2010

A Comparative Study on Fault Detection and Self-Reconfiguration

Ning Ge
Cleveland State University

A COMPARATIVE STUDY ON FAULT DETECTION AND SELF-RECONFIGURATION

NING GE

Bachelor of Mechanical Engineering

Dalian University of Technology

July, 2008

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

CLEVELAND STATE UNIVERSITY

DECEMBER, 2010

This thesis has been approved for the

Department of Mechanical Engineering and the

College of Graduate Studies by

Thesis/Dissertation Committee Chairperson, Dr. Paul P. Lin

Department/Date

Dr. Zhiqiang Gao

Department/Date

Dr. Lili Dong

Department/Date

Dr. Hanz Richter

Department/Date

ACKNOWLEDGEMENT

I would like to thank my advisor Dr. Paul P. Lin, who provided essential support to my graduate study, for his kindness during the last two years, and immense guidance towards the completion of this thesis.

I also would like to thank Dr. Zhiqiang Gao, Dr. Lili Dong and Dr. Hanz Ritcher for the encouragement while I was taking the courses, the advices for this thesis and serving as my committee members.

Finally, I want to express my gratitude to my wife, Chi Ma, for her constant encouragement and support. I also want to thank my parents and friends who have believed in me and inspired me.

A COMPARATIVE STUDY ON FAULT DETECTION AND SELF-RECONFIGURATION

NING GE

ABSTRACT

Extended State Observer (ESO) and the $\alpha-\beta-\gamma$ Tracker are introduced and compared. In comparison, the ESO is found to be more noise resistant. The extended state used for the estimation of the general system dynamics in real time makes it suitable for fault detection. Four control schemes are proposed for self-reconfiguration upon fault detection. These schemes are Active Disturbance Rejection Control, Tracker-based Feedback Control, Fuzzy Logic Control and Tracker-based PID Control. To compare their control performance, these schemes are applied to three different applications namely Active Engine Vibration Isolation System, Three-Tank Dynamic System and MEMS Gyroscope System. The advantages and disadvantages of using the control schemes for each application are presented.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

CHAPTER I

GENERAL INTRODUCTION

State observers are classified as model-based and model free. The objective of this thesis is to perform a comparative study in the context of fault detection and reconfiguration, when the noise is assumed to be unemployed and employed. The rate of convergence and estimation accuracy will be investigated.

Practically speaking, any dynamic system is subject to faults. In general, there are three kinds of faults: actuator faults, sensor faults and process faults. There have been many studies on the first two kinds of faults, but very few on the last kind of faults. This thesis focuses on the process faults that greatly affect the system dynamics.

The ideas of diagnostic observers probably originated from Beard [1] and Jones [2] with fault detection for linear systems and Clark et.al. [3] for detecting instrument malfunction in control systems. In this thesis, a relatively new observer called Extended State Observer (ESO) and an optimal design of $\alpha-\beta-\gamma$ filter, also known as $\alpha-\beta-\gamma$

Tracker, which is developed by Tenne and Singh [4], will be introduced for the purpose of fault detection.

Fault detection has two classes of designs. One is model-based, which relies on the mathematical system models, the other is model free or partially model-based, which does not need the system models or needs to know part of the system information. By means of the two observers mentioned in this thesis, many researchers have done the similar work. Lin and Singh [5] diagnosed the nonlinear dynamic system with multiple faults by means of a real-time tracker, which is model free. Ye, Lin and Gao [6] investigated the application of fault diagnosis using Extended State Observer (ESO), which only requires to tune one system parameter, the observer bandwidth. Radke [7] also conducted the fault detection by using ESO and applied on a three-tank system, and gave the comparisons of the effects of disturbance estimation between ESO and conventional filters and observers. Later, Ye [8] also used ESO to detect the road surface condition in his thesis.

It is desirable for a dynamic system to self-reconfigure upon detecting a fault. This can be accomplished by using effective control schemes. This thesis conducted a comparative study on control performance among four control schemes with three applications.

The thesis is organized as follows: Chapter I gives the general introduction. Chapter II presents the observers used for fault detection. Chapter III discusses the issue of tracking speeds and tracking accuracy, and makes comparisons between the given observers. Chapter IV presents four control schemes for self-reconfiguration, which are conpared in three applications shown in Chapter V. Finally, observations and conclusions are given in Chapter VI.

CHAPTER II

EXTENDED STATE OBSERVER (ESO) AND $\alpha-\beta-\gamma$ TRACKER

In this chapter, two fault detection techniques using Extended State Observer (ESO) and $\alpha-\beta-\gamma$ tracker are introduced. They are both state observers, but with different characteristics. The ESO observes states and also filters noise, but requires some basic knowledge of the system dynamics, such as the order. The $\alpha-\beta-\gamma$ tracker, on the other hand, relies only on the system output, and does not require any knowledge of the system model. It is essentially a model free state estimator. However, it does not filter noise by itself.

2.1 Extended State Observer (ESO)

State observer is used to estimate the internal states of the system using its input and output. In control history, state observer has always played an important role, which could be generally divided into two kinds. The first kind is based on a known model and the other kind is based on unknown or partially known model. With an accurate system model, the state observer could obtain accurate estimates. However, most dynamic systems cannot be accurately modeled. In reality, disturbances, noise and many other factors play a role in system model uncertainty. Extended State Observer (ESO), which belongs to the second kind of observers, is introduced below.

2.1.1 ESO Formulation

The concept of Extended State Observer (ESO) was originally proposed by Han [9]. By tuning several parameters in the system, ESO could offer relatively accurate estimates of system outputs and dynamics. Later, Gao [10] enhanced the tuning by simplifying the number of tuning parameters to one.

Consider a general second-order system

$$\ddot{y} = a_1\dot{y} + a_2 y + D + bu \tag{2.1}$$

where y, D, u are respectively output, external disturbance and input of the system, and a_1, a_2 and b are usually unknown.

$$\ddot{y} = a_1\dot{y} + a_2 y + D + (b - b_0)u + b_0 u = f + b_0 u \tag{2.2}$$

where $f = a_1\dot{y} + a_2 y + D + (b - b_0)u$ is called generalized disturbance, or simply disturbance. The f contains unknown internal states $a_1\dot{y} + a_2 y + (b - b_0)u$ and external disturbance *D*.

ESO provides a mean of estimating f and y in real time. To build the observer, the plant can be expressed as

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + b_0 u \\ \dot{x}_3 = \dot{f} \\ y = x_1 \end{cases} \tag{2.3}$$

where x_3 is referred to as Extended State.

Rewriting the equations in state space form, gives

$$\begin{cases} \dot{x} = Ax + Bu + E\dot{f} \\ y = Cx \end{cases} \tag{2.4}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ b_0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, E = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The state space observer, denoted as extended state observer (ESO), is constructed as

$$\begin{cases} \dot{z} = Az + Bu + L(y - \hat{y}) \\ \hat{y} = Cz \end{cases} \tag{2.5}$$

where $\hat{y}$ is the estimation of the system output y, and L is the observer gain vector which can be obtained by employing a pole placement technique. L can be expressed as

$$L = [\beta_1 \quad \beta_2 \quad \beta_3] \tag{2.6}$$

The three parameters β_1, β_2, β_3 need to be tuned in the observer. As the order of the plant and observer increases, the number of parameters that need to be tuned also increases. Gao [10] developed ω_o -parameterization technique to simplify the observer tuning. This parameterization assigns all observer eigenvalues at $-\omega_o$, and makes all parameters of an observer a function of ω_o. Here ω_o is the bandwidth of the observer.

$$\lambda(s) = s^3 + \beta_1 s^2 + \beta_2 s + \beta_3 = \left(s + \omega_o\right)^3 \tag{2.7}$$

where $\beta_1 = 3\omega_o$, $\beta_2 = 3\omega_o{}^2$, and $\beta_3 = \omega_o{}^3$. The parameterization method can be extended to a nth-order plant, and the only parameter to tune is the bandwidth ω_o.

2.1.2 Fault Detection by Means of the ESO

Most fault detection techniques rely only on the change of system outputs. More specifically, a fault is considered detected when the abrupt change of the systems' outputs exceed the predetermined threshold values. With ESO, an augmented/extended state is used to estimate the general system dynamics, which provides a foundation for detecting process faults in real time because the faults greatly affect the system dynamics. On using the ESO, the fault is considered detected when the abrupt change of the system dynamics exceeds the pre-determined threshold value.

2.2 Introduction to $\alpha-\beta-\gamma$ Tracker

The $\alpha-\beta-\gamma$ tracker can be used without the knowledge of the system model. It is a one-step-ahead position and velocity predictor. It has been assumed that the jerk (i.e. time derivative) is negligible or the time interval (Δt) is very small, so it could be used for third-order systems or orders lower than that. When the jerk or time interval is not small, a higher-order tracker, called the $\alpha-\beta-\gamma-\delta$ tracker denoted by Wu, et.al. [11] could be applied.

2.2.1 Tracker Formulations

The tracker formulations are given below:

$$x_p(k+1) = x_s(k) + \Delta t v_s(k) + \frac{1}{2}\Delta t^2 a_s(k) \tag{2.8}$$

$$v_p(k+1) = v_s(k) + \Delta t a_s(k) \tag{2.9}$$

where $x_s(k)$, $v_s(k)$ and $a_s(k)$ are the smoothed position, velocity and acceleration at the step k, respectively.

The smoothing equations are as follows:

$$\begin{cases} x_s(k) = x_p(k) + \alpha\left[x_o(k) - x_p(k)\right] \\ v_s(k) = v_p(k) + \dfrac{\beta}{\Delta t}\left[x_o(k) - x_p(k)\right] \\ a_s(k) = a_s(k-1) + \dfrac{\gamma}{2\Delta t^2}\left[x_o(k) - x_p(k)\right] \end{cases} \tag{2.10}$$

where x_o, x_s and x_p are the observed (measured), smoothed and predicted positions respectively; v_o, v_s and v_p are the observed (measured), smoothed and predicted velocities respectively; a_s is the smoothed acceleration, Δt is the time interval and α, β and γ are smoothing parameters.

Applying the z-transform to (2.8) to (2.10) and solving for the ratio x_p / x_o leads to the prediction transfer function in z-domain, which is:

$$G(z)=\frac{(\alpha+\beta+\frac{1}{4}\gamma)z^2+(-2\alpha-\beta+\frac{1}{4}\gamma)z+\alpha}{z^3+(\alpha+\beta+\frac{1}{4}\gamma-3)z^2+(-2\alpha-\beta+\frac{1}{4}\gamma+3)z+\alpha-1}=\frac{x_p}{x_o} \tag{2.11}$$

Jury's Stability Test

The roots of the characteristic polynomial, the denominator of the transfer function, are required to lie within the unit circle of stability. Because of the differences in the Z and S domains, the Routh-Hurwitz (RH) criteria cannot be used directly with digital systems. This is because digital systems and continuous-time systems have different regions of stability. However, two methods: Bilinear transform and Jury's Stability Test can be used to analyze the stability of digital systems. Jury's test is a procedure similar to the RH test, except it has been modified to analyzed to analyze digital systems in the Z domain directly. In short, Jury's stability test is a stability criterion for discrete-time system. Jury's stability test [12] results the constraints on the three parameters as:

$$0<\alpha<2 \tag{2.12a}$$

$$0<\beta<4-2\alpha \tag{2.12b}$$

$$0<\gamma<\frac{4\alpha\beta}{2-\alpha} \tag{2.12c}$$

Unlike the popular Kalman or the Extended Kalman filter, the tracker does not require large computations or matrix inversion.

To start the tracking process, three initial positions are needed.

$$x_p(1)=x_o(1);x_p(2)=x_o(2);x_p(3)=x_o(3) \tag{2.13a}$$

$$v_s(3) = \frac{x_o(3) - x_o(2)}{\Delta t}; a_s(3) = \frac{x_o(3) - 2x_o(2) + x_o(1)}{\Delta t^2} \tag{2.13b}$$

$$\begin{cases} x_p(4) = x_p(3) + v_s(3)\Delta t + \frac{1}{2} a_s(3)\Delta t^2 \\ v_p(4) = v_p(3) + a_s(3)\Delta t \end{cases} \tag{2.13c}$$

Accelerations are used to help estimate the positions and velocities. In essence, only the positions and velocities can be tracked by means of this tracker.

2.2.2 Example of $\alpha - \beta - \gamma$ Tracking

Assume that the target position is described by equation (2.14) and with no noise.

$$x(t) = \log(t^3 - 5t + 1) \tag{2.14}$$

where x is in mm and t is in sec. Figure 1 shows the path for the first 100 seconds.

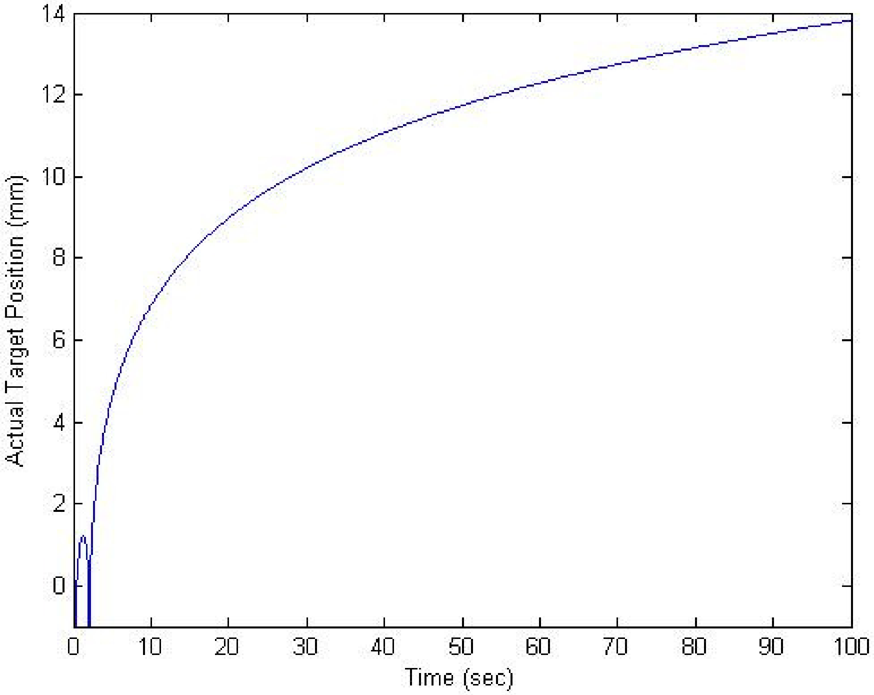


Figure 1 Actual target position

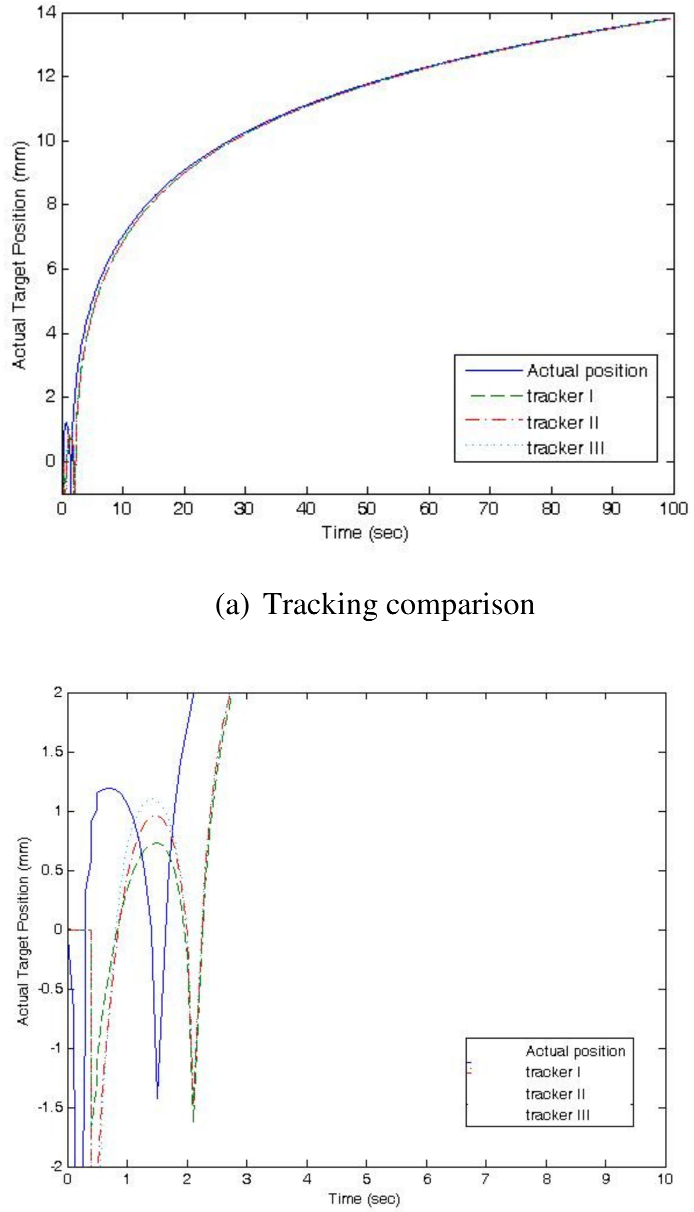
Figures 2 and 3 show the comparison between actual target position and the position predicted by the tracker with three different values of coefficients:

$\alpha = 0.6$, $\beta = 0.3$ and $\gamma = 0.08$ (tracker I)

$\alpha = 1$, $\beta = 0.5$ and $\gamma = 0.3$ (tracker II)

$\alpha = 1$, $\beta = 1$ and $\gamma = 1$ (tracker III)

The selected α, β and γ values all satisfy the Jury's stability test as stated in equation (2.12).

