Table of Contents

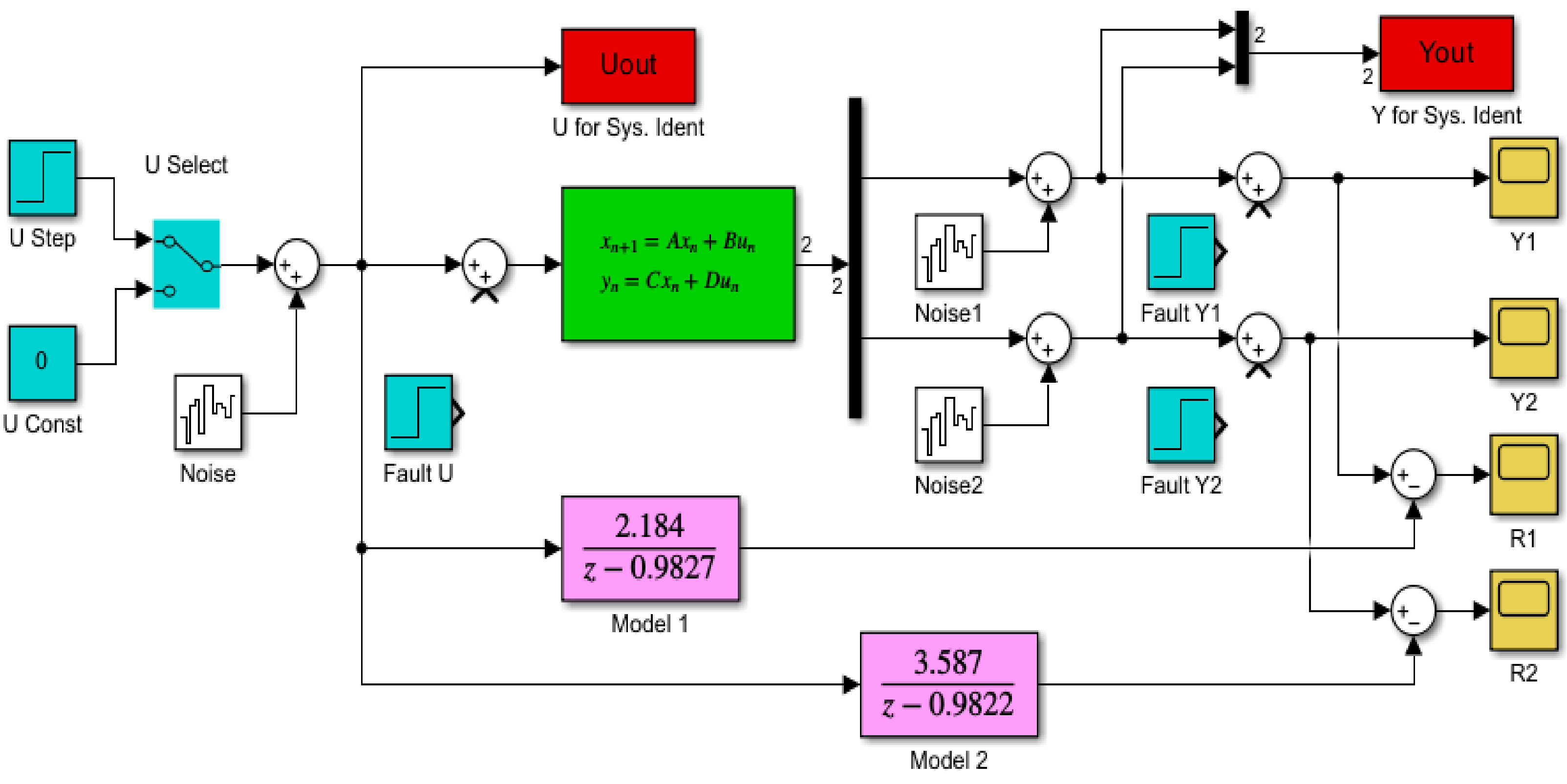
Introduction

A fault represents a deviation from the normal behavior of a system due to unexpected events, the occurrence of faults must be detected as early as possible to prevent any serious consequences. For this purpose, Fault detection and diagnosis techniques (FDD) are used for detection of occurrence of faults (fault detection) and the localization of detected faults (fault isolation). Most contributions in FDD rely on the analytical redundancy principle, the procedure of using model information to generate additional signals which are then compared with the plant measurements [1]. Output estimation is a powerful tool for building mathematical models of systems.

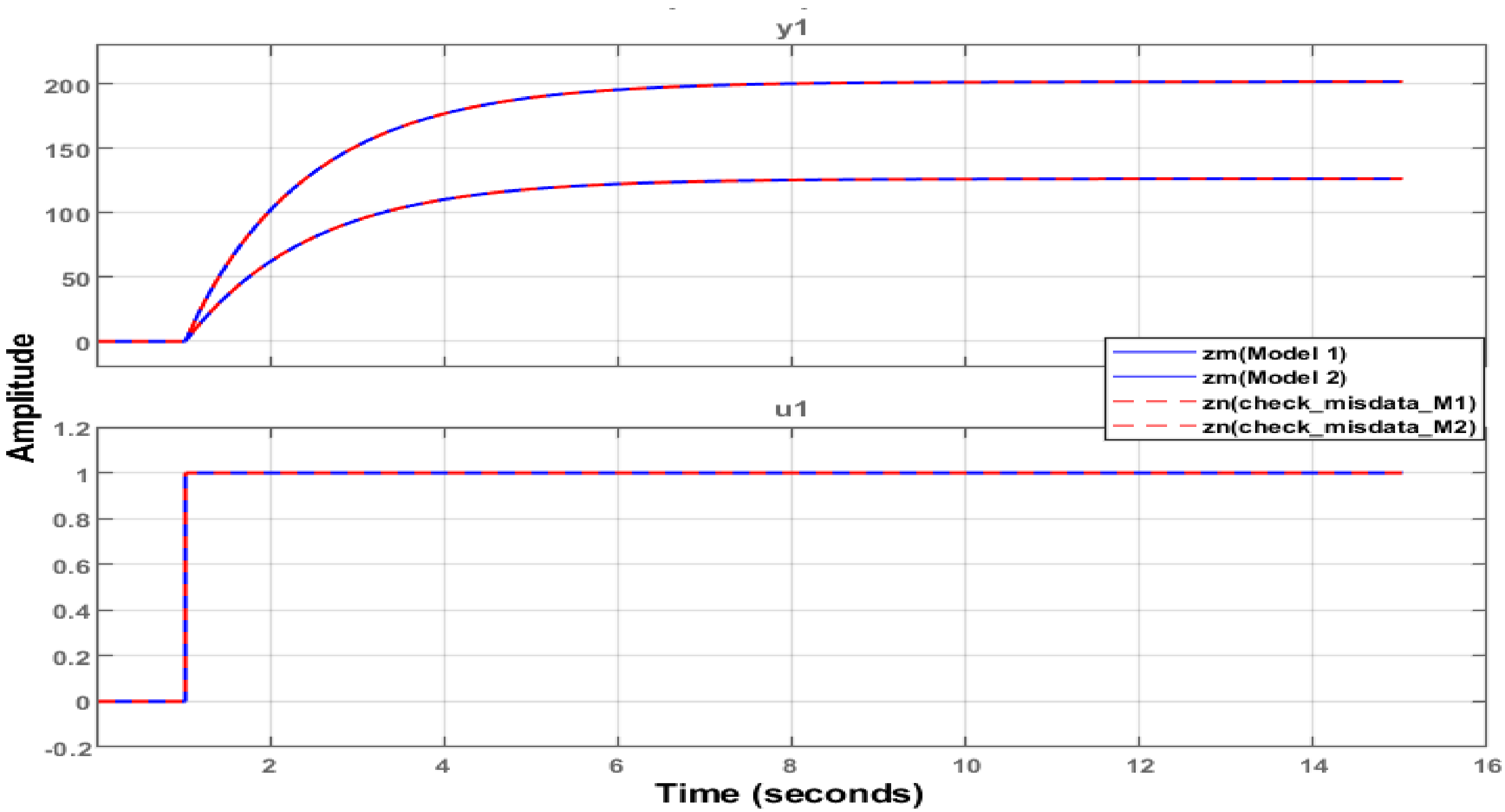
To build these models, I'm simulating the given discrete state-space plant (single-input and two-outputs model) in Matlab and Simulink. Simulations are performed on the plant for 15 seconds with a sampling rate of 40 Hz (0.025 sec) also a unit step impulse and white noise are added to simulate the real environment because there will always be some noise and disturbances and some error in sensor measurements.

Using output estimation function **"oe"** in MATLAB [2]we obtain models as shown in figure 1, so from the real plant we estimate two first order mathematical models of our system. We move from this difference equations to discrete transfer functions. The **oe** function estimates transfer function Model 1, Model 2 with an of order 1. The estimated models are stored as m1&m2. Once we get our estimated mathematical model of the system and its parameters, we can use this models to predict the output and behavior of the system for new input signals.

SIMULINK DIAGRAM OF REAL PLANT AND ESTIMATED MODEL M1 AND M2



STEP RESPONSE OF INPUT-OUTPUT DATA



Check for missing data, measurement errors and outliers

In order to perform FDD effectively it is important to check for missing data, measurement errors, disturbances and outliers in the measured data.

