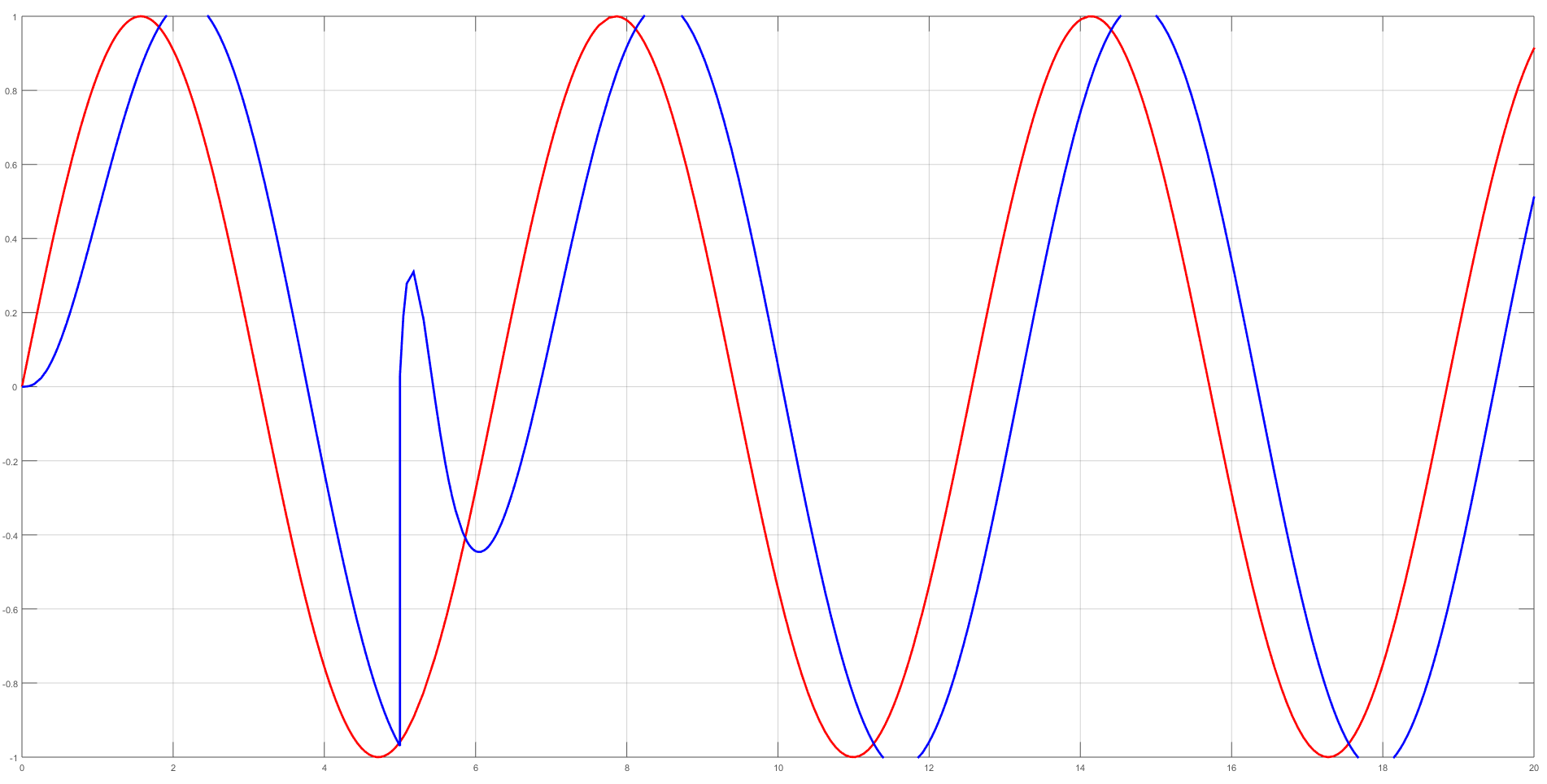
**FUZZY CONTROLLER, STEP RESPONSE, DISTURBANCE AT 5s, MODEL UNCERTAINITY**

****

**FUZZY CONTROLLER, SINE WAVE TRACKING, DISTURBANCE AT 5s, MODEL UNCERTAINITY**

****

**POLE-PLACEMENT, STEP RESPONSE, DISTURBANCE AT 5s, MODEL UNCERTAINITY**

****

**POLE-PLACEMENT, SINE WAVE TRACKING, DISTURBANCE AT 5s, MODEL UNCERTAINITY**

1. **Comparison**

* Sine tracking

I was unable to tune the design using pole-placement method to track sine wave effectively. Not only amplitude of output was smaller by 5% than reference signal amplitude but most importantly there was a lag in tracking sine wave. Practically, this design was unable to track sine wave. Increasing signal frequency yielded even more inferior results.

On the other hand, fuzzy controller was able to track sine wave effectively. There was small oscillation at all times because of the simple design of fuzzy controller. I believe with more practice one could design fuzzy rules more effectively to minimize this behavior.

* Disturbance at t=5s:

The overshoot in step response was smaller for fuzzy controller. Settling time was similar. However, as for the fuzzy controller there were oscillation because of simple design of the rules.

* Step response:

Settling time was similar (around t=2s) for both controllers. However, fuzzy control induced oscillations in the response.

* Uncertainty

Fuzzy controller was robust to model uncertainty. Responses in case of model uncertainty were virtually the same as for certain model. Pole-placement method was not robust to model uncertainty.