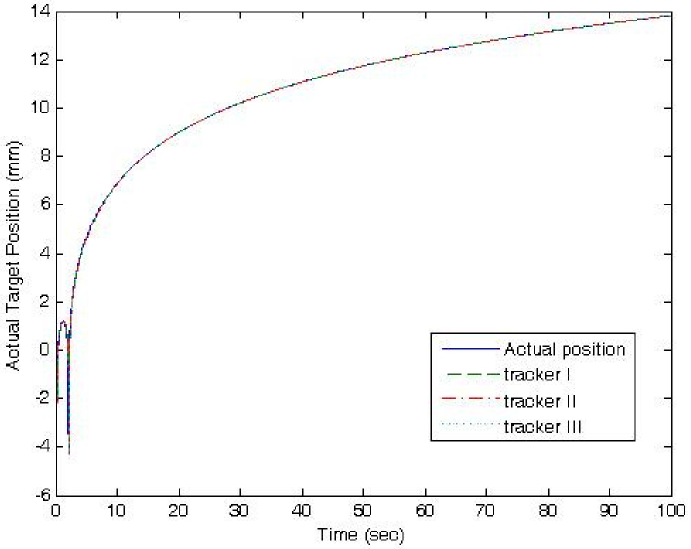


(a) Tracking comparison

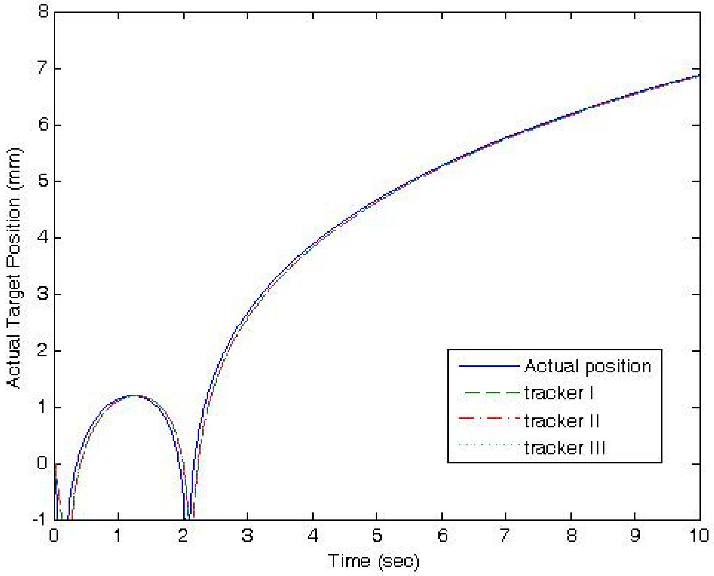
(b) Close look of initial tracking

Figure 2 Comparison between actual and predicted target position when $\Delta t = 0.1s$

Time interval affects the tracking accuracy. When the chosen time interval is 0.1s, the maximum tracking error for Tracker III is around 10%. The maximum tracking error occurs at the beginning, but quickly decreases. The error approaches zero after about 20 sec; when the time interval is changed to 0.01s, the maximum tracking error for Tracker III is around 1%. The error approaches zero after about 10 sec.



(a) Tracking comparison



(b) Close look of initial tracking

Figure 3 Comparison between actual and predicted target position when $\Delta t = 0.01s$

In conclusion, regardless of the selected α, β and γ values, the tracker will soon accurately track the target, as long as the three coefficient values satisfy the Jury's stability test.

Since the $\alpha - \beta - \gamma$ tracker does not have the extended state to track the system dynamics, the fault detection, in this case, is limited to monitoring the change of the system outputs. However, in some special cases, the change of the system outputs is not obvious enough for fault detection. By monitoring the change of the derivatives of the system outputs (i.e. velocity), this tracker could still be used for fault detection.

CHAPTER III

COMPARISON BETWEEN ESO AND $\alpha-\beta-\gamma$ TRACKER

In this chapter, the Extended State Observer (ESO) and the $\alpha-\beta-\gamma$ tracker are compared in terms of tracking accuracy and tracking speed. A three-tank dynamic system is chosen for a case study.

3.1 Three-Tank Dynamic System Design

A nonlinear three-tank dynamic system [13-15] is shown in Figure 4, which consists of two pumps and three cylindrical tanks that are connected by cylindrical small pipes. In this multiple-input-multiple-output (MIMO) system, the inputs are the flow rate of each pump, and the outputs are the water levels of the three tanks. The pipe blockage is in terms of degree of fault between 0 and 1, where 0 and 1 correspond to complete blockage and no blockage respectively.

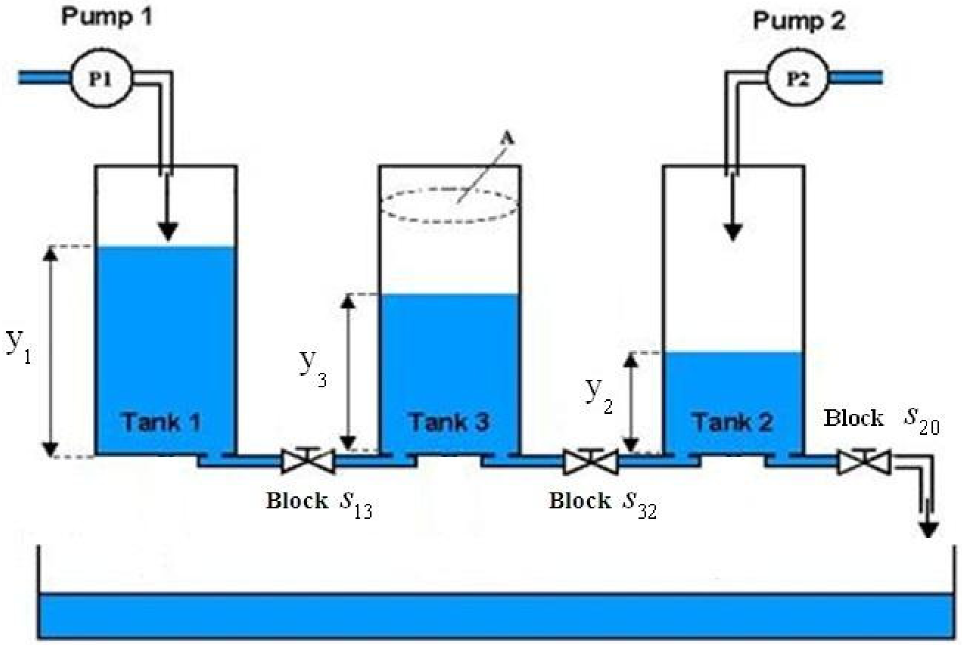


Figure 4 Three-Tank System Schematics

The following few assumptions are made for this system:

1) The two inputs (the pump rates) are controllable, and three outputs (the water levels of three tanks) are measurable
2) Single or multiple faults will not occur until the system reaches its steady state
3) Multiple faults do not occur simultaneously.

Furthermore, the exact system model is assumed unknown. However, since measured data are available, it is necessary to use the exact system model that can be derived from using the Torricelli's law in order to compare the tracking speed and tracking accuracy. The three dynamic equations are:

$$\begin{cases} A\dfrac{dy_1}{dt} = -a_1 s_{13} sign(y_1 - y_3)\sqrt{2g\left|y_1 - y_3\right|} + Q_1 \\ A\dfrac{dy_2}{dt} = -a_3 s_{32} sign(y_3 - y_2)\sqrt{2g\left|y_3 - y_2\right|} - a_2 s_{20}\sqrt{2gy_2} + Q_2 \\ A\dfrac{dy_3}{dt} = a_1 s_{13} sign(y_1 - y_3)\sqrt{2g\left|y_1 - y_3\right|} - a_3 s_{32} sign(y_3 - y_2)\sqrt{2g\left|y_3 - y_2\right|} \end{cases} \tag{3.1}$$

where

A : the circular cross-section area of each tank (same for all);

g : gravitational acceleration (9.81 m/sec2);

sign: sign function, “1” if >0 and “-1” if <0;

a_1, a_2, a_3 : the circular cross-section area of each pipe;

Q_1, Q_2 : the pump flow reates;

y_1, y_2, y_3 : the water level of each tank

s_{13}, s_{32}, s_{20} : the pipe blockage

In using the ESO, the equations for this first-order system can be stated as:

$$\dot{y} = f + b_0 u \tag{3.2}$$

where b_0 is a constant and u is the system input, and the f is known as general system dynamics which is shown below:

$$\begin{cases} f_1 = \left[-a_1 s_{13} sign\left(y_1 - y_3 \right) \sqrt{2g\left| y_1 - y_3 \right|} \right] / A \\ f_2 = \left[a_3 s_{32} sign\left(y_3 - y_2 \right) \sqrt{2g\left| y_3 - y_2 \right|} - a_2 s_{20} \sqrt{2g y_2} \right] / A \\ f_3 = \left[a_1 s_{13} sign\left(y_1 - y_3 \right) \sqrt{2g\left| y_1 - y_3 \right|} - a_3 s_{32} sign\left(y_3 - y_2 \right) \sqrt{2g\left| y_3 - y_2 \right|} \right] / A \end{cases} \tag{3.3}$$

where f_1, f_2 and f_3 are the general system dynamics of tank 1, tank 2 and tank 3, respectively.

3.2 Comparisons without noise

The tracking speed and tracking accuracy are compared between the $\alpha-\beta-\gamma$ tracker and the ESO. The comparisons are made when the system is fault free and when a single fault ($s_{13} = 0.6$) occurs after reaching its steady state. For the ESO, ω_o was chosen as 2. For the $\alpha-\beta-\gamma$ tracker, the three values α, β and γ are 1, 1.5 and 2, respectively. Figure 5 shows the difference of water levels between the ESO and the exact values.

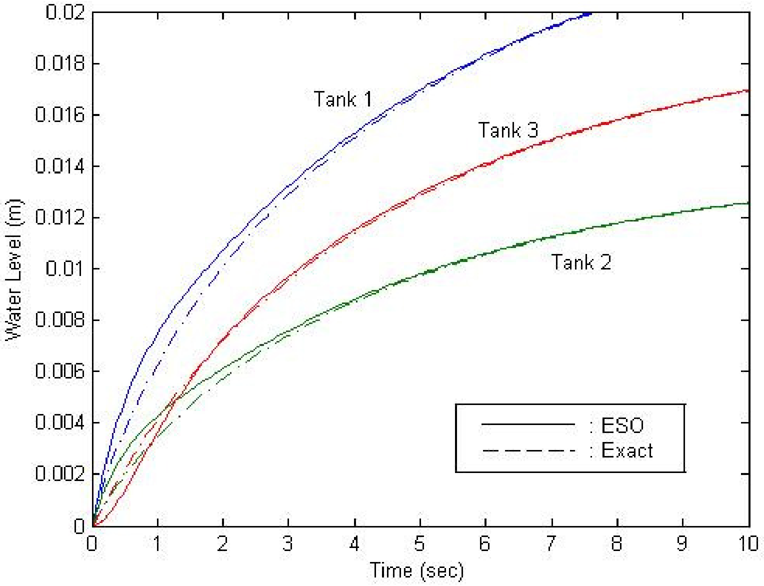


Figure 5 Output Comparison between ESO and Exact Values (Fault Free)

Figure 6 shows the difference of water levels between the $\alpha-\beta-\gamma$ tracker and the exact values.

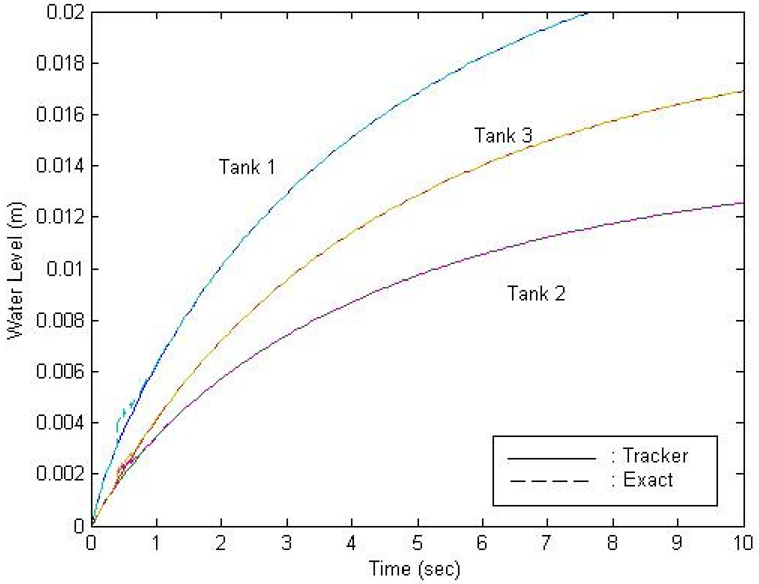


Figure 6 Output Comparison between Tracker and Exact Values (Fault Free)

Figures 5 and 6 show that the $\alpha-\beta-\gamma$ tracker tracks the system much quicker than the ESO. Table I lists detailed output comparison between the two methods with 0.01 sec sampling time. Fault free was assumed during the given time period.