Missing data can lead to incorrect conclusion of the behavior of the system, so we need to account for its absence in FDD algorithm. In Matlab ‘misdata’ function reconstructs the input and output time domain data set by reasonable estimates. It minimizes the output prediction error based on current reconstruction.

Outliers are data points that are significantly different from expected system behavior. It’s important to carefully analyze the data and look for patterns/trends that are inconsistent with the expected behavior of the system, this can be caused by many factors like measurement errors, this leads to incorrect and noisy data, we can filter the data but that reduces the speed so we should account for this and correct the errors before performing FDD.

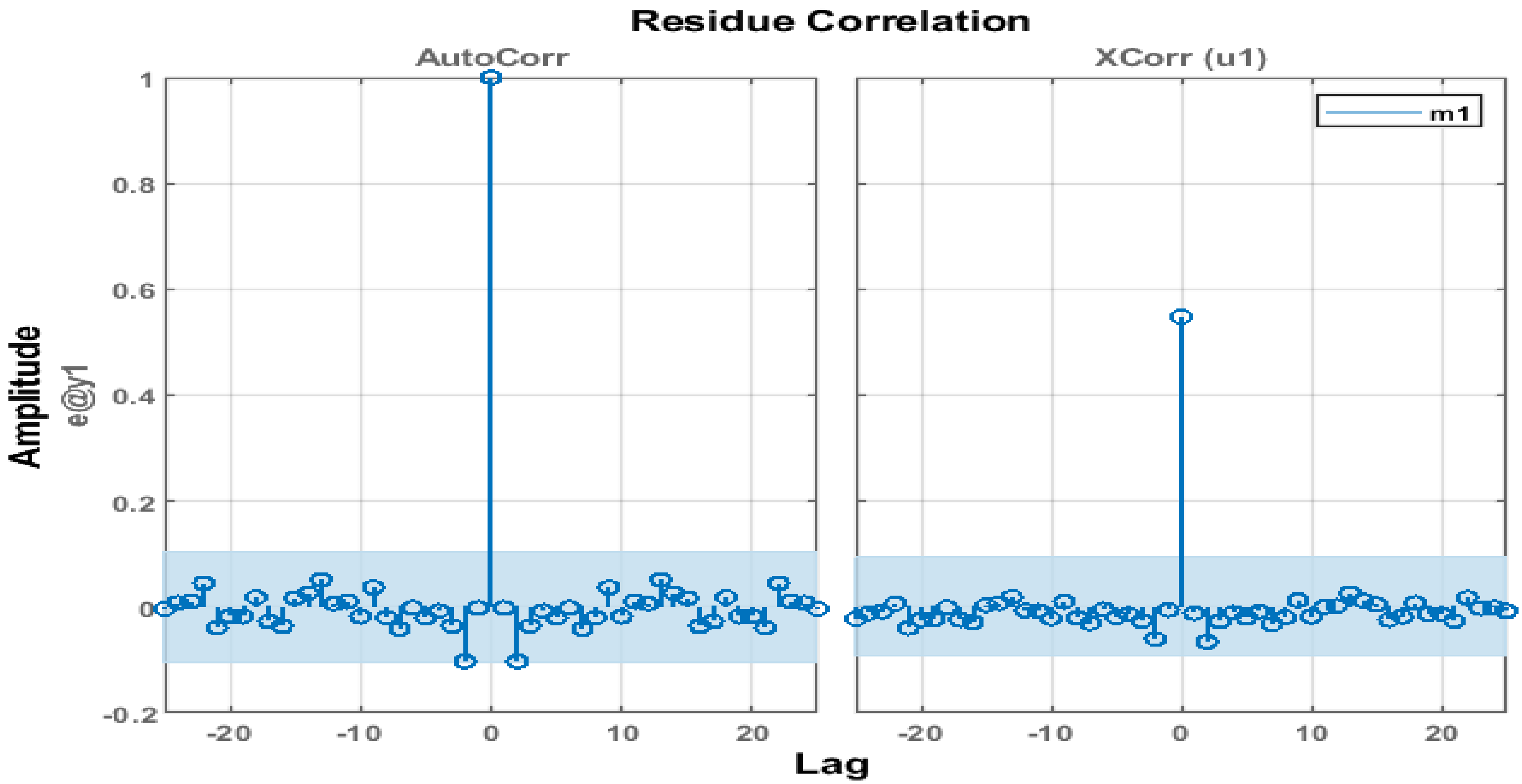
By examining the residual correlational structures (using “resid” function in Matlab) , [1] and identifying any anomalies, outliers it is possible to develop FDD algorithms that are Robust and accurate and can effectively identify faults and abnormal conditions in the system.

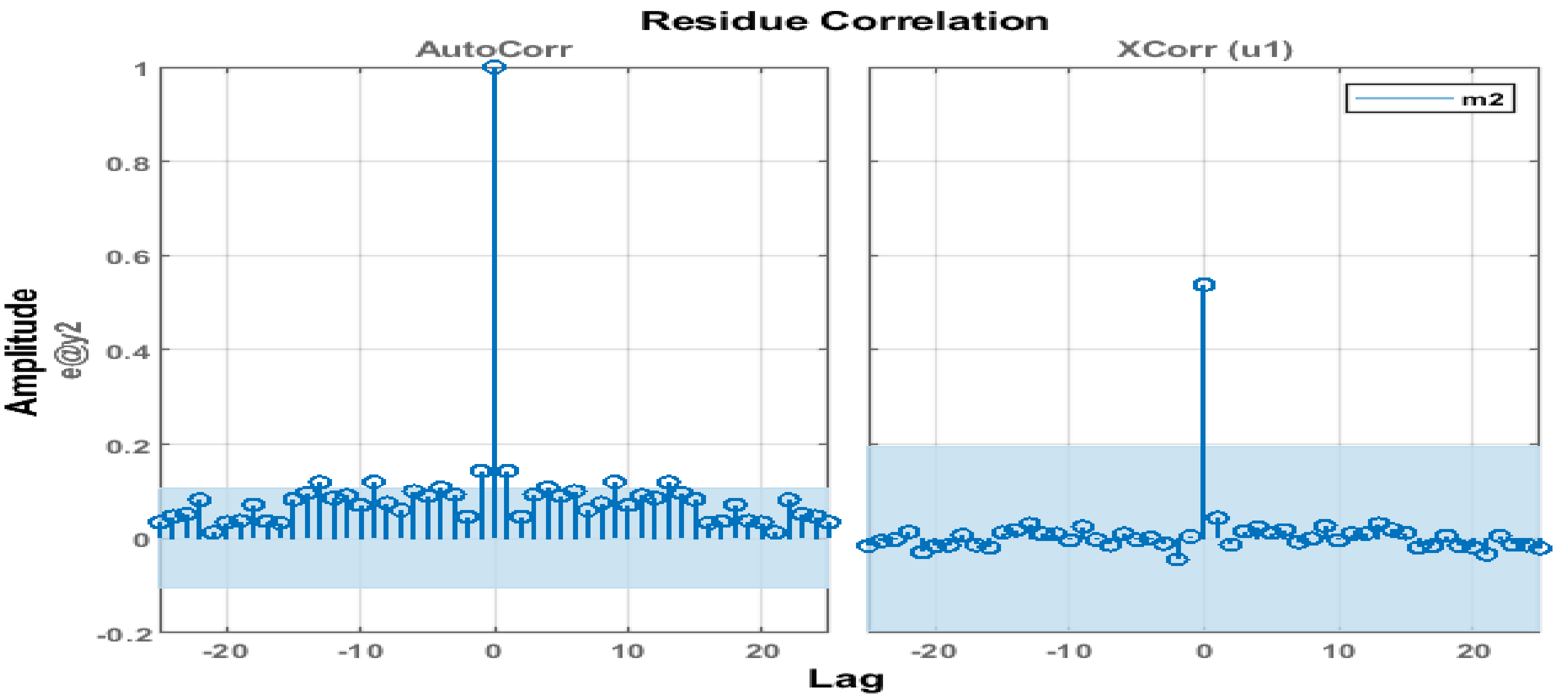
Residual Analysis and Correlation

The main component of any FDD system is the residual generator which produces residual signals grouped in a vector 'R' by processing the available measurements 'y' and the known values of control inputs 'u'.