Table I Comparisons of data from measured, ESO and the tracker

Time (sec)	Method	Water Level (mm)		
		Tank1	Tank2	Tank3
t = 0.1	Exact	0.97	0.45	0.43
	ESO	1.23	0.78	0.07
	Tracker	0.97	0.45	0.43
t = 0.2	Exact	1.77	0.87	0.89
	ESO	2.33	1.44	0.28
	Tracker	1.77	0.87	0.89
t = 0.3	Exact	2.48	1.26	1.33
	ESO	3.29	1.99	0.60
	Tracker	2.48	1.26	1.33
t = 0.4	Exact	3.13	1.62	1.77
	ESO	4.13	2.45	0.97
	Tracker	3.13	2.00	2.12
t = 0.5	Exact	3.73	1.97	2.20
	ESO	4.87	2.86	1.40
	Tracker	4.56	2.19	2.41
t = 0.6	Exact	4.29	2.29	2.61
	ESO	5.52	3.21	1.85

	Tracker	4.7	2.43	2.74
t = 0.7	Exact	4.82	2.60	3.02
	ESO	6.09	3.52	2.31
	Tracker	5.00	2.69	3.09
t = 0.8	Exact	5.33	2.90	3.40
	ESO	6.60	3.80	2.77
	Tracker	5.44	2.95	3.45
t = 0.9	Exact	5.82	3.19	3.78
	ESO	7.07	4.05	3.22
	Tracker	5.88	3.22	3.81
t = 1.0	Exact	6.28	3.46	4.14
	ESO	7.50	4.28	3.67
	Tracker	6.32	3.48	4.16
t = 1.1	Exact	6.73	3.73	4.49
	ESO	7.89	4.50	4.10
	Tracker	6.75	3.74	4.50
t = 1.2	Exact	7.15	3.98	4.83
	ESO	8.26	4.71	4.52
	Tracker	7.17	3.99	4.84
t = 1.3	Exact	7.57	4.22	5.16
	ESO	8.61	4.90	4.92
	Tracker	7.57	4.23	5.17
t = 1.4	Exact	7.96	4.46	5.48
	ESO	8.95	5.09	5.30

	Tracker	7.97	4.46	5.49
t = 1.5	Exact	8.35	4.69	5.80
	ESO	9.27	5.27	5.67
	Tracker	8.35	4.69	5.80
t = 1.6	Exact	8.72	4.91	6.10
	ESO	9.58	5.45	6.02
	Tracker	8.72	4.91	6.10
t = 1.7	Exact	9.08	5.13	6.39
	ESO	9.88	5.62	6.35
	Tracker	9.08	5.13	6.39
t = 1.8	Exact	9.43	5.33	6.67
	ESO	10.17	5.79	6.67
	Tracker	9.43	5.33	6.67
t = 1.9	Exact	9.77	5.53	6.95
	ESO	10.45	5.96	6.98
	Tracker	9.77	5.53	6.95
t = 2.0	Exact	10.10	5.73	7.22
	ESO	10.73	6.12	7.28
	Tracker	10.10	5.73	7.22

The average error and root mean square (RMS) error from Table I are listed in Table II:

Table II Comparison on Average Error and RMS Error (Fault Free)

Condition: Fault Free Time Duration: 2 sec. Unit: mm	Via ESO			Via $\alpha-\beta-\gamma$ Tracker		
	Tank1	Tank2	Tank3	Tank1	Tank2	Tank3
Average Error	0.9470	0.6735	0.3945	0.0885	0.0480	0.0450
RMS Error	0.9849	0.6987	0.4845	0.2224	0.1060	0.0981

For better filtering the noise, the ESO chose a relatively small bandwidth, which caused the fact that in this simulation, the tracker significantly outperformed the ESO in terms of average error and RMS error. With the ESO, the RMS errors are fairly close to the average errors, which mean that the estimations were scattered more evenly around the target values within a range. Table I and Table II indicate that the $\alpha-\beta-\gamma$ tracker outperforms the ESO in terms of average estimation error and RMS estimation error.

Among the three tanks, the estimation error for tank 3 is the smallest when using both methods. This is perhaps due to the fact that pumps (inputs) are directly connected only to tanks 1 and 2.

The rate of convergence and estimation accuracy are further compared when a fault occurring at $t = 40s$. Figures 7 and 8 show comparisons.

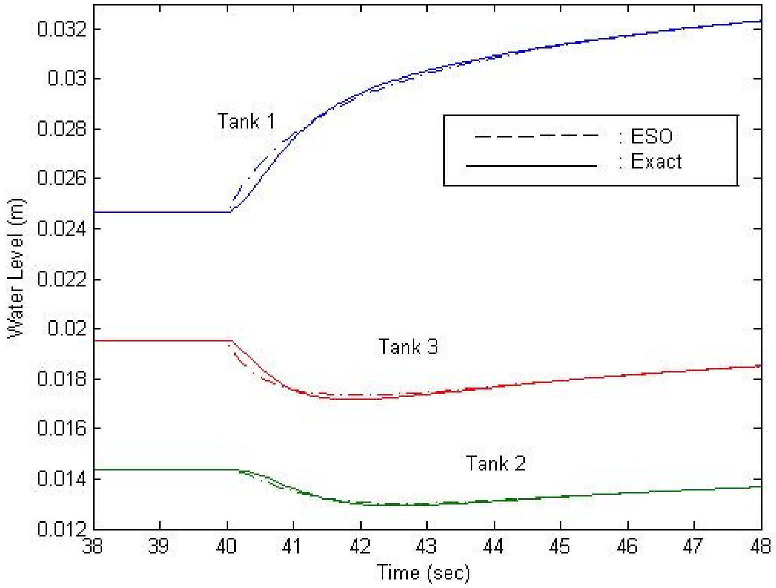


Figure 7 Output Comparison between ESO and Exact Values (Fault at t=40s)

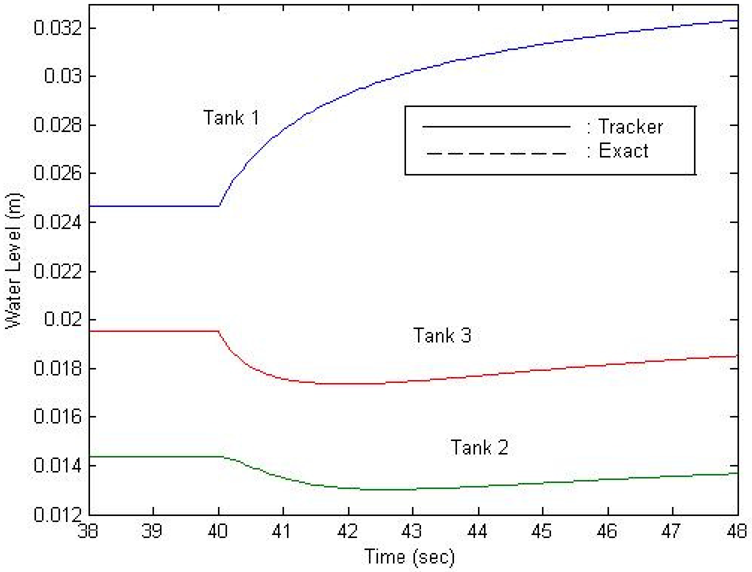


Figure 8 Output Comparison between Tracker and Exact Values (Fault at t=40s)

Table III lists the detailed comparison.

Table III Comparisons of data from measured, ESO and the tracker

Time (sec)	Method	Water Level (10^{-3}m)		
		Tank1	Tank2	Tank3
t = 40.1	Exact	25.16	14.32	19.07
	ESO	24.77	14.35	19.43
	Tracker	25.16	14.32	19.07
t = 40.2	Exact	25.58	14.26	18.72
	ESO	24.99	14.33	19.24
	Tracker	25.58	14.26	18.72
t = 40.3	Exact	25.95	14.16	18.45
	ESO	25.29	14.29	18.99
	Tracker	25.95	14.16	18.45
t = 40.4	Exact	26.28	14.06	18.21
	ESO	25.63	14.21	18.73
	Tracker	26.30	14.06	18.24
t = 40.5	Exact	26.59	13.95	18.07
	ESO	25.97	14.14	18.47
	Tracker	26.59	13.95	18.07
t = 40.6	Exact	26.87	13.85	17.93
	ESO	26.33	14.04	18.23
	Tracker	26.87	13.85	17.93
t = 40.7	Exact	27.13	13.75	17.81
	ESO	26.67	13.93	18.03

	Tracker	27.13	13.75	17.81
t = 40.8	Exact	27.37	13.65	17.72
	ESO	26.99	13.82	17.84
	Tracker	27.37	13.65	17.72
t = 40.9	Exact	27.59	13.57	17.64
	ESO	27.30	13.72	17.69
	Tracker	27.59	13.57	17.64
t = 41.0	Exact	27.80	13.49	17.58
	ESO	27.59	13.61	17.56
	Tracker	27.80	13.49	17.58
t = 41.1	Exact	27.99	13.41	17.52
	ESO	27.85	13.51	17.46
	Tracker	27.99	13.41	17.52
t = 41.2	Exact	28.17	13.35	17.48
	ESO	28.09	13.42	17.38
	Tracker	28.17	13.35	17.48
t = 41.3	Exact	28.34	13.30	17.44
	ESO	28.31	13.34	17.31
	Tracker	28.34	13.30	17.44
t = 41.4	Exact	28.50	13.25	17.40
	ESO	28.52	13.26	17.26
	Tracker	28.50	13.25	17.40
t = 41.5	Exact	28.65	13.20	17.39
	ESO	28.71	13.20	17.22

	Tracker	28.65	13.20	17.39
t = 41.6	Exact	28.79	13.17	17.37
	ESO	28.88	13.14	17.20
	Tracker	28.79	13.17	17.37
t = 41.7	Exact	28.92	13.14	17.36
	ESO	29.03	13.09	17.18
	Tracker	28.92	13.14	17.36
t = 41.8	Exact	29.05	13.11	17.36
	ESO	29.18	13.05	17.18
	Tracker	29.05	13.11	17.36
t = 41.9	Exact	29.17	13.09	17.35
	ESO	29.31	13.02	17.17
	Tracker	29.17	13.09	17.35
t = 42.0	Exact	29.29	13.07	17.35
	ESO	29.44	12.99	17.18
	Tracker	29.29	13.07	17.35

Table IV Comparison on Average Error and RMS Error (Single Fault)

Condition: Fault at 40s Time Duration: 2 sec. Unit: mm	Via ESO			Via $\alpha-\beta-\gamma$ Tracker		
	Tank1	Tank2	Tank3	Tank1	Tank2	Tank3
Average Error	0.2879	0.0945	0.2265	0.0010	0	0.0015
Root Mean Square of Error	0.3629	0.1119	0.2749	0.0045	0	0.0067

Once again, using the ESO with the chosen bandwidth, the $\alpha-\beta-\gamma$ tracker outperforms the ESO in the case of single fault occurrence. Based on the aforementioned data comparisons with fault free and single fault occurrence, and with the chosen bandwidth of ESO and three parameters of the tracker, it can be concluded that the $\alpha-\beta-\gamma$ tracker is better than the ESO in terms of estimation accuracy. The ESO's estimation errors primarily result from its slower rate of convergence.

In this simulation, the tracker generally takes less than half a second or equivalent to less than 50 time steps, to completely track the system, while the ESO takes a few seconds to achieve the same. Once the tracker tracks the system, the tracking becomes stabilized even with a sudden output change such as due to a fault. In contrast, the ESO slowly responds to the system's sudden change.

After comparing the rate of convergence and estimation accuracy between the ESO and the $\alpha-\beta-\gamma$ tracker, it is desired to study their control performance. Four control schemes are to be compared in Chapter IV.

3.3 Comparisons with noise

In the real world, noise and disturbance are unavoidable. In this section, 5% white noise will be added to the system output, then Figures 9, 10, 11 and 12 show the rate of convergence and estimation accuracy of the ESO and the Tracker.

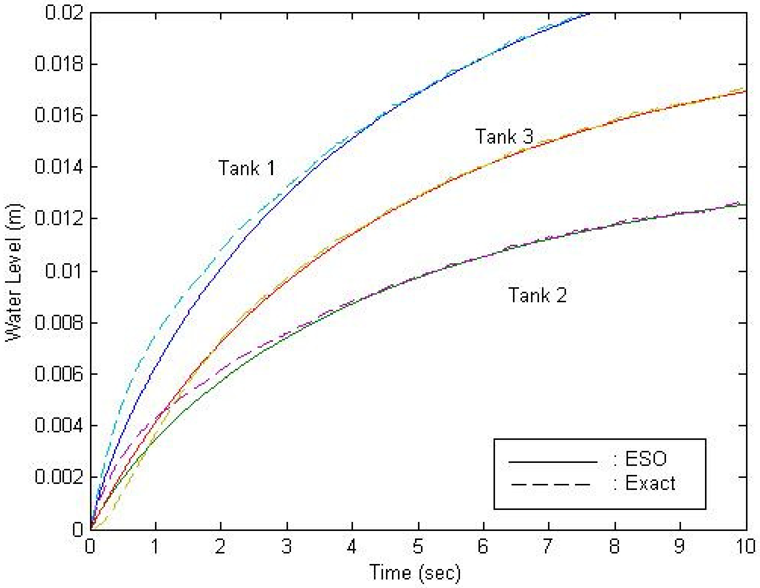


Figure 9 Output Comparison between ESO and Exact values (Fault Free)

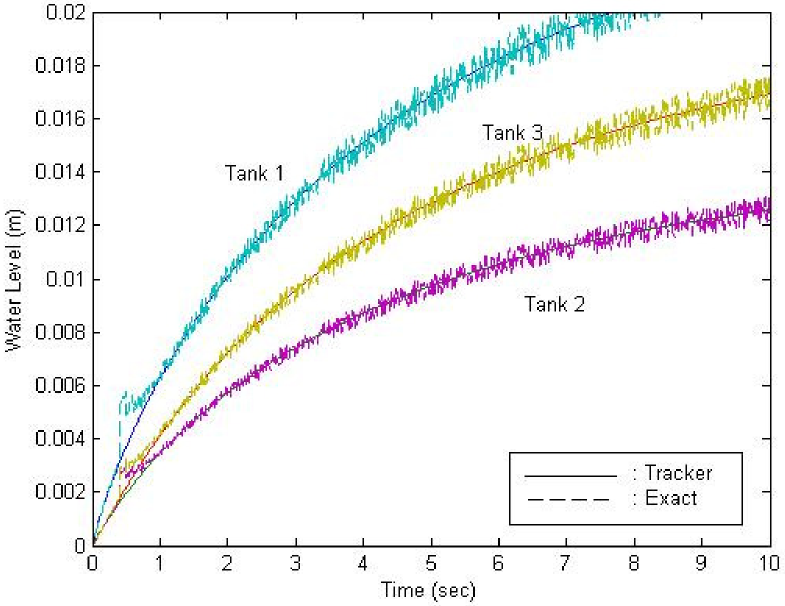


Figure 10 Output Comparison between Tracker and Exact Values (Fault Free)

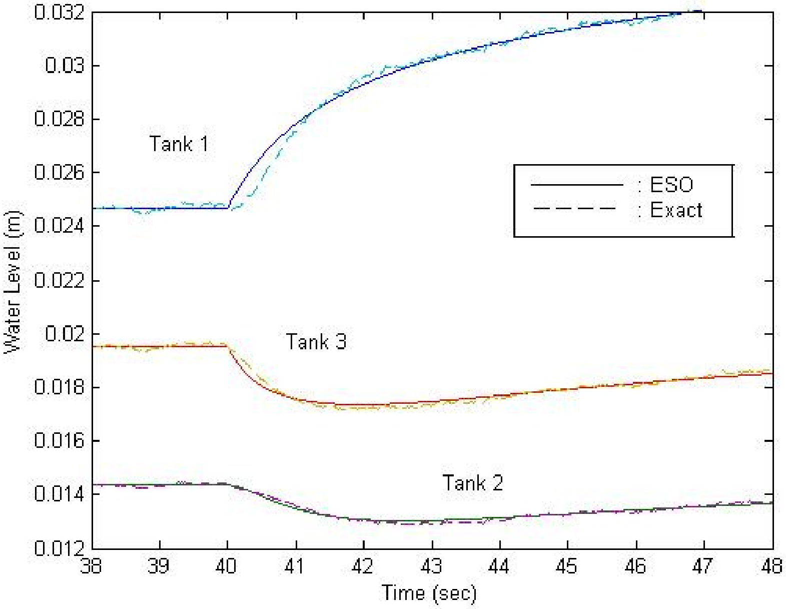


Figure 11 Output Comparison between ESO and Exact values (Fault at t=40s)

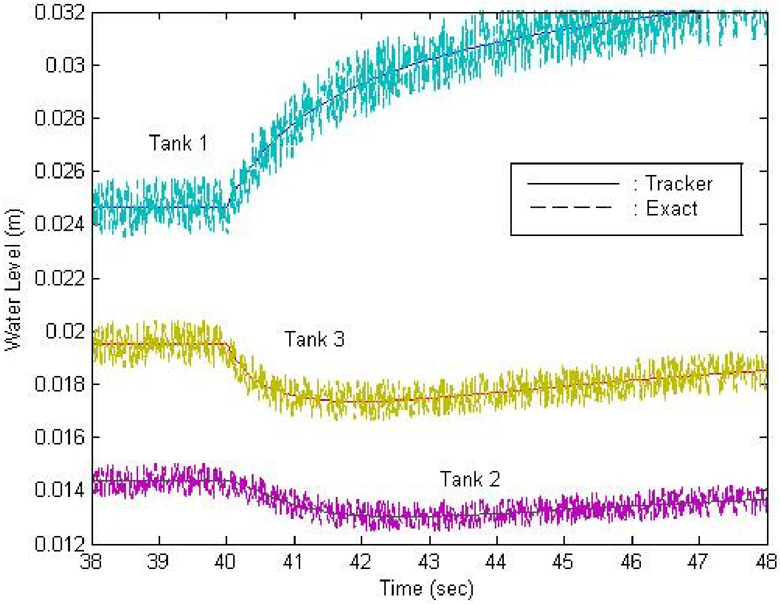


Figure 12 Output Comparison between Tracker and Exact Values (Fault at t=40s)

With the noise, the ESO could still estimate the system output while the tracker was not available to filter the noise which caused the disaster of estimation.