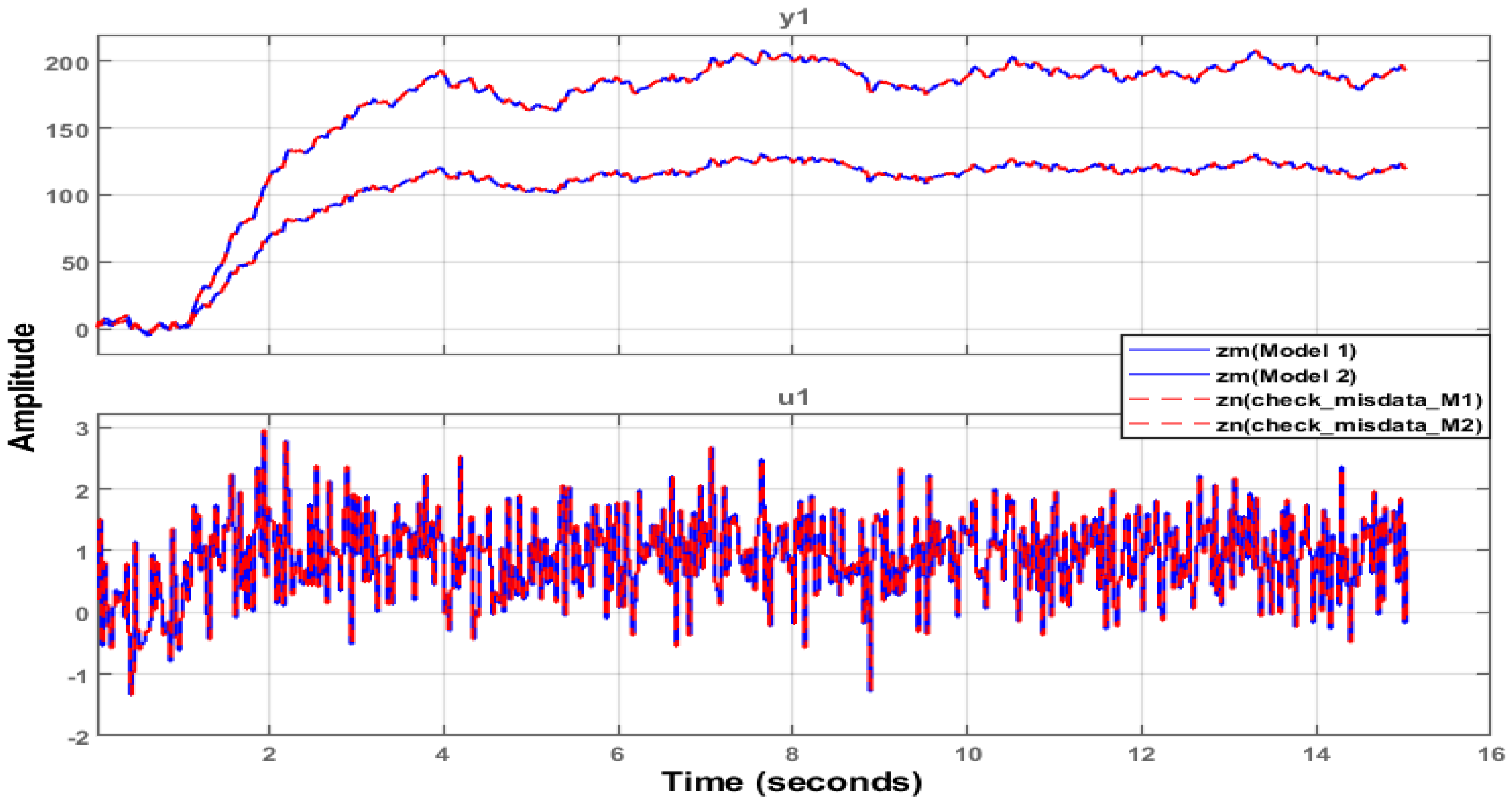
Residual analysis is an important step in the developing and validating the models, using **'resid' (Compute and test residuals)** function in Matlab, computes the 1-step-ahead prediction errors (residuals) for an identified model, We plot the AutoCorr (shows correlation of residual with themselves) and XCorr (shows the correlation of residuals with input and output data over time). These plots assess the quality of residual correlation structure and adequacy of the models. The residuals must have small magnitude, zero mean, Low autocorrelation, except at zero-time lag [2].

These plots are powerful tools for diagnosing model fit and identifying potential areas for improvement. Upon examining this plots do not indicate a mis specified model, missing variables, or unaccounted disturbances. Also, the cross-correlation between residuals of model 1,2 and the inputs lie in the 99% confidence band for all lags.





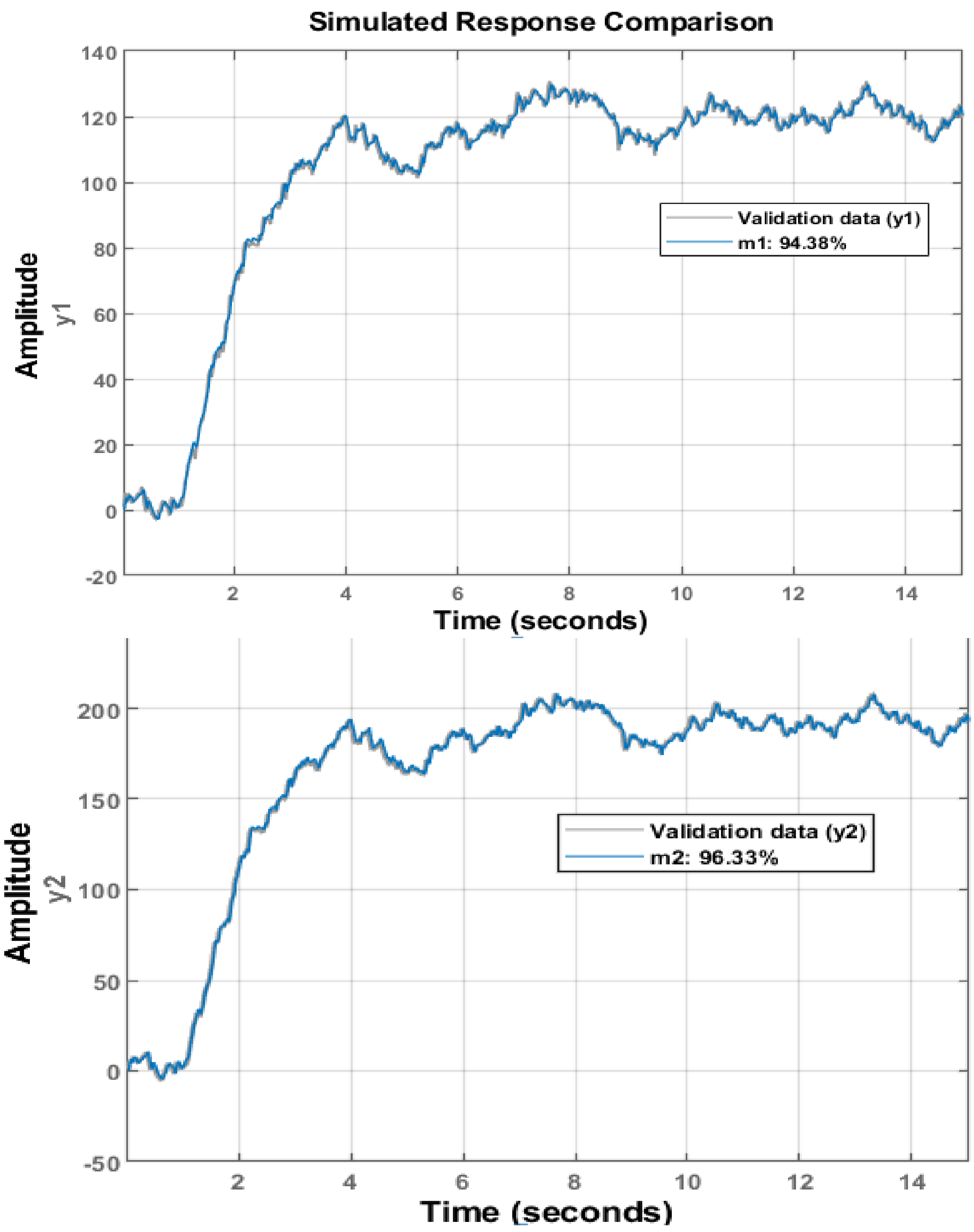
MODEL RESPONSES WITH NO FAULTS AND NOISE POWER 0.01



Comparing and validating the system responses

To compare the simulated output from the two models with the actual output we use the **'compare'** function.

The comparison plot indicates that both model m1 and m2 are good-fit and practically identical to the real plant, hence simulations of our model are considered valid.



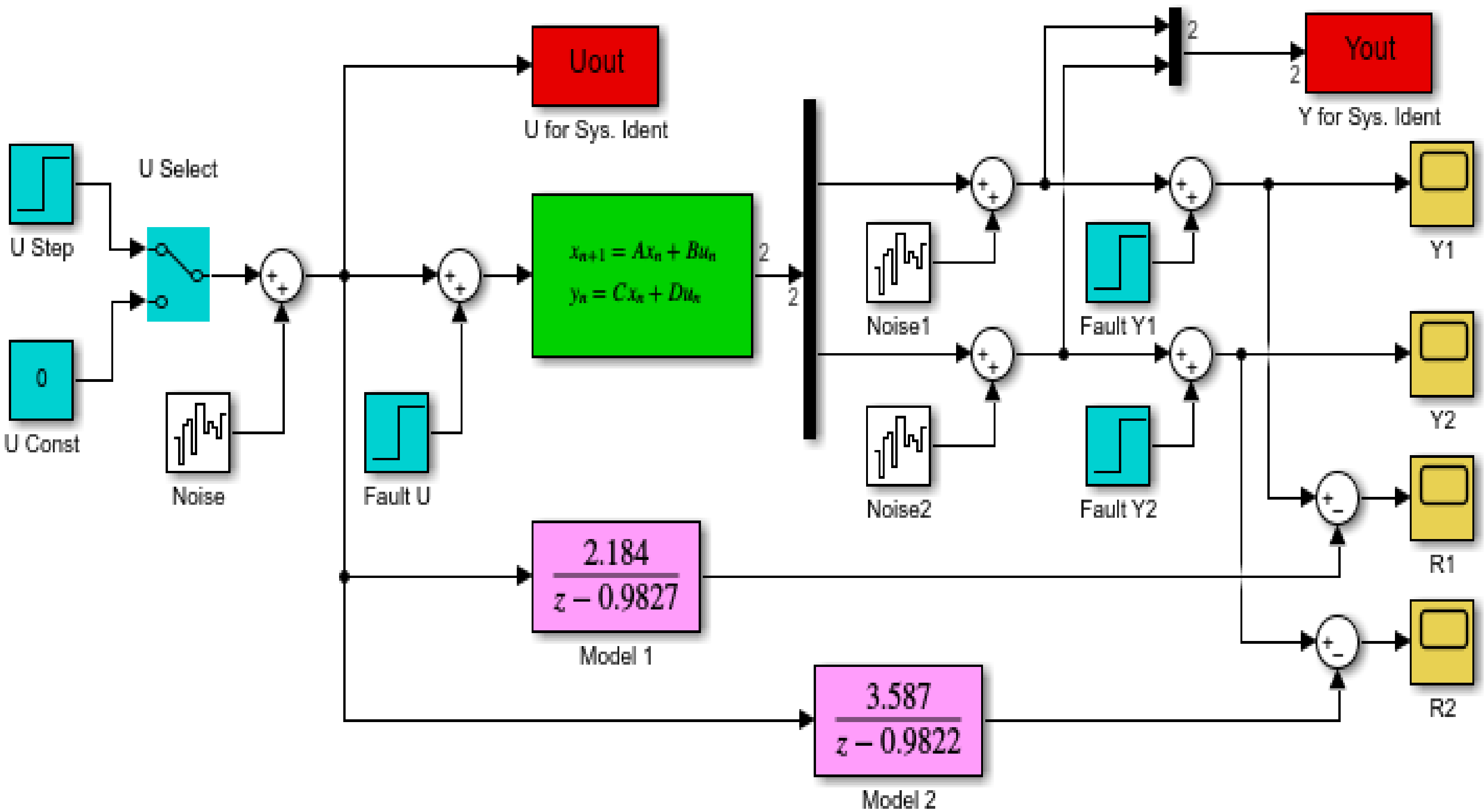
Parity Relation Scheme

Model-based fault detection and isolation (FDI) using parity relations is a commonly used approach for detecting and isolating faults in dynamic systems. Parity relations are mathematical equations that relate the system inputs, outputs, and states to each other. In this approach, a mathematical model of the system is developed, and parity relations are derived based on the model [3]. The basic idea is to use the redundancy in the system, by comparing the outputs predicted by the model 1,2 with the actual outputs measured by the sensors. If there is a discrepancy between the predicted and measured outputs, it is an indication of a fault.

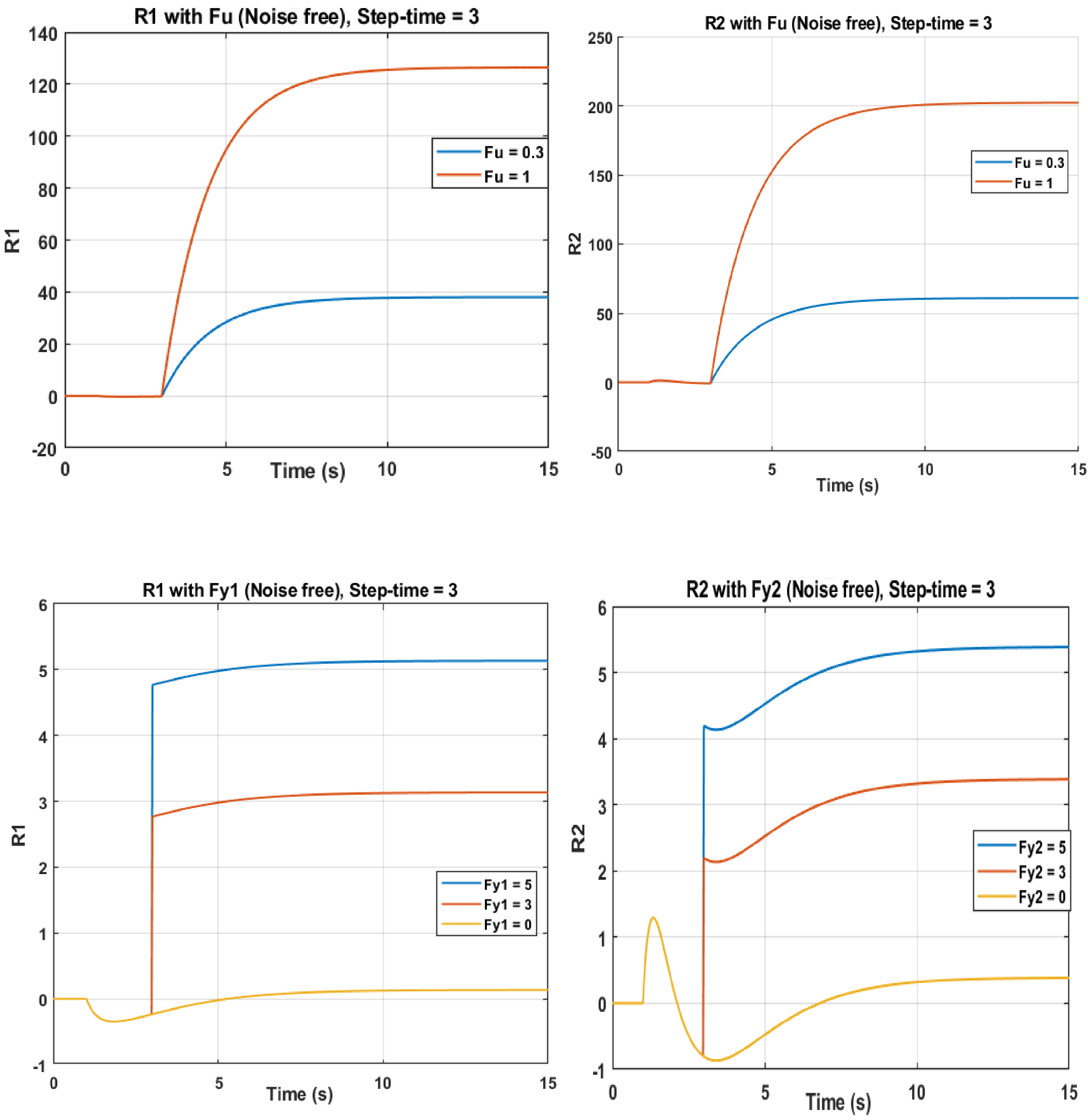
The role of the residual signals is to indicate the presence or absence of faults, and therefore the residual r must be equal (or close) to zero in the absence of faults and significantly different from zero after a fault occurs.

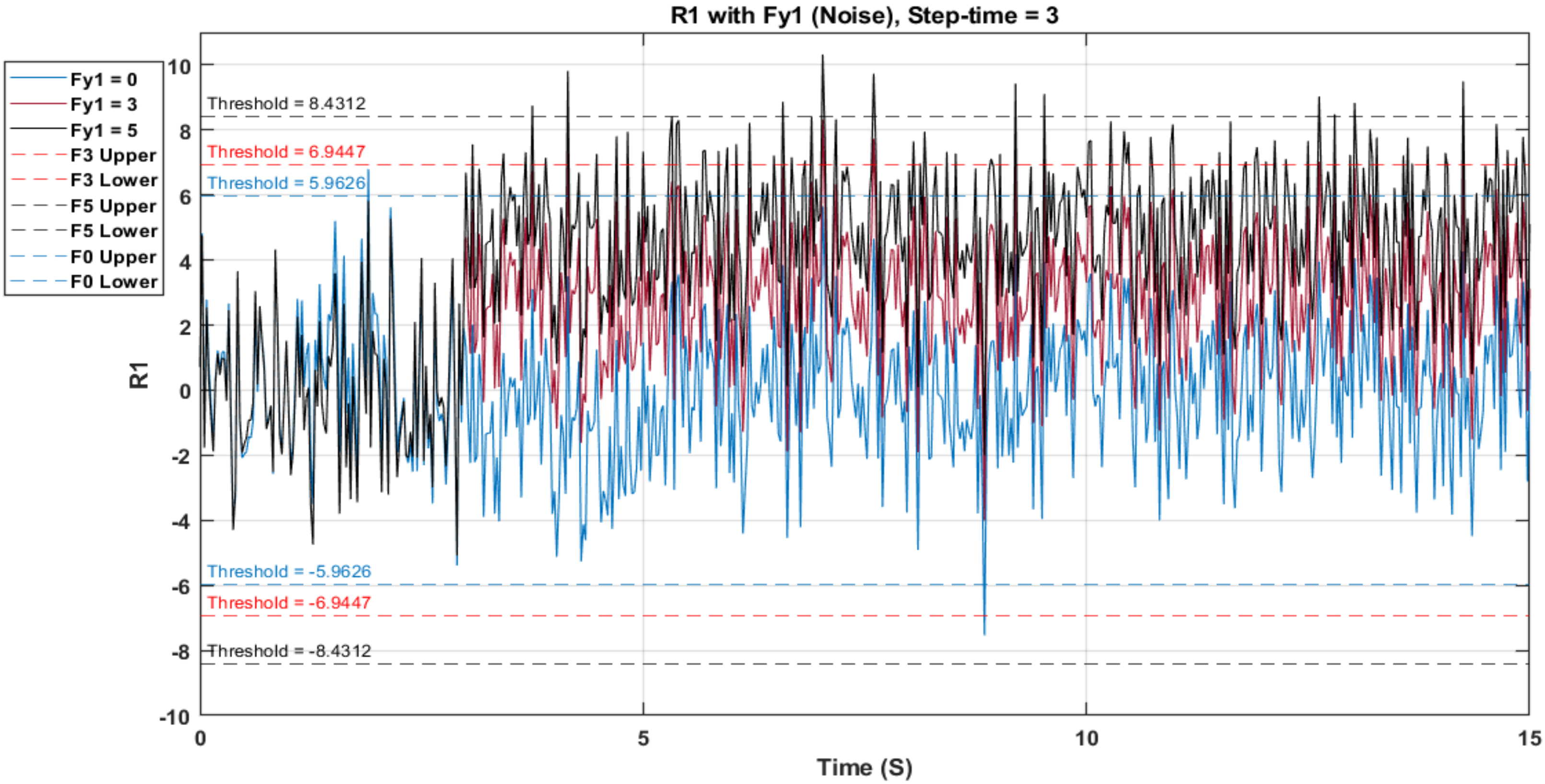
Now we apply different magnitudes of Abrupt Faults both at inputs and outputs and generate Residuals to check the overall consistency of mathematical equations of system with measurements.