3.4 Fault Detection of ESO and $\alpha-\beta-\gamma$ Tracker

Without Noise

When using the ESO, a fault is considered detected if the abrupt change of general system dynamics (Δf_i , $i=1,2,3$) exceeds the predetermined value. Figure 13 clearly shows the abrupt change of each tank's water level when a fault occurred at $t=40s$.

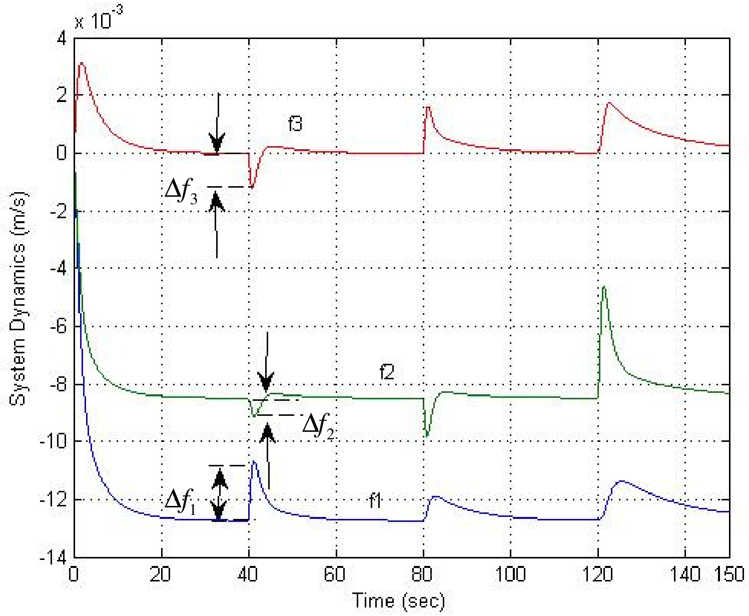


Figure 13 System dynamics from ESO of three-tank system for fault detection

When using the tracker, however, fault detection scheme is different because there is no "Δf" in its formulation. In this case, a fault is considered detected if the change of the water levels (Δy_i, $i = 1, 2, 3$) exceeds the predetermined value. Figure 14 shows the detected faults based on observing the Δy_i.

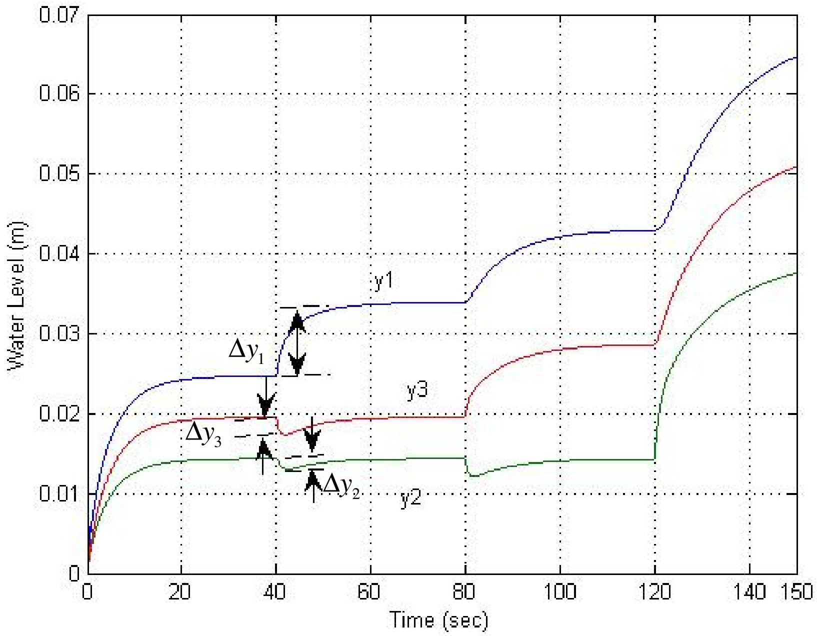


Figure 14 Water levels from tracker of three-tank system for fault detection

Although in the simulation, the tracker has a better rate of convergence and estimation accuracy, the Δy_i are not as profound as the Δf_i. Besides, the tracker is very sensitive to the measurement noise, thus it is not recommended as a method for fault detection. The self-reconfiguration is to quickly adjust the system inputs by means of an effective control scheme as soon as a fault is detected. The control goal is to restore the water level of each tank to its original state if possible.

With Noise

When 5% white noise has been added to the system, Figures 15 and 16 will give the plots for fault detection.

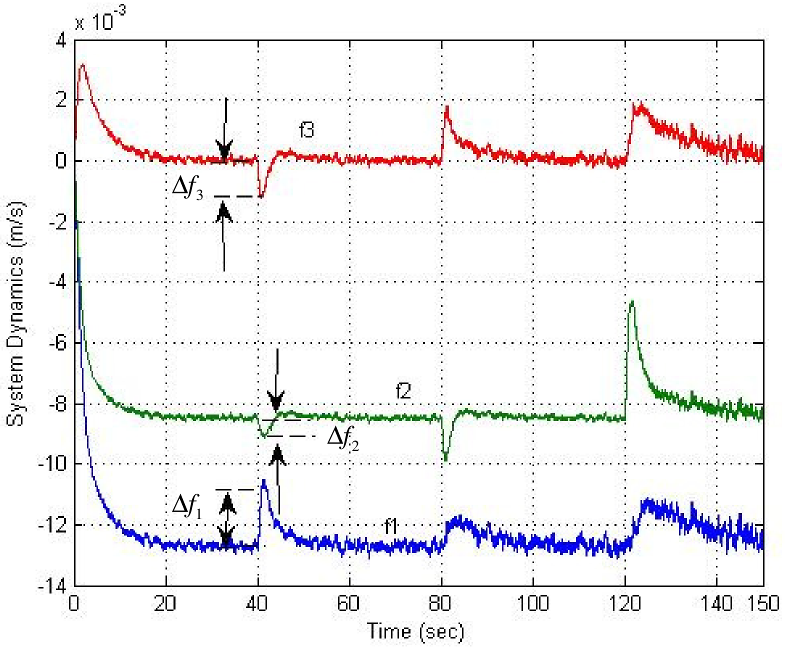


Figure 15 System dynamics from ESO of three-tank system for fault detection

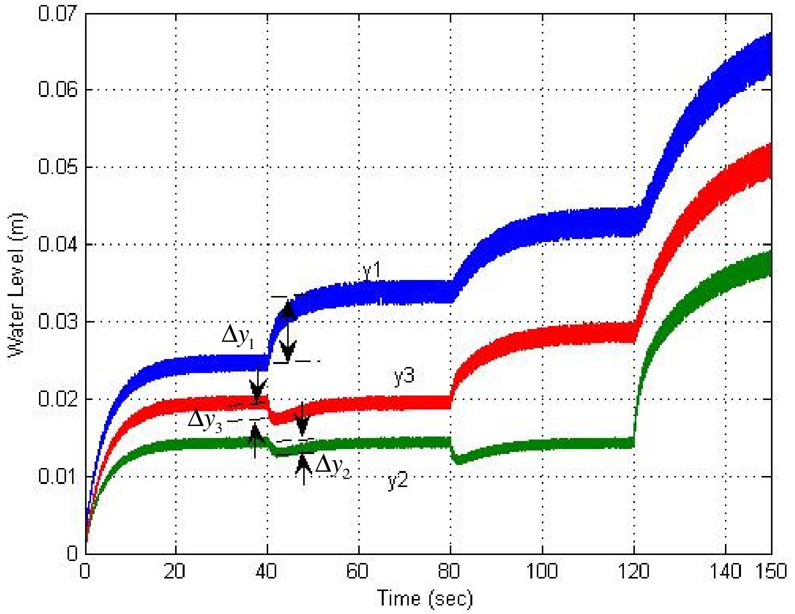


Figure 16 Water levels from tracker of three-tank system for fault detection

Figures 15 and 16 show the fault detection of the ESO and the Tracker while there is noise existed. It could be clearly seen that the Tracker would be harder to used for fault detection with such noise, however, the ESO has the ability to filter the noise, which rarely influenced the fault detection of the ESO.

CHAPTER IV

CONTROL DESIGN METHODS

It is desirable to perform control for self-reconfiguration as soon as a fault or faults is detected. In this chapter, the following four control schemes are proposed and their performances are compared.

Scheme 1: Active Disturbance Rejection Control (ADRC)

Scheme 2: Tracker-based Feedback Control

Scheme 3: Fuzzy Logic Control

Scheme 4: Tracker-based PID Control

4.1 Active Disturbance Rejection Control (ADRC)

Active Disturbance Rejection Control, known as ADRC, was originally proposed by Han in [9,16-17] for nonlinear control. Later, Gao [10] simplified the control law and the number of tuning parameters to tune. The ADRC is a relatively new design methodology that uses a very simple model, an integrator or a double integrator for a first-order or second-order system. For this kind of controller design, any nonlinear or time-varying part is treated as disturbance and rejected after an estimation being made. The result is a high performance control system that is tuned only with two parameters: the control loop bandwidth and the observer bandwidth. In essence, only one parameter (observer bandwidth) needs to be tuned because the control loop bandwidth can be related to the observer bandwidth.

The ADRC is built by using the feedback states that can be observed by the ESO. For a general second-order system, the dynamic equation can be written as

$$\ddot{y} = f(y, \dot{y}, w, t) + b_0 u \tag{4.1}$$

where f is the general system dynamics containing disturbance and uncertainty. The basic idea is to find an estimation of f, called $\hat{f}$, and use it in the control law as

$$u = (u_0 - \hat{f}) / b_0 \tag{4.2}$$

Substituting equation (18) into (17), gives an integral system

$$\ddot{y} = (f - \hat{f}) + u_0 \approx u_0 \tag{4.3}$$

which can be easily controlled by a Proportional-Derivative (PD) controller as

$$u_0 = k_p(r - \hat{y}) - k_d\dot{\hat{y}} \tag{4.4}$$

where K_p and K_d are the proportional gain and derivative gain, respectively. r is referenced output value, $\hat{y}$ is the estimated output via the ESO and $\dot{\hat{y}}$ is estimated time derivative of the estimated output. With the observer being properly designed, the control law can be expressed as

$$u = \frac{k_p(r - \hat{y}) - k_d\dot{\hat{y}} - \hat{f}}{b_0} = \frac{k_p(r - z_1) - k_d z_2 - z_3}{b_0} \tag{4.5}$$

where the states z$_1$, z$_2$, and z$_3$ represent the estimated system output $\hat{y}$, its derivative $\dot{\hat{y}}$, and estimated system dynamics $\hat{f}$. The proportional gain and derivative gain can be selected as $k_p = \omega_c^2$, and $k_d = 2\xi\omega_c$, where ω_c and ξ are the control loop bandwidth and damping ratio. Critical damping (i.e. ξ = 1) is chosen to avoid system oscillations. The ω_c is usually chosen as $\frac{1}{5}$ to $\frac{1}{3}$ of ω_o.

By relating ω_c to ω_o, the controller can be easily designed when the observer bandwidth ω_o is properly tuned. An nth-order plant with unknown dynamics and external disturbance can be written as

$$y^{(n)} = f(t, y, \dot{y}, \cdots, y^{(n-1)}, u, \dot{u}, \cdots u^{(n-1)}, w) + bu \tag{4.6}$$

Rewriting the plant to the state space model form

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \vdots \\ \dot{x}_n = x_{n+1} + b_0 u \\ \dot{x}_{n+1} = \dot{f} \\ y = x_1 \end{cases} \tag{4.7}$$

or

$$\begin{cases} \dot{x} = Ax + Bu + Ef \\ y = Cx \end{cases} \tag{4.8}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}_{(n+1)\times(n+1)}, B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ b_0 \\ 0 \end{bmatrix}_{(n+1)\times 1}, C = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \end{bmatrix}_{1\times(n+1)}, E = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}_{(n+1)\times 1}$$

The ESO can be constructed as

$$\begin{cases} \dot{z} = Az + Bu + L(y - \hat{y}) \\ \hat{y} = Cz \end{cases} \tag{4.9}$$

where $z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \\ z_{n+1} \end{bmatrix}, L = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \\ \beta_{n+1} \end{bmatrix}$.

The observer bandwidth, ω_o can be designed by using a parameterization method by placing all the observer poles at $-\omega_o$, which can be written as [10]

$$s^n + \beta_1 s^{n-1} + \cdots + \beta_{n-1} s + \beta_n = \left(s + \omega_0\right)^n \tag{4.10}$$

The parameters in L can then be determined from Equation (4.10). In the case of n=3, $\beta_1 = 3\omega_o$, $\beta_2 = 3\omega_o^2$, $\beta_3 = \omega_o^3$. In the case of n=2, $\beta_1 = 2\omega_o$, $\beta_2 = \omega_o^2$. In the case of n=1, $\beta_1 = \omega_o$.

With the observer proper designed, the ADRC control law for a nth-order plant can be designed as

$$u = \frac{k_p(r - \hat{y}) - k_{d_1}\dot{\hat{y}} - \cdots - k_{d_{n-1}}\hat{y}^{(n-1)} - \hat{f}}{b_0} = \frac{k_p(r - z_1) - k_{d_1} z_2 - \cdots - k_{d_{n-1}} z_n - z_{n+1}}{b_0} \tag{4.11}$$

where the controller gains are determined by setting the poles at $-\omega_c$,

$$s^n + k_{d_{n-1}} s^{n-1} + \cdots + k_{d_1} s + k_p = \left(s + \omega_c\right)^n \tag{4.12}$$

where ω_c is also chosen as $\frac{1}{5}$ to $\frac{1}{3}$ of ω_o.

The following content of stability analysis is quoted from [18]. The state observer is

$$\begin{aligned} \dot{z} &= Az + Bu + L(y - \hat{y}) \\ \hat{y} &= Cz \end{aligned} \tag{4.13}$$

The control law is

$$u = (-z_3 + u_0)/b \tag{4.14}$$

$$u_0 = k_p(r - z_1) - k_d z_2 \tag{4.15}$$

where r is the set point, the controller tuning is further simplified with $k_d = 2\omega_c$ and $k_p = \omega_c^2$, where ω_c is the closed-loop bandwidth [10].

Stability Analysis

Gao [18] performed the ADRC's stability analysis. It is repeated below. Assuming the system dynamics $f(y, \dot{y}, w, t)$ is completely unknown, and $e = x - z$, where x is the system output, is the tracking error in the observer. So we get

$$\dot{e} = A_e e + d \tag{4.16}$$

with

$A_e = A - LC = \begin{bmatrix} -\beta_1 & 1 & 0 \\ -\beta_2 & 0 & 1 \\ -\beta_3 & 0 & 0 \end{bmatrix}$, and $d = Eh$. Assuming the observer gain in (4.13) is chosen so that A_e is Hurwitz, the observer error, e, for the ESO is bounded for any bounded h. When the above lemma is generalized to the dynamic system described by

$$\dot{\eta} = M\eta + g(\eta) , \qquad (4.17)$$

where $\eta \in \Re^n$ and $M \in \Re^{nxn}$, the state η in (4.17) is bounded if M is Hurwitz and $g(\eta)$ is bounded. The above combining lemmas give that the ADRC design of (4.13), (4.14) and (4.15) yields a BIBO stable closed-loop system if the observer (4.13) itself and the state feedback control law (4.15) for the double integrator are stable, respectively.

4.2 Tracker-based Feedback Control

The $\alpha - \beta - \gamma$ tracker presented in Chapter III showed its superiority of rate of convergence and estimation accuracy over the ESO. To take advantage of that, a new control scheme called "Tracker-based Feedback Control" is proposed. In this scheme, the system output y is measured and tracked, and the general system dynamics f is calculated with the exact system equations by means of the tracker. The control law is the same as the ADRC.

4.3 Fuzzy Logic Control

Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory, which was proposed by Lotfi Zadeh in 1965, to deal with approximate reasonings. Fuzzy logic then had been introduced to processing industry, traffic control and house hold applications in the 1970’s. In fuzzy logic, membership functions are used to represent fuzzy sets.

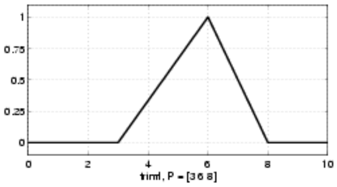
Fuzzy Sets

Fuzzy sets contain elements that have degrees of membership. A fuzzy set is a set without a crisp, clearly defined boundary. In fuzzy logic, a fuzzy set could explain the degree to which it belongs to the set, that is unlike the conventional set. The characteristic function can be denoted as any value from 0 to 1. For example, if X is a collection of objects denoted generically by x, then a fuzzy set A in X is defined as a set of ordered pairs: $A = \{(x, \mu_A(x)) \mid x \in X\}$ (4.18)

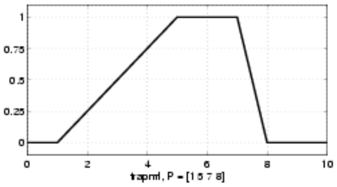
where $\mu_A(x)$ is called the membership function (MF for short) for the fuzzy set A.

Membership Functions

As the equation (4.18) showed, the membership function maps each element of X to a membership value between 0 and 1. It represents the degree of truth as an extension of valuation and associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. Once the functions are inferred, scaled, and combined, they are defuzzified into a crisp output which drives the system. There are many types of membership functions, Figure 17 shows eleven types of membership functions that are most commonly used in the fuzzy logic system design. The triangular membership function has been used in this thesis.



trimf



trapmf

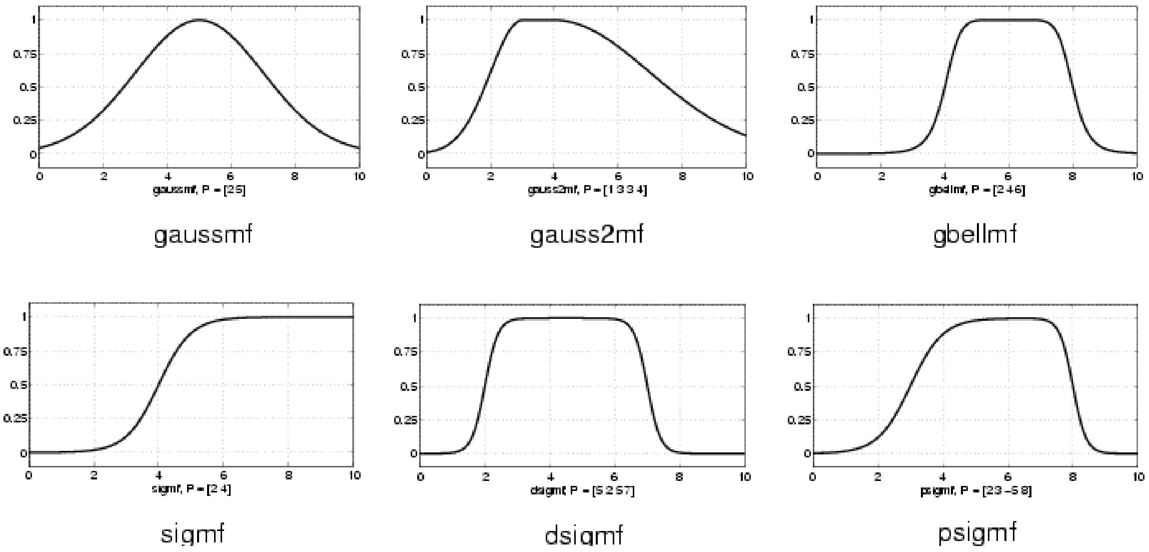


Figure 17 Commonly used membership functions

When triangular membership function is defined, three parameters need to be determined. The mathematical expression for triangular membership function specified parameters a, b, and c can be expressed as Equation.

$$trimf(a,b,c) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases} \tag{4.19}$$

The input and output each has its own membership function.

Logical Operations

Logical operations are operations using fuzzy operators. These operations are generalization of crisp set operations. The basic three operations are NOT (fuzzy complements), AND (fuzzy intersections), and OR (fuzzy unions), as Equations showed below, respectively.

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{4.20}$$

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \tag{4.21}$$

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \tag{4.22}$$

Fuzzy If-Then Rules

Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic. The if-then rules are used to formulate the conditional statements that comprise fuzzy logic. A single fuzzy if-then rule assumes the form

If X=A Then Y=B

where A and B are linguistic values defined by fuzzy sets on the ranges (universes of discourse) X and Y. The 'If' part is called the antecedent or premise, while the 'Then' part is called the consequence or conclusion.

In short, fuzzy logic control is accomplished by fuzzifying the input variables, executing the fuzzy if-then rules, and defuzzifying its output variables to crisp values to used as control signals.

4.4 Tracker-based PID Control

To compare control performance, another control scheme called "Tracker-based PID Control" is presented. The Proportional-Integral-Derivative (PID) controller is most widely used in industry. Over 85% of all dynamic controllers are of the PID variety. The

PID controller calculates an error as the difference between the reference value and the measured one, and attempts to minimize the error by adjusting the process control inputs.

The PID control configuration is shown in Figure 18.

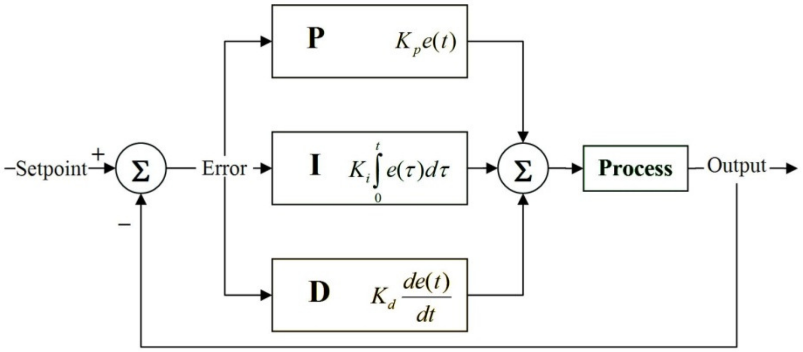


Figure 18 Schematics of PID controlled system

As shown, the control signal flowing into “process” is the sum of three terms. Each term is a function of the tracking error. The term P represents the proportional error, the term I represents the integral error, and the term D represents the derivative error. Each of these three works independently.

Although the PID control is widely used, tuning the three gains is laborious. Some tuning methods have been created for this and some tuning software were designed for the PID control. This study uses Ziegler-Nichols Rule, which is the conventional PID tuning method. The control applications are to be presented in the next chapter.

CHAPTER V

APPLICATIONS

In this chapter, three control applications are given. They are (1) active engine vibration isolation system, (2) three-tank system, and (3) MEMS gyroscope. The following four control schemes are employed and compared with the given applications.

(1) Fuzzy logic control, ADRC and PID control are applied to the active engine vibration isolation system.

(2) Fuzzy logic control, ADRC, Tracker-based Feedback Control and Tracker-based PID are applied to the three-tank system. The response time and noise filtering are also compared.

(3) ADRC and Tracker-based Feedback Control are applied to the MEMS gyroscope.

In addition, response time and noise filtering are investigated on the MIMO three-tank system.

5.1 Control for Active Engine Vibration Isolation System

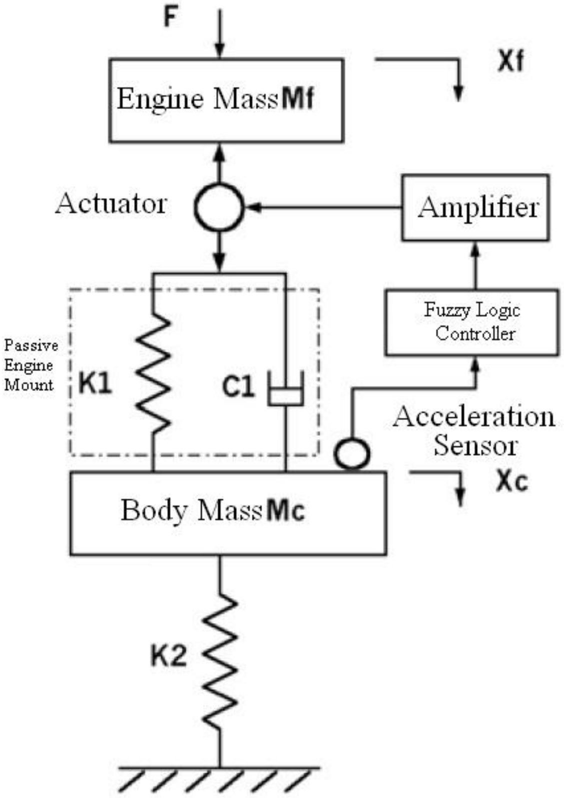


Figure 19 Engine Vibration Isolation System

The mobile active engine vibration isolation system uses a piezoelectric actuator and the passive engine mount to reduce the engine vibration [19]. Figure 19 showed the system schematics. The actuator can displace $\pm 45\mu m$ with applied voltage ± 300 volt and force of 1100N. The length-voltage function is given as

$$\Delta L = d_{33} \cdot n \cdot V + \frac{F}{K_a} \tag{5.1}$$

where K_a is the stiffness of the actuator, F is the engine force. When K_a is large enough, equation (5.1) could be rewritten as

$$\Delta L = d_{33} \cdot n \cdot V \tag{5.2}$$

where ΔL is the actuator displacement, d_{33} is the voltage-length constant, n is the number of the actuator, and V is the voltage, which is also the actual control signal. The system equations are

$$\begin{aligned} m_f \ddot{x}_f &= -k_1(x_f - x_c - L) - c_1(\dot{x}_f - \dot{x}_c - \dot{L}) + F \\ m_c \ddot{x}_c &= k_1(x_f - x_c - L) + c_1(\dot{x}_f - \dot{x}_c - \dot{L}) - k_2 x_c \end{aligned} \tag{5.3}$$

where m_f is the engine mass, m_c is the vehicle body mass, k_1 is the stiffness coefficient of the passive mount, c_1 is the damping coefficient of the passive mount, k_2 is the tire stiffness coefficient, x_f is the vertical displacement of the engine, x_c is the vertical displacement of the vehicle body, L is the vertical displacement change of the actuators, F is the external vertical force, which is considered the external disturbance. The engine vibration isolation will be controlled by the following three schemes.

During the simulation, the parameters of the system equations are chosen as: $m_f = 75kg$, $m_c = 800kg$, $k_1 = 60000N/m$, $k_2 = 300000N/m$, $c_1 = 367Ns/m$, $n = 110$, and $d_{33} = 700 \times 10^{-12} m/v$. The amplitude of the disturbance force is assumed to $100N$, and the frequency is assumed to be $50Hz$.

5.1.1 By Means of Active Disturbance Rejection Control

This controller uses actuator voltage as the control signal and tracks the vertical displacement of the vehicle body. The second equation of (5.2) is used for control. Observing from equation (5.1) knows that L can be regarded as the control signal once

d_{33}, n and V are determined. Since there is a time derivative term in equation (5.2), it is best to integrate both sides of the equation.

$$m_c\dot{x}_c = k_1(\int x_f - \int x_c - \int L) + c_1(x_f - x_c - L) - k_2\int x_c \tag{5.4}$$

Note that the first order of ESO formulation is

$$\dot{y}(t) = f + b_0 u(t) \tag{5.5}$$

The above first-order equation is then compared with the standard ESO formulation. Thus, L is $u(t)$, $b_0 = -\frac{c_1}{m_c}$ and $\dot{y}$ is $\dot{x}_c$. The general system dynamics f can then be expressed as

$$f = \frac{k_1}{m_c}(\int x_f - \int x_c - \int L) + \frac{c_1}{m_c}(x_f - x_c) - \frac{k_2}{m_c}\int x_c \tag{5.6}$$

Let x_1 be y, x_2 be f, the observer can be designed as

$$\begin{cases} \dot{x} = Ax + Bu + L(y - \hat{y}) \\ \hat{y} = Cx \end{cases} \tag{5.7}$$

where

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},\ A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} b \\ 0 \end{bmatrix},\ L = \begin{bmatrix} \omega_o^2 \\ 2\omega_o \end{bmatrix},\ C = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},\ b = b_0.$$

Since the equation is first-order, according to equation (4.10) in Chapter IV, the controller only has a proportional gain k_p, and it can be denoted as $k_p = \omega_c$. As usual, the control bandwidth ω_c is chosen as $\omega_c = \frac{1}{5}\omega_o$.

5.1.2 By Means of Fuzzy Logic Control

To use fuzzy logic control, the input and output membership function need to be constructed first. Two input variables, error and change of error are used. The fuzzy inferences output is u, which is essentially the control signal.

Membership Functions

The membership functions for error (input variable I) are shown in Figure 20.

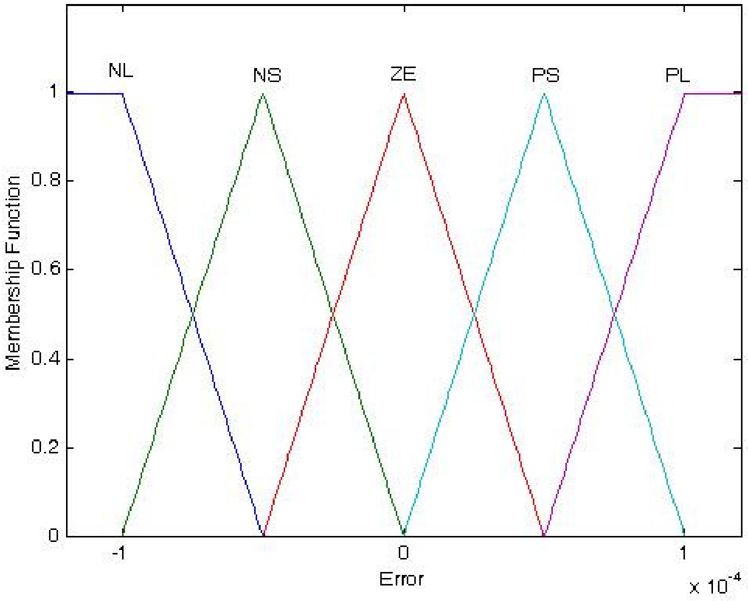


Figure 20 Membership function for error

NL (Negative Large):	<= -0.5e-5;
NS (Negative Small):	-1e-5 → 0;
ZE (Zero):	-0.5e-5 → 0.5e-5;
PS (Positive Small):	0 → 1e-5;
PL (Positive Large):	>=0.5e-5.

The membership functions for change of error (input variable II) are shown in Figure 21.

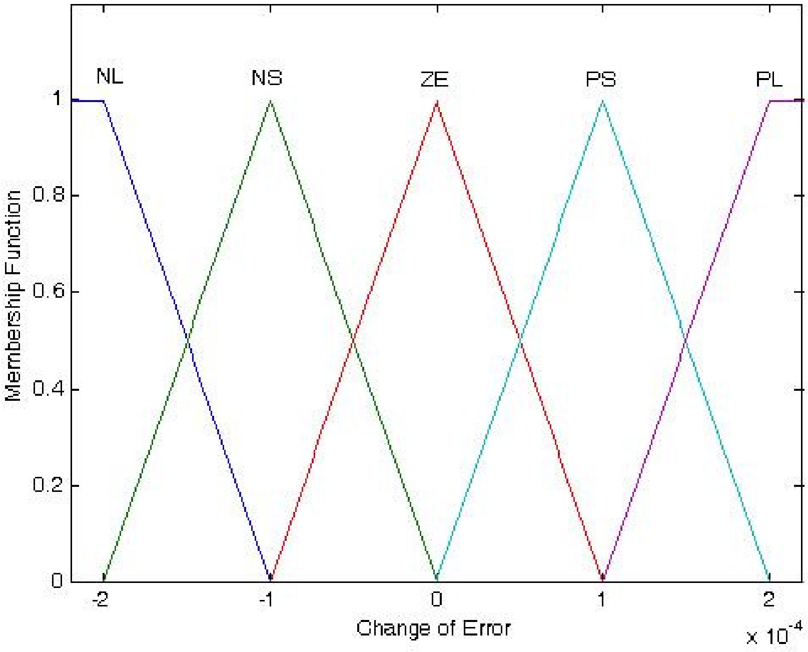


Figure 21 Membership function for change of error

NL (Negative Large):	<= -1e-4;
NS (Negative Small):	-2e-4 → 0;

ZE (Zero):	-1e-4 → 1e-4;
PS (Positive Small):	0 → 1e-4;
PL (Positive Large):	>=1e-4.