SIMULINK REPRESENTATION OF MODEL 1,2 AT INPUT AND OUTPUT ABRUPT FAULTS (STEP-WISE)

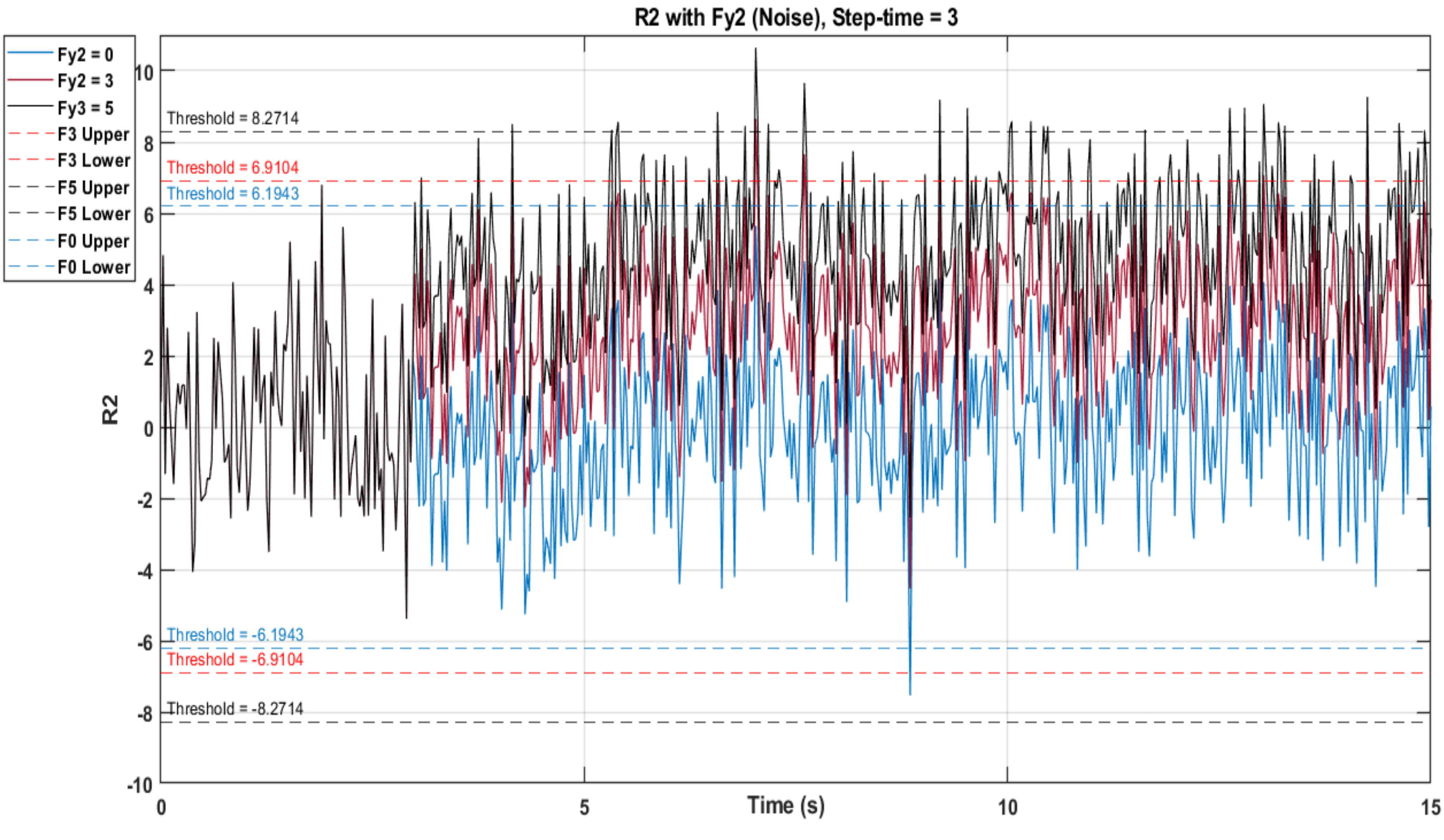


Different Fault scenarios and Graphs for Residual R1, R2

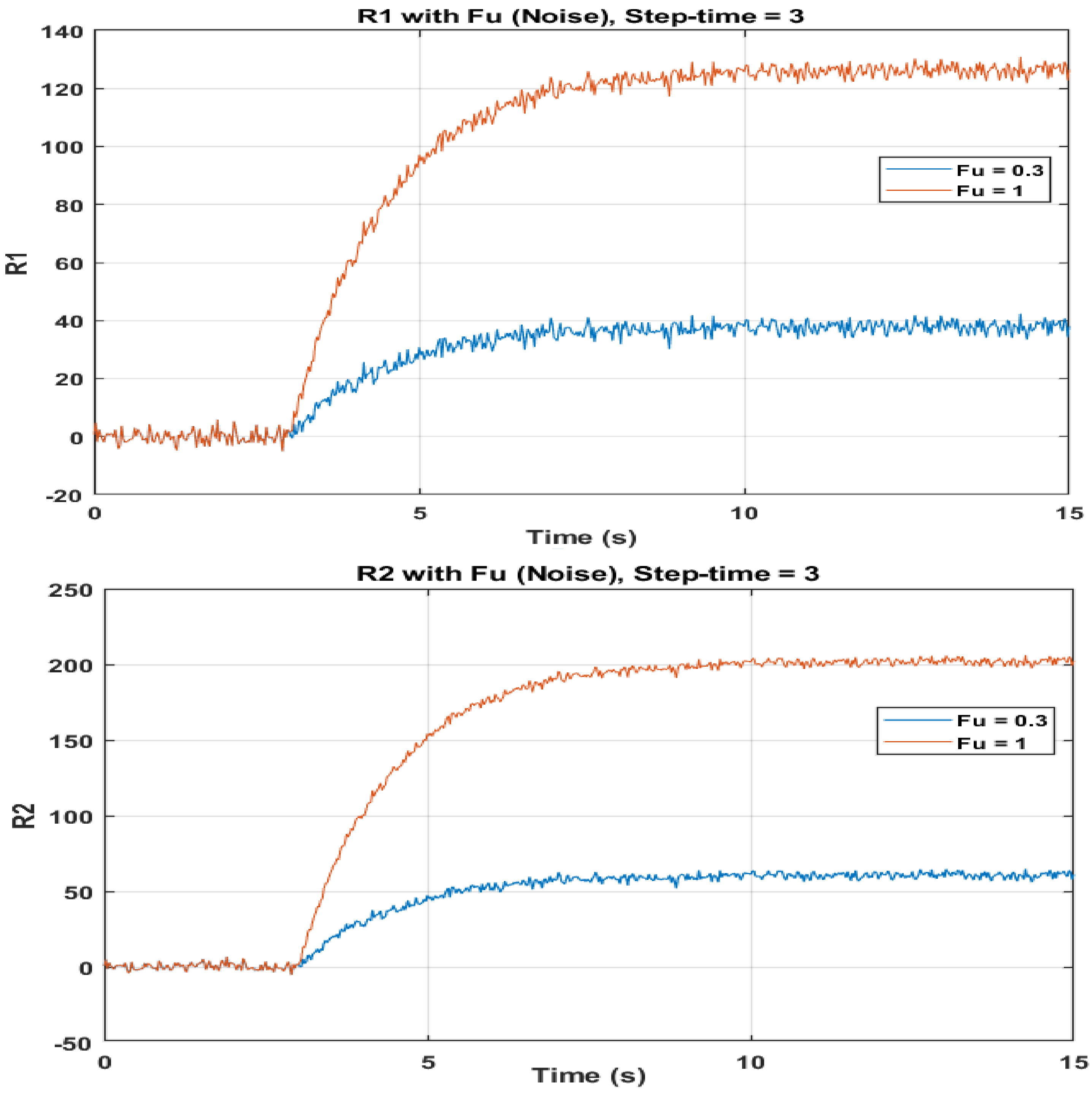




R1 with Fy1 (Noise), Step-time = 3
Fy1 = 0
Fy1 = 3
Fy1 = 5
F3 Upper
F3 Lower
F5 Upper
F5 Lower
F0 Upper
F0 Lower
Threshold = 8.4312
Threshold = 6.9447
Threshold = 5.9626
Threshold = -5.9626
Threshold = -6.9447
Threshold = -8.4312
R1
Time (S)