The membership functions for control signal (output variable) are shown in Figure 22.

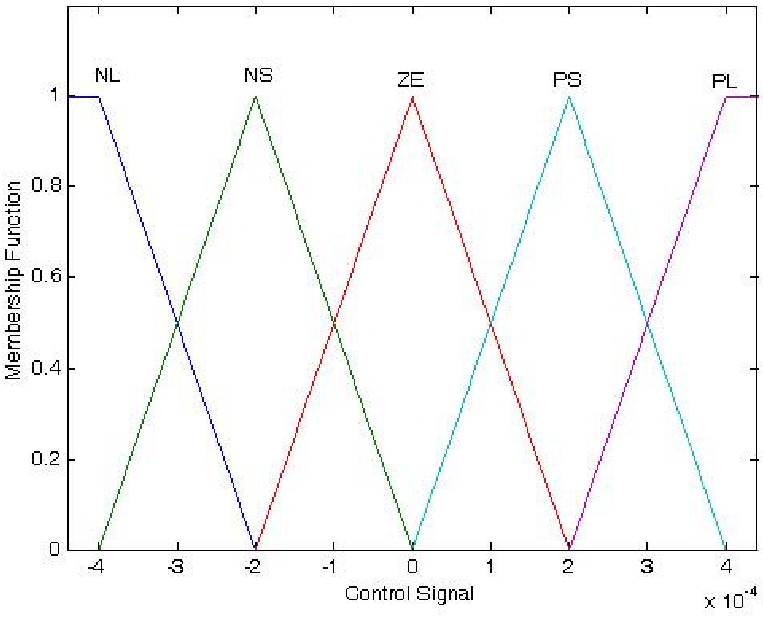


Figure 22 Membership functions for control signal

NL (Negative Large):	<= -2e-4;
NS (Negative Small):	-4e-4 → 0;
ZE (Zero):	-2e-4 → 2e-4;
PS (Positive Small):	0 → 4e-4;
PL (Positive Large):	>=2e-4.

Fuzzy rules

Table V Fuzzy Rule Table

	NL	**NS**	**Z**	**PS**	**PL**
NL	NL	NL	NS	NS	NS
NS	NL	NS	Z	Z	Z
Z	NL	NS	Z	PS	PL
PS	Z	Z	Z	PS	PL
PL	PS	PS	PS	PL	PL

NL represents negative large, NS represents negative small, Z represents zero, PS represents positive small, and PL represents positive large.

The 25 linguistic descriptions of fuzzy rules are shown as follows:

Rule 1: If (Error is NL) and (Change of Error is NL) then (U is NL).

Rule 2: If (Error is NS) and (Change of Error is NL) then (U is NL).

Rule 3: If (Error is Z) and (Change of Error is NL) then (U is NS).

Rule 4: If (Error is PS) and (Change of Error is NL) then (U is NS).

Rule 5: If (Error is PL) and (Change of Error is NL) then (U is NS).

Rule 6: If (Error is NL) and (Change of Error is NS) then (U is NL).

Rule 7: If (Error is NS) and (Change of Error is NS) then (U is NS).

Rule 8: If (Error is Z) and (Change of Error is NS) then (U is Z).

Rule 9: If (Error is PS) and (Change of Error is NS) then (U is Z).

Rule 10: If (Error is PL) and (Change of Error is NS) then (U is Z).

Rule 11: If (Error is NL) and (Change of Error is Z) then (U is NL).

Rule 12: If (Error is NS) and (Change of Error is Z) then (U is NS).

Rule 13: If (Error is Z) and (Change of Error is Z) then (U is Z).

Rule 14: If (Error is PS) and (Change of Error is Z) then (U is PS).

Rule 15: If (Error is PL) and (Change of Error is Z) then (U is PL).

Rule 16: If (Error is NL) and (Change of Error is PS) then (U is Z).

Rule 17: If (Error is NS) and (Change of Error is PS) then (U is Z).

Rule 18: If (Error is Z) and (Change of Error is PS) then (U is Z).

Rule 19: If (Error is PS) and (Change of Error is PS) then (U is PS).

Rule 20: If (Error is PL) and (Change of Error is PS) then (U is PL).

Rule 21: If (Error is NL) and (Change of Error is PL) then (U is PS).

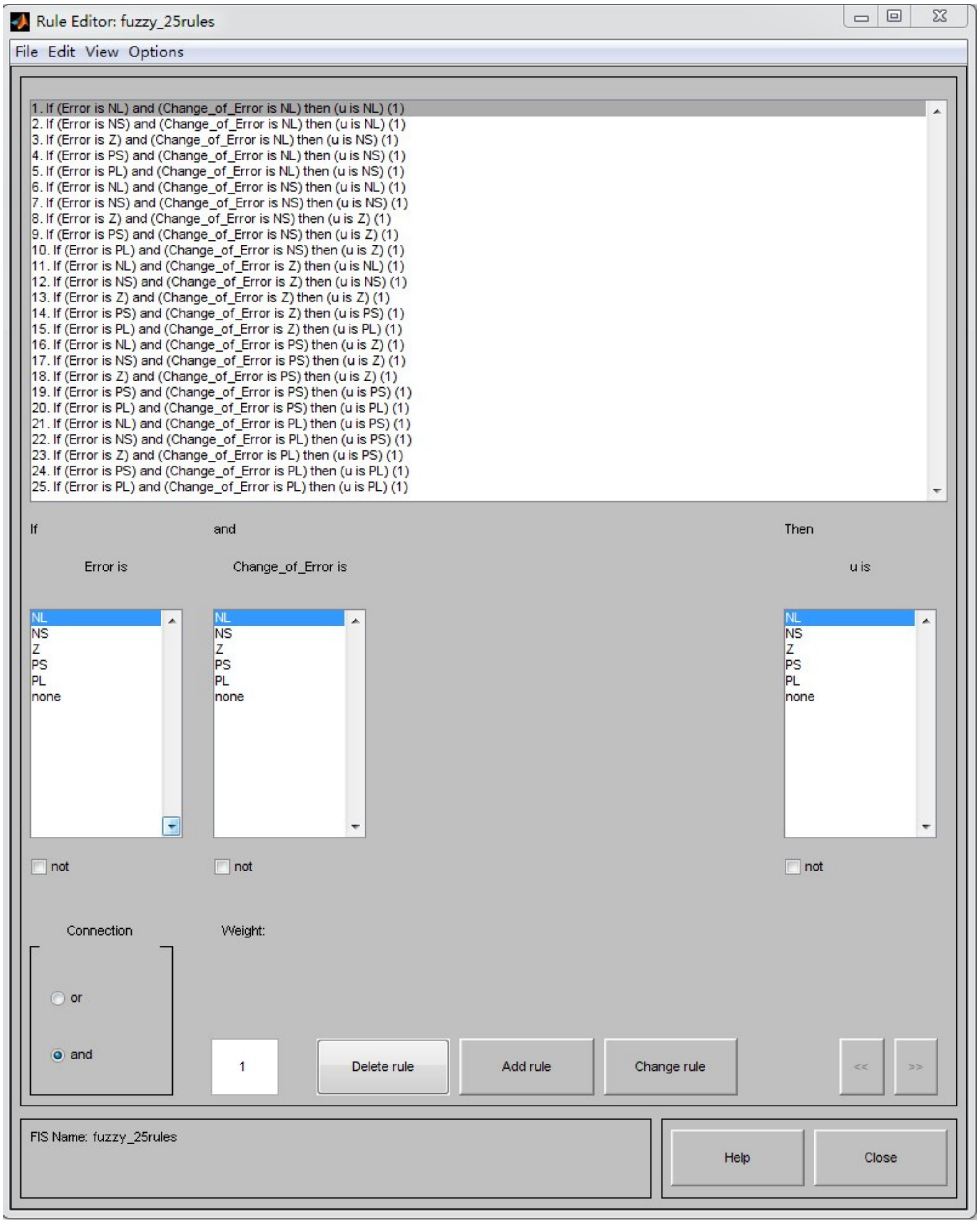
Rule 22: If (Error is NS) and (Change of Error is PL) then (U is PS).

Rule 23: If (Error is Z) and (Change of Error is PL) then (U is PS).

Rule 24: If (Error is PS) and (Change of Error is PL) then (U is PL).

Rule 25: If (Error is PL) and (Change of Error is PL) then (U is PL).

The Fuzzy Logic Toolbox in Matlab was used to develop inference system. Figure 23 shows the rule editor in the interface of the toolbox.



```
1. If (Error is NL) and (Change_of_Error is NL) then (u is NL) (1)
2. If (Error is NS) and (Change_of_Error is NL) then (u is NL) (1)
3. If (Error is Z) and (Change_of_Error is NL) then (u is NS) (1)
4. If (Error is PS) and (Change_of_Error is NL) then (u is NS) (1)
5. If (Error is PL) and (Change_of_Error is NL) then (u is NS) (1)
6. If (Error is NL) and (Change_of_Error is NS) then (u is NL) (1)
7. If (Error is NS) and (Change_of_Error is NS) then (u is NS) (1)
8. If (Error is Z) and (Change_of_Error is NS) then (u is Z) (1)
9. If (Error is PS) and (Change_of_Error is NS) then (u is Z) (1)
10. If (Error is PL) and (Change_of_Error is NS) then (u is Z) (1)
11. If (Error is NL) and (Change_of_Error is Z) then (u is NL) (1)
12. If (Error is NS) and (Change_of_Error is Z) then (u is NS) (1)
13. If (Error is Z) and (Change_of_Error is Z) then (u is Z) (1)
14. If (Error is PS) and (Change_of_Error is Z) then (u is PS) (1)
15. If (Error is PL) and (Change_of_Error is Z) then (u is PL) (1)
16. If (Error is NL) and (Change_of_Error is PS) then (u is Z) (1)
17. If (Error is NS) and (Change_of_Error is PS) then (u is Z) (1)
18. If (Error is Z) and (Change_of_Error is PS) then (u is Z) (1)
19. If (Error is PS) and (Change_of_Error is PS) then (u is PS) (1)
20. If (Error is PL) and (Change_of_Error is PS) then (u is PL) (1)
21. If (Error is NL) and (Change_of_Error is PL) then (u is PS) (1)
22. If (Error is NS) and (Change_of_Error is PL) then (u is PS) (1)
23. If (Error is Z) and (Change_of_Error is PL) then (u is PS) (1)
24. If (Error is PS) and (Change_of_Error is PL) then (u is PL) (1)
25. If (Error is PL) and (Change_of_Error is PL) then (u is PL) (1)
```

Connection: ○ or ● and

Weight: 1

FIS Name: fuzzy_25rules

Figure 23 Fuzzy logic rules editor for active engine noise isolation

To demonstrate how the fuzzy logic rules work, a few examples of randomly selected inputs (i.e. error and change of error) are listed below.

When Error is -0.3e-5 (NS), Change of Error is 0 (Z), U is -1.16e-4 (NS)

When Error is -0.8e-5 (NL), Change of Error is -0.8e-4 (NS), U is -2.38e-4 (NL)

When Error is 0 (Z), Change of Error is 1.2e-4 (PL), U is 4.83e-5 (PS)

When Error is 0.7e-5 (PL), Change of Error is -0.8e-4 (NS), U is 6.14e-5 (Z)

The results appear to follow the rules.

5.1.3 Comparison of Simulation Results in Engine Vibration Isolation

The system output is the acceleration of the vehicle body. Results are compared using ADRC, Fuzzy Logic Control and PID control. All simulations are without noise. The three coefficients of the PID controller is : $K_p = 3$, $K_i = 0.6$ and $K_d = 0.1$.