R2 with Fy2 (Noise), Step-time = 3
Fy2 = 0
Fy2 = 3
Fy3 = 5
F3 Upper
F3 Lower
F5 Upper
F5 Lower
F0 Upper
F0 Lower
Threshold = 8.2714
Threshold = 6.9104
Threshold = 6.1943
Threshold = -6.1943
Threshold = -6.9104
Threshold = -8.2714
R2
Time (s)


R1 with Fu (Noise), Step-time = 3
140
120
100
80
60
40
20
0
-20
R1
0
5
10
15
Time (s)
Fu = 0.3
Fu = 1
R2 with Fu (Noise), Step-time = 3
250
200
150
100
50
0
-50
R2
0
5
10
15
Time (s)
Fu = 0.3
Fu = 1

Selection of Thresholds

- Visualize the variations of residuals over time
- Compute statistical measures in Matlab Mean, Standard Deviation (std), maximum (max) and minimum (min) values of residuals. This helps us determine the behavior of the system.
- Based on statistical measure I measure the noise level as standard deviation of residuals and multiple by + 3*(std) for upper threshold limit and -3*std(R) for lower thresholds

Threshold_R1 = 3*std(R1) and Threshold _R2=3*std(R2)

Fault Magnitudes	+/- Thresholds R1	Fault magnitudes	+/- Thresholds R2
Fy1 = 0	3*std(R1) = 5.9626	Fy2 = 0	3*std(R2) = 6.1943
Fy1 = 3	3*std(R1) = 6.9447	Fy2 = 3	3*std(R2) = 6.9104
Fy1 = 5	3*std(R1) = 8.4312	Fy2 = 5	3*std(R2) = 8.2714

Residuals Sensitivity

Residuals are highly sensitive even at small magnitudes of Input Faults Fu=0.3,1

Residuals	Fault Fu	Fault Fy1	Fault Fy2	Comments
R1	1	1	0	R1 is highly sensitive to Fu
R2	1	0	1	R2 is highly sensitive to Fu

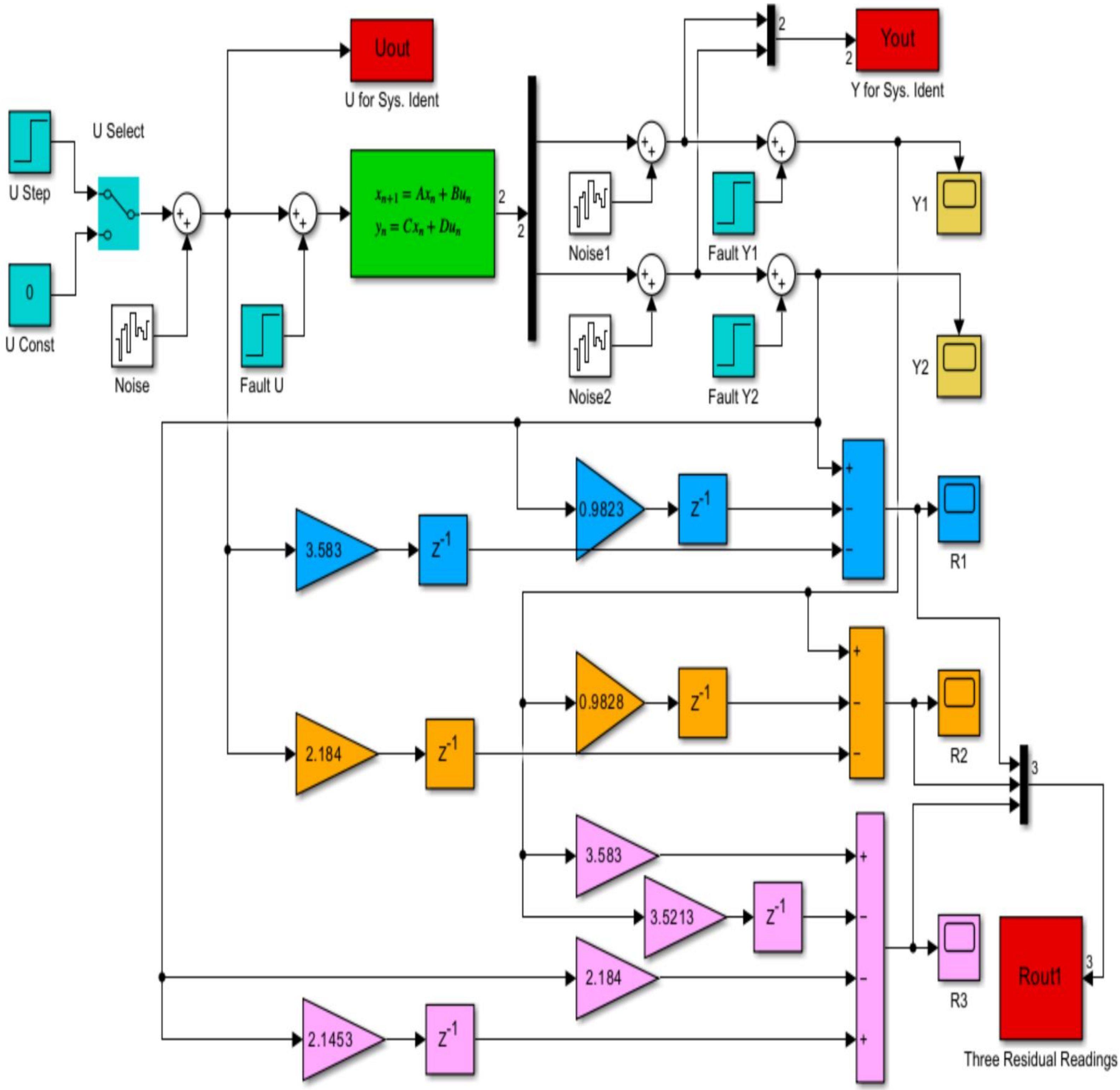
Residual Generation using Structured Residuals

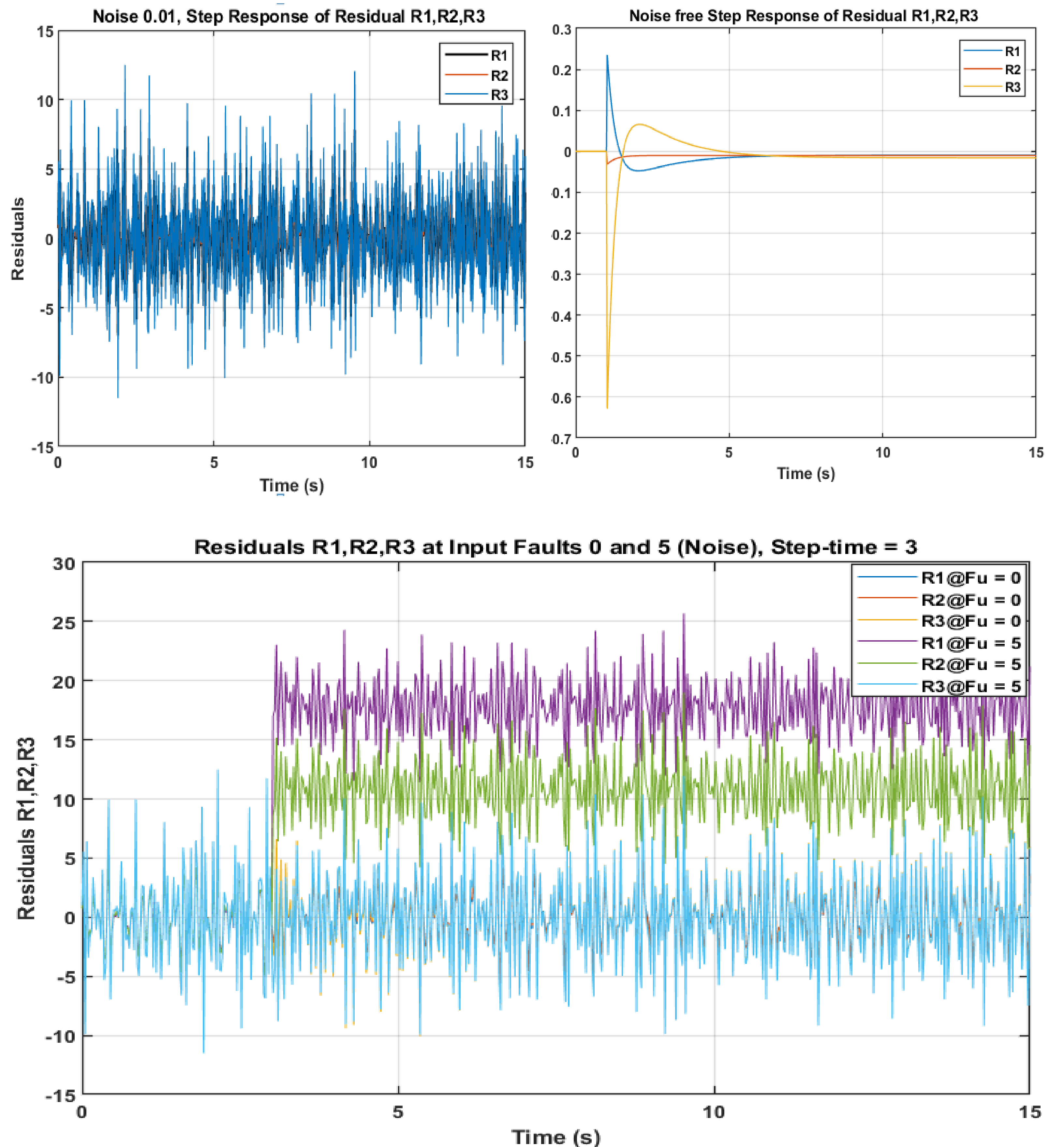
Structured residuals involve identifying the faults of interest and designing the specific residuals to be sensitive to those specific faults. This residuals are designed to have specific properties that make them sensitive to specific types of faults and insensitive to others. The aim is to generate residuals that are highly sensitive to faults of interest, while being robust to noise and disturbances.