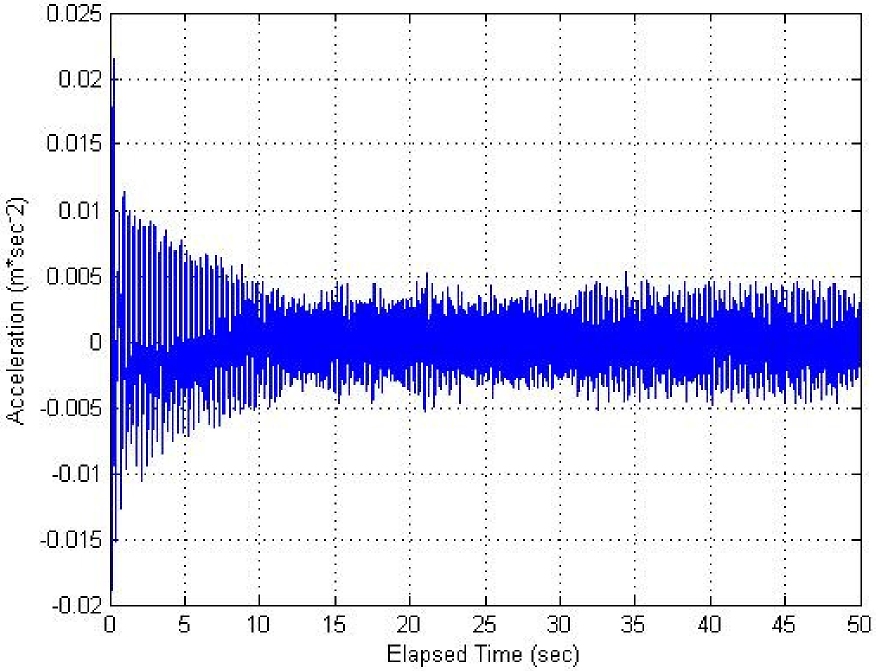


Figure 24 Output of conventional PID controlled system

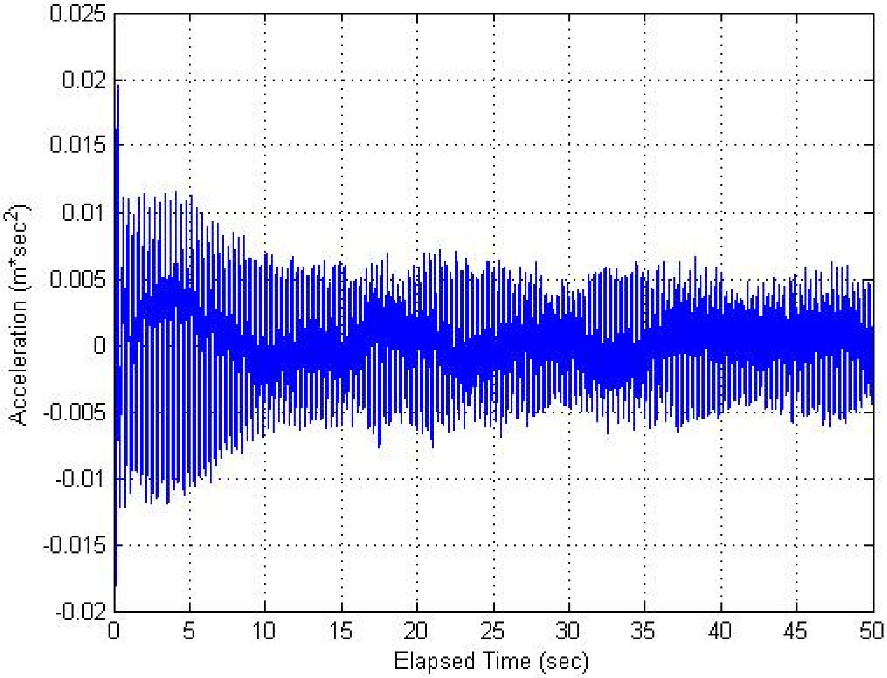


Figure 25 Output of ADRC controlled system

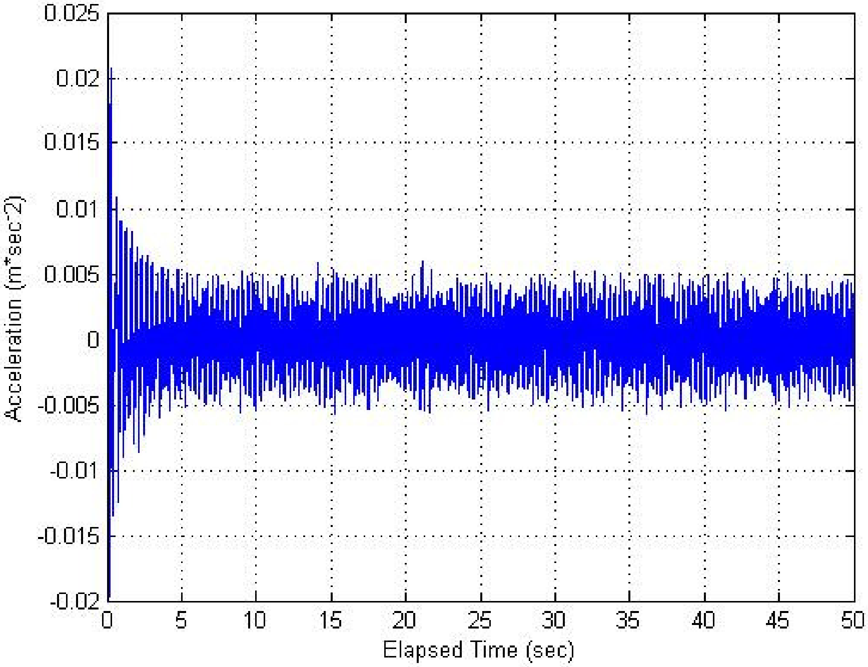


Figure 26 Output of fuzzy logic controlled system

Regardless of which control technique was used, the acceleration amplitude eventually fell within the range of ±0.005 m/sec^2. However, the fuzzy control quickly reduced the oscillation amplitude in about five seconds, whereas the PID took about 10 seconds and the ADRC took about more than 10 seconds.

For this control application, the fuzzy logic control is most effective among the three. Not only it does not require knowledge of the system model, the design of fuzzy inference system is very simple with only 25 fuzzy rules. In comparison, the PID control requires the exact knowledge of the system model, whereas the ADRC requires some knowledge of the system model, such as the order of the equations of motion.

5.2 Three-Tank System Control

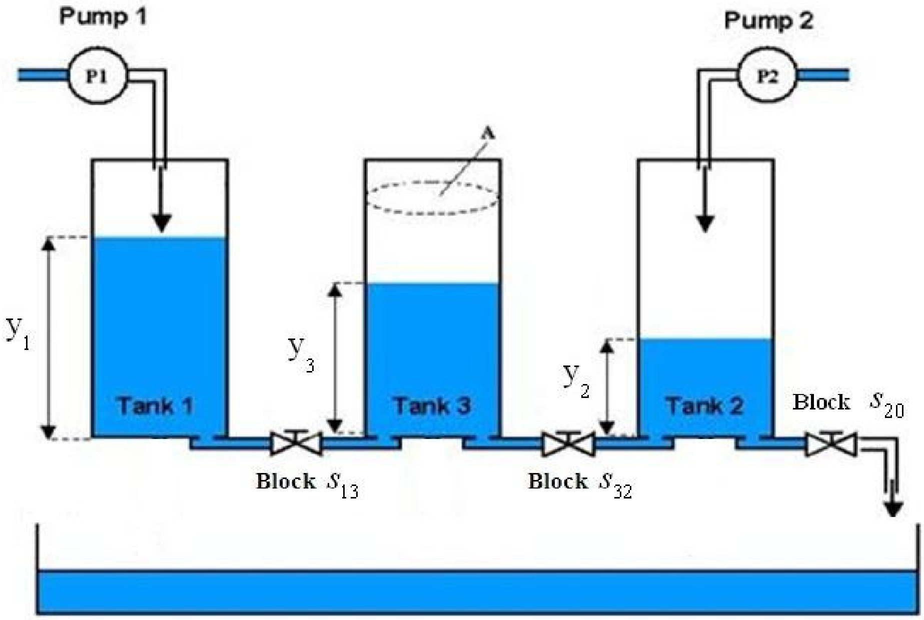


Figure 27 Three-Tank System Schematics

The nonlinear dynamic three-tank system is once again shown in Figure 27. It is a multiple-input-multiple-output (MIMO) system in which two pump rates are the inputs, and the water levels of the three tanks are the outputs. The pipe blockage is in terms of degree of fault between 0 and 1, where 0 and 1 correspond to complete blockage and no blockage, respectively.

The control for this application can be regarded as process control. The control objective is to quickly restore the three water levels as soon as a fault occurred. In other words, the control is for self-reconfiguration.

The three dynamic equations for the three-tank system are given as:

$$\begin{cases} A\dfrac{dy_1}{dt} = -a_1 s_{13} sign(y_1 - y_3)\sqrt{2g\,|y_1 - y_3|} + Q_1 \\ A\dfrac{dy_2}{dt} = -a_3 s_{32} sign(y_3 - y_2)\sqrt{2g\,|y_3 - y_2|} - a_2 s_{20}\sqrt{2gy_2} + Q_2 \\ A\dfrac{dy_3}{dt} = a_1 s_{13} sign(y_1 - y_3)\sqrt{2g\,|y_1 - y_3|} - a_3 s_{32} sign(y_3 - y_2)\sqrt{2g\,|y_3 - y_2|} \end{cases} \tag{5.8}$$

where

A : the circular cross-section area of each tank (same for all);

a_1, a_2, a_3 : the circular cross-section area of each pipe;

Q_1, Q_2 : the pump flow reates;

y_1, y_2, y_3 : the water level of each tank

s_{13}, s_{32}, s_{20} : the pipe blockage

The control objective is to track the actual water levels of tank 1 and tank 3, and minimize the difference from the desired ones. The system outputs and the control inputs are:

$$y(t) = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix},\ u(t) = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}.$$

A single fault or multiple faults can occur at any time. But, it is assumed that multiple faults do not occur at the same time.

5.2.1 By Means of ADRC

Let $u(t)$ and $y(t)$ be the systems input and output vectors, respectively, the equations can be rewritten as

$$\dot{y}(t) = f + b_0 u(t) \tag{5.9}$$

where $u(t) = \begin{bmatrix} Q_1 \\ Q_2 \\ 0 \end{bmatrix}$, $y(t) = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$, $f = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$, and $b_0 = \frac{1}{A}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

f is the system dynamics vector, which is given as

$$f_1 = \left[-a_1 s_{13} sign\left(y_1 - y_3\right)\sqrt{2g\left|y_1 - y_3\right|} \right] / A \tag{5.10a}$$

$$f_2 = \left[a_3 s_{32} sign\left(y_3 - y_2\right)\sqrt{2g\left|y_3 - y_2\right|} - a_2 s_{20}\sqrt{2gy_2} \right] / A \tag{5.10b}$$

$$f_3 = \left[a_1 s_{13} sign\left(y_1 - y_3\right)\sqrt{2g\left|y_1 - y_3\right|} - a_3 s_{32} sign\left(y_3 - y_2\right)\sqrt{2g\left|y_3 - y_2\right|} \right] / A \tag{5.10c}$$

Since this is a first-order system, the ADRC is reduced to simple proportional (p) control in which K_p is chosen as the control bandwidth, ω_c. Thus

$$u = \frac{1}{b}[k_p(y-\hat{y}) - \hat{f}] = \frac{1}{b}[\omega_c(y-\hat{y}) - \hat{f}], \tag{5.11}$$

where $\omega_c = \frac{1}{3}\omega_0$ and ω_o (observer bandwidth) is chosen as 2.

The first simulation assumed a single fault of $s_{13} = 0.6$ (i.e. 40% blockage), occurring at $t = 40$ second. Figure 28 shows that the ADRC closely follows the actual system outputs in the case of no faults.

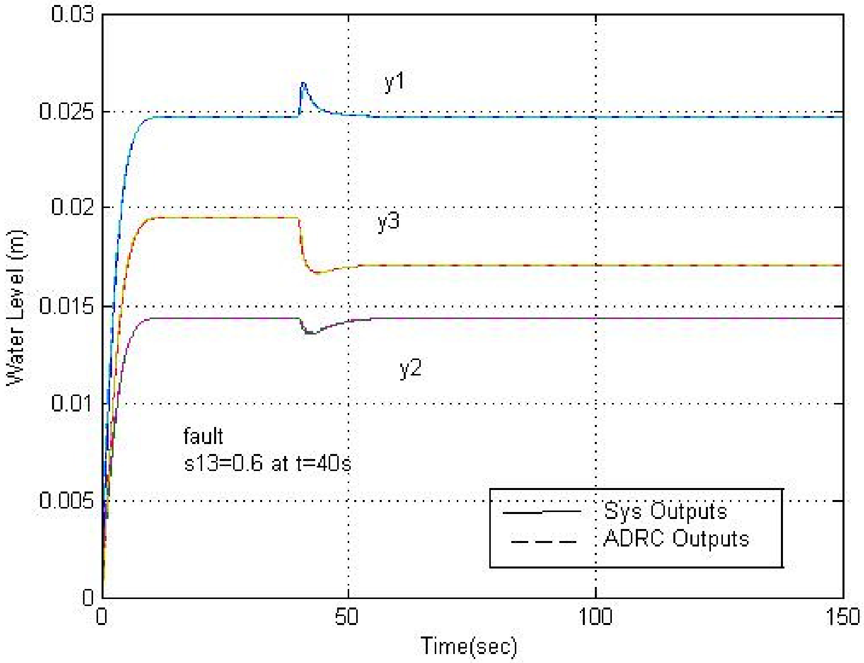


Figure 28 System outputs and the control signal from ADRC (single fault)

The second simulation is about two separate faults $s_{13} = 0.6$ occurring at $t = 40$ second and $s_{32} = 0.6$ occurring at $t = 80$ second. Figure 29 shows the ADRC responded to two faults.

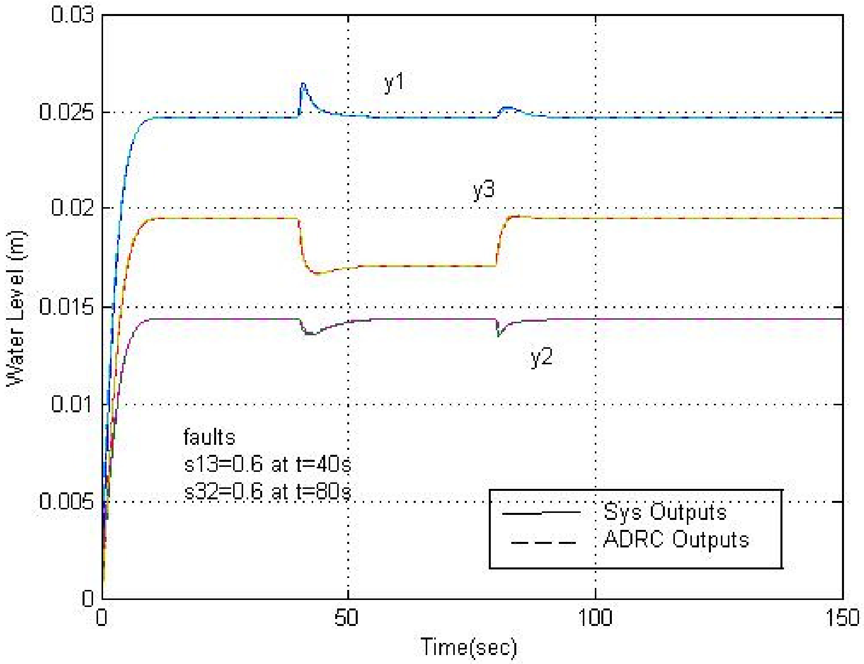


Figure 29 System outputs and the control signal from ADRC (double faults)

The third condition is that there are three separate faults, $s_{13} = s_{32} = s_{20} = 0.6$, occurring at $t = 40$ second, $t = 80$ second and $t = 120$ second, respectively. Figure 30 shows how the ADRC responded to three separate faults.

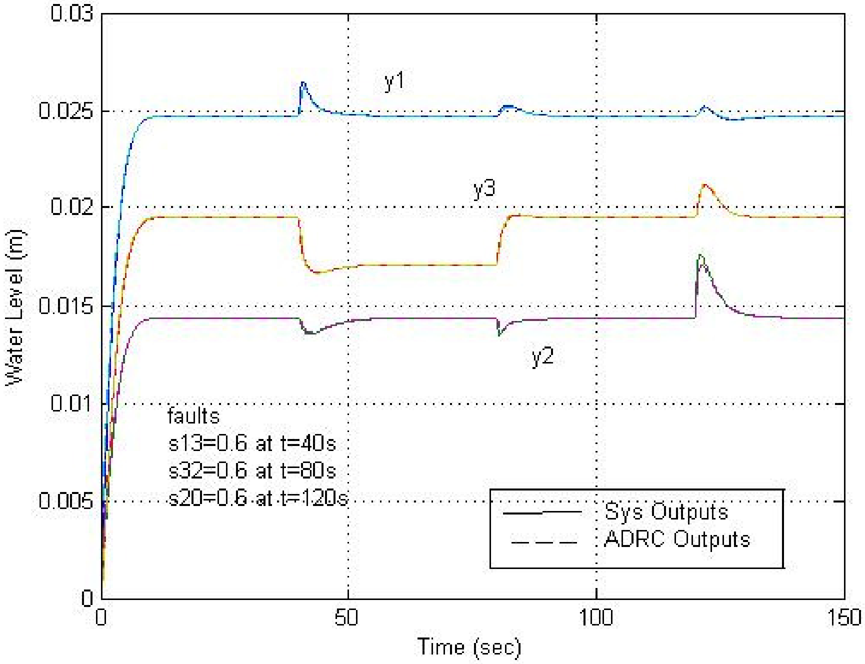


Figure 30 System outputs and the control signal from ADRC (triple faults)

5.2.2 By Means of Tracker-based Feedback Control

Like ADRC, the controller for the Tracker-based Feedback Control is almost the same, so the conditions of the simulations we used are the same as 5.2.1. The simulation results are shown in Figure 31, 32 and 33.