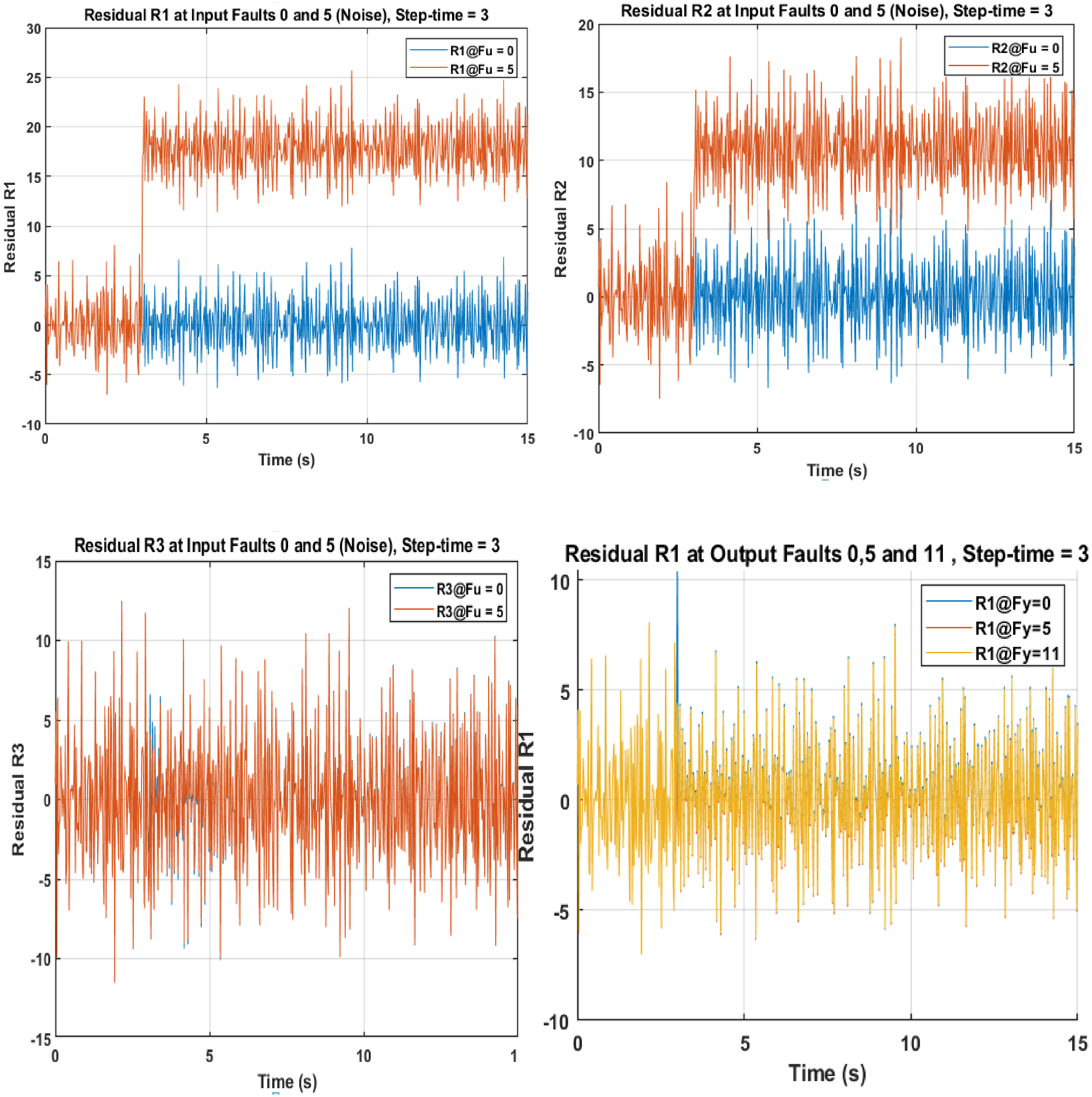
Structured residuals are designed to be orthogonal to each other, This property allows each residual to detect a specific fault without interference from other faults.

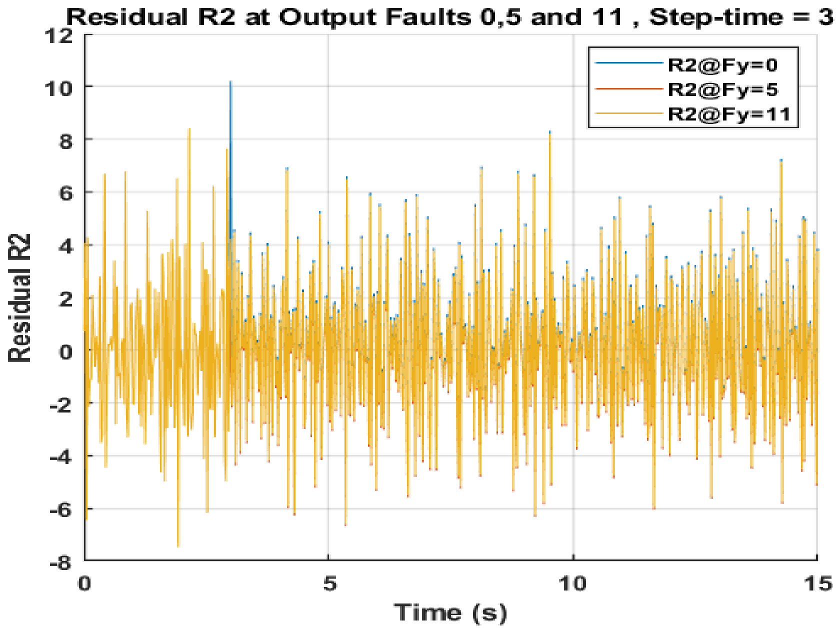
Symmetry: Structured residuals are designed to be symmetric, meaning they are equally sensitive to faults in either direction [1]. This property helps to detect faults that produce symmetrical effects in the system.

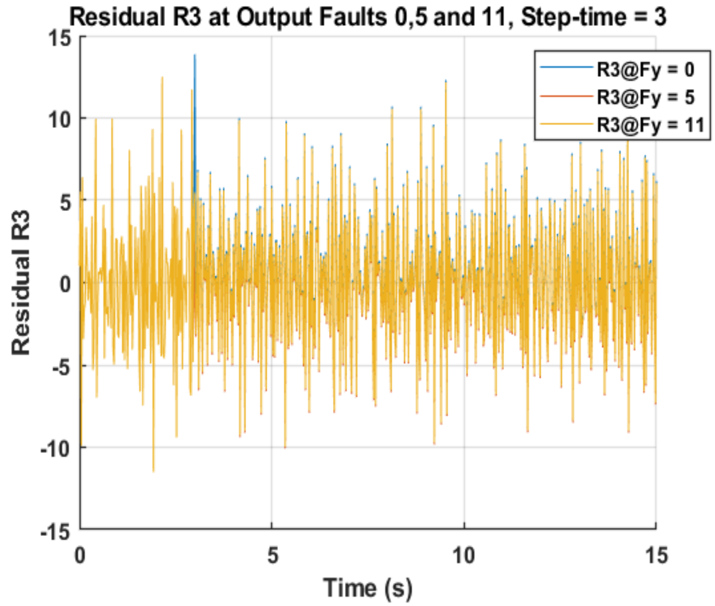
SIMULINK REPRESENTATION OF STRUCTURAL RESIDUALS

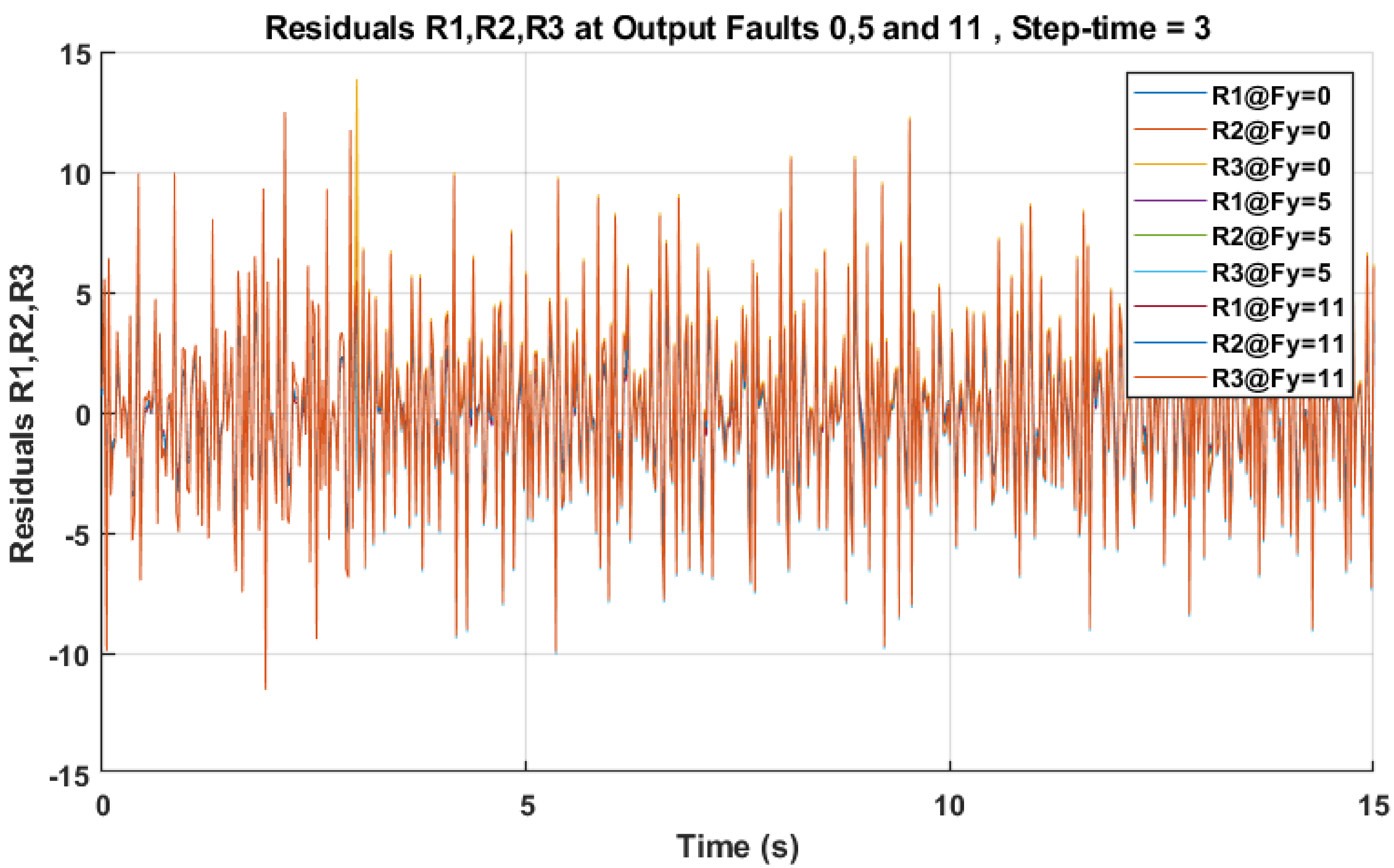



Noise 0.01, Step Response of Residual R1,R2,R3
R1
R2
R3
Residuals
Time (s)
Noise free Step Response of Residual R1,R2,R3
R1
R2
R3
Time (s)
Residuals R1,R2,R3 at Input Faults 0 and 5 (Noise), Step-time = 3
R1@Fu = 0
R2@Fu = 0
R3@Fu = 0
R1@Fu = 5
R2@Fu = 5
R3@Fu = 5
Residuals R1,R2,R3
Time (s)


Residual R1 at Input Faults 0 and 5 (Noise), Step-time = 3
R1@Fu = 0
R1@Fu = 5
Residual R1
Time (s)
30
25
20
15
10
5
0
-5
-10
0
5
10
15
Residual R2 at Input Faults 0 and 5 (Noise), Step-time = 3
R2@Fu = 0
R2@Fu = 5
Residual R2
Time (s)
20
15
10
5
0
-5
-10
0
5
10
15
Residual R3 at Input Faults 0 and 5 (Noise), Step-time = 3
R3@Fu = 0
R3@Fu = 5
Residual R3
Time (s)
15
10
5
0
-5
-10
-15
0
5
10
1
Residual R1 at Output Faults 0,5 and 11 , Step-time = 3
R1@Fy=0
R1@Fy=5
R1@Fy=11
Residual R1
Time (s)
10
5
0
-5
-10
0
5
10
15


Residual R2 at Output Faults 0,5 and 11 , Step-time = 3
R2@Fy=0
R2@Fy=5
R2@Fy=11
Residual R2
Time (s)


Residual R3 at Output Faults 0,5 and 11, Step-time = 3
R3@Fy = 0
R3@Fy = 5
R3@Fy = 11
Residual R3
Time (s)


Residuals R1,R2,R3 at Output Faults 0,5 and 11 , Step-time = 3
R1@Fy=0
R2@Fy=0
R3@Fy=0
R1@Fy=5
R2@Fy=5
R3@Fy=5
R1@Fy=11
R2@Fy=11
R3@Fy=11
Residuals R1,R2,R3
Time (s)

Threshold Selection

Thresholds were a bit wider than usual due to high noise in the system. And if we choose standard deviation we have lower limits of R1,R2,R3.
Thus, it is suggested to add low pass filter to the system to clear the noise which might make the fault detection easier for the thresholds. It is also suggested to use different fault detection method for task3, such as adaptive threshold or statistical methods like interquartile range IQR.
The simulation is completed without any faults being present in the system despite of having faults at the Output

Residuals	Fu	Fy1	Fy2	No Fault
R1	1	1	0	0
R2	1	0	1	0
R3	0	1	1	0

References

1. Jie Chen, Ron J. Patton Robust Model-Based Fault Diagnosis for Dynamic Systems
    a. https://link.springer.com/book/10.1007/978-1-4615-5149-2
2. Matlab System Identification Toolbox
    a. https://uk.mathworks.com/help/ident/index.html?s_tid=srchtitle_system%20identification_1
3. Rolf Isermann Fault-Diagnosis Applications Model-Based Condition Monitoring: Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems

Appendix

Matlab Scripts

```
clear all
close all
sim Task1_With_Noise.slx
% Model-1
z1      =   iddata(Yout(:,1),Uout,0.025);
% Validation input-output data, specified as an iddata
object. Data can
% have multiple input-output channels. Data is time-domain.
order   =   [1 1 1];
m1      =   oe(z1,order)   % Discrete-time Output error
(transfer function) model
% Model-2
z2      =   iddata(Yout(:,2),Uout,0.025);
order   =   [1 1 1];
m2      =   oe(z2,order)  % Discrete-time Output error
(transfer function) model
% Check for Missing data and Outliers
Missing_data_M1    =   misdata(z1,m1)
Missing_data_M2    =   misdata(z2,m2)
% Model checking and Validation
zm=merge(z1,z2)
zn=merge(Missing_data_M1,Missing_data_M2)
zm.exp = {'Model 1';'Model 2'}
zn.exp = {'Missing_data_M1';'Missing_data_M2'}
figure(1)
plot(zm,'b-')
grid on
legend('show')
hold on
plot(zn,'r--')
hold off
```

```
grid on
legend('show')

m2.OutputName = {'y2'}
z2.OutputName={'y2'}
figure(2)
compare(z1,m1)      % Validation of model 1 with the real
plant
grid on
legend('show')
figure(3)
compare(z2,m2)      % Validation of model 2 with the real
plant
grid on
legend('show')
figure(4)
resid(z1,m1)        % resid plots the autocorrelation of the
residuals and the cross-correlation of the residuals with the
input signals.
grid on
legend('show')
figure(5)
resid(z2,m2)
% The cross-correlation between residuals of m2 and the
inputs lie in the 99% confidence band for all lags.
grid on
legend('show')

Threshold_R1_fy1_0 = 3*std(R1)
Threshold_R2_fy1_0 = 3*std(R2)
yline(5.9626, '--c', 'Threshold = 5.9626');
yline(-5.9626, '--c', 'Threshold = -5.9626');
```

```
figure(1)
hold on
plot(tout,R1)
grid on
legend('show')
xlabel('Time (s)')
ylabel('Residual R1')
title('Residual R1 at Output Faults 0,5 and 11 , Step-time = 
3')
hold off
figure(2)
hold on
plot(tout,R2)
grid on
legend('show')
xlabel('Time (s)')
ylabel('Residual R2')
title('Residual R2 at Output Faults 0,5 and 11 , Step-time = 
3')
hold off
figure(3)
grid on
hold on
plot(tout,R3)
legend('show')
xlabel('Time (s)')
ylabel('Residual R3')
title('Residual R3 at Output Faults 0,5 and 11, Step-time = 
3')
hold off
figure(4)
hold on
grid on
plot(tout,Rout1)
legend('show')
```

```
xlabel('Time (s)')
ylabel('Residuals R1,R2,R3')
title('Residuals R1,R2,R3 at Output Faults 0,5 and 11 , Step-
time = 3')
hold off
```