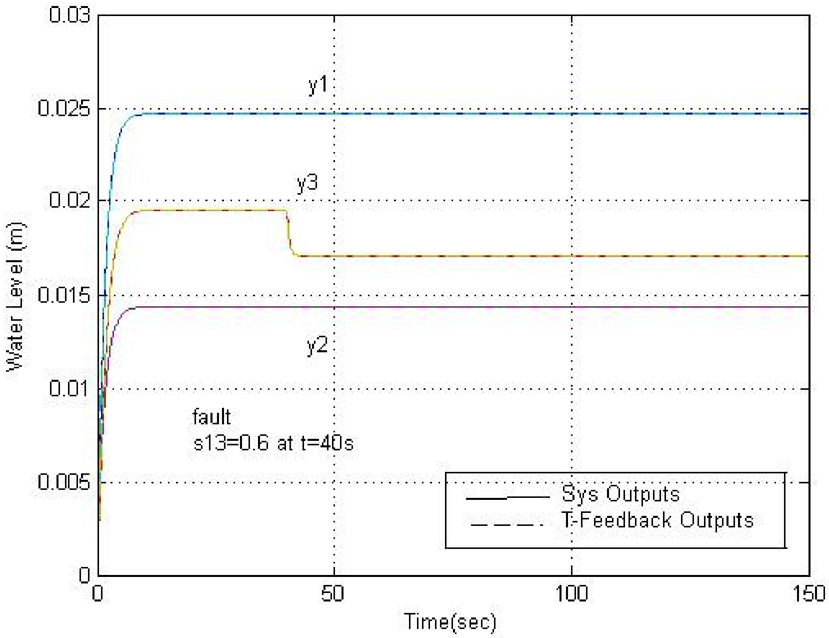


Figure 31 System outputs and the control signal from tracker-based Feedback Control

(single fault)

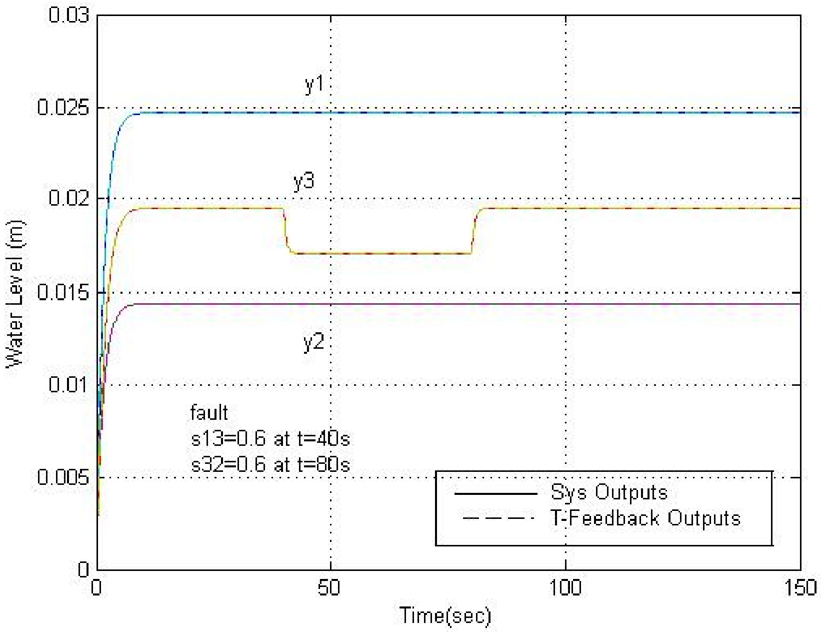


Figure 32 System outputs and the control signal from tracker-based Feedback Control

(double faults)

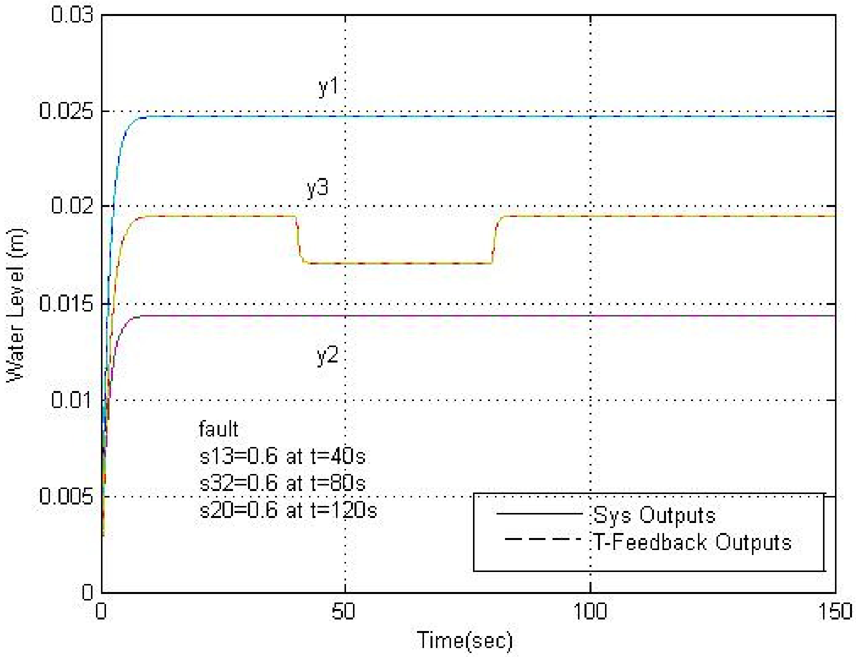


Figure 33 System outputs and the control signal from Tracker-based Feedback Control (triple faults)

5.2.3 By Means of Fuzzy Logic Control

Due to the similarity between Active Engine Vibration Isolation System and Three-Tank System, the 25 rule fuzzy controller is applied to the three-tank system. Figure 34, 35 and 36 show the simulation results for single fault, two faults and three faults, respectively.

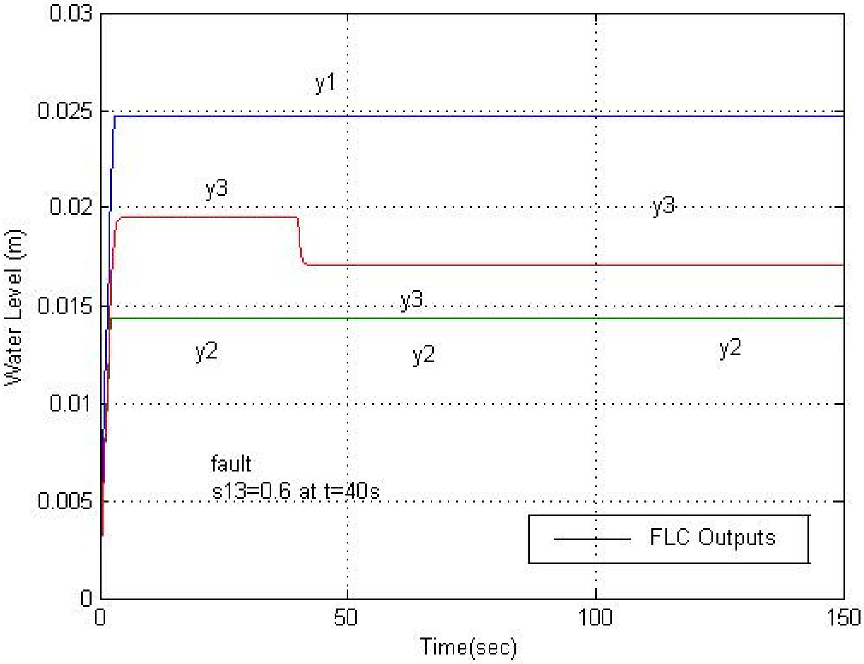


Figure 34 System outputs and the control signal from fuzzy logic control (single fault)

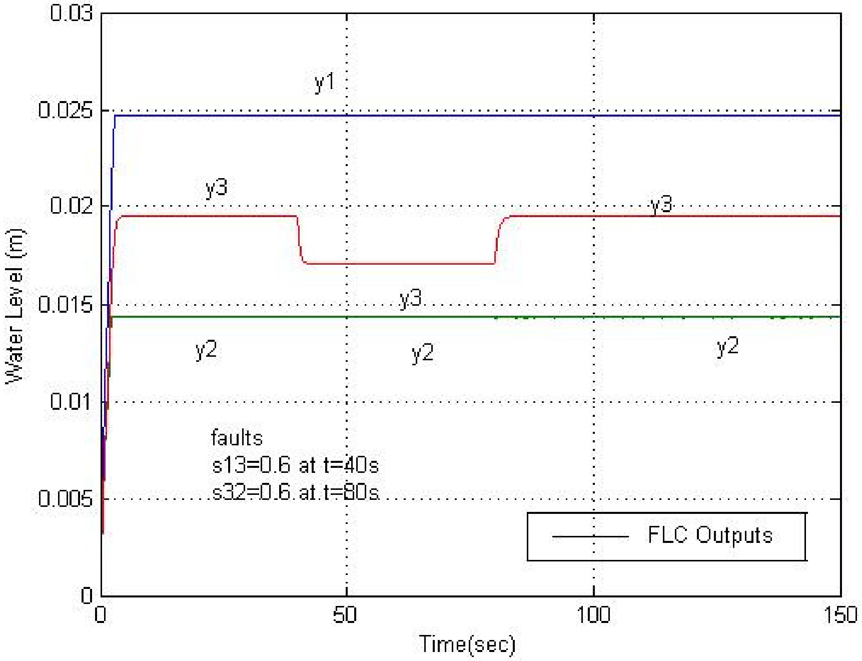


Figure 35 System outputs and the control signal from fuzzy logic control (double faults)

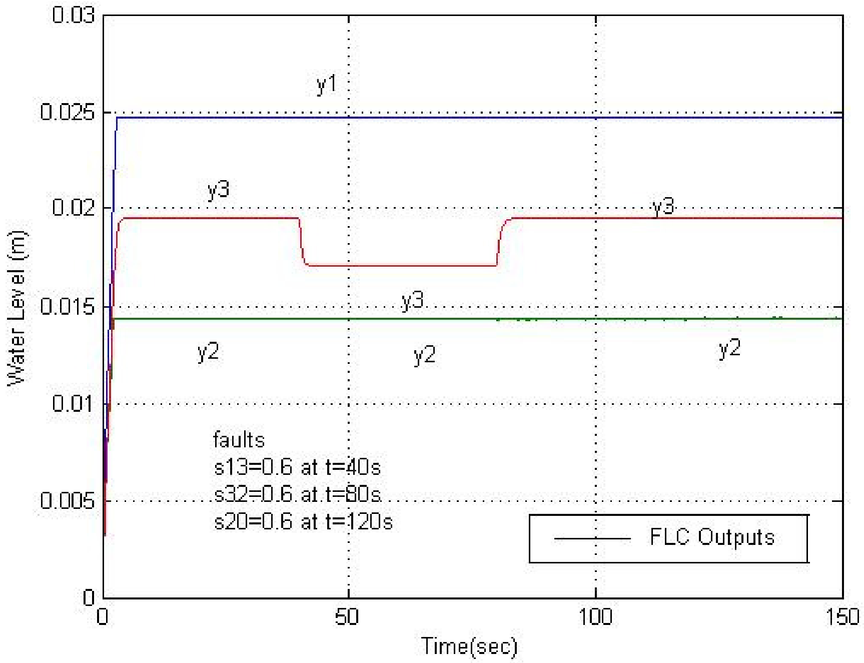


Figure 36 System outputs and the control signal from fuzzy logic control (triple faults)

5.2.4 By Means of Tracker-based PID Control

Tuning the three PID control parameters (k_p, k_i and k_d) is much more laborious than using the ADRC. The chosen parameters are $k_p = 2$, $k_i = 0$ and $k_d = 0$.

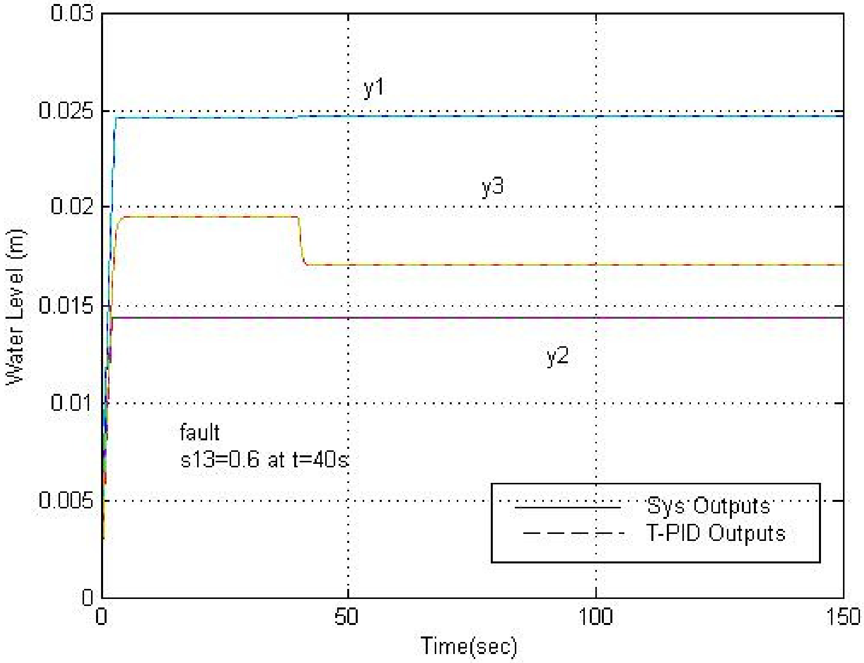


Figure 37 System outputs and the control signal from tracker-based PID (single fault)

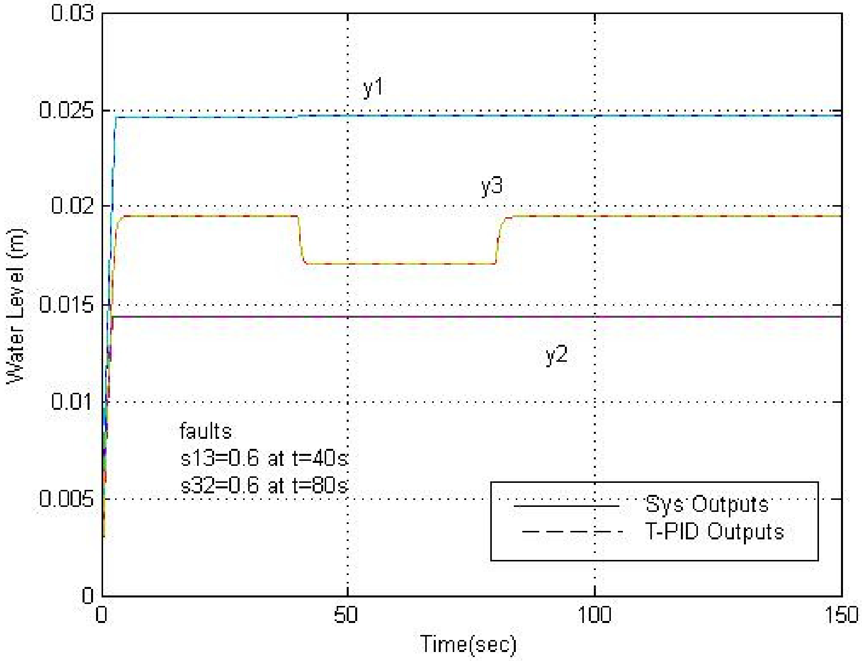


Figure 38 System outputs and the control signal from tracker-based PID (double faults)

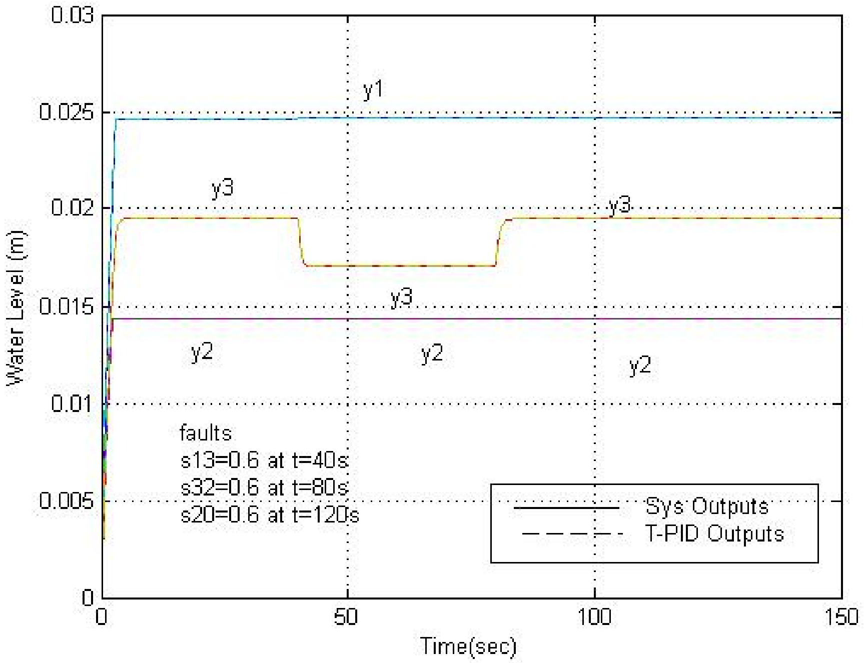


Figure 39 System outputs and the control signal from tracker-based PID (triple faults)

5.2.5 Observation on Three-Tank System Control

Figure 40, 41 and 42 show the comparisons for each two of the simulation results by employing ADRC, tracker-based Feedback Control and tracker-based PID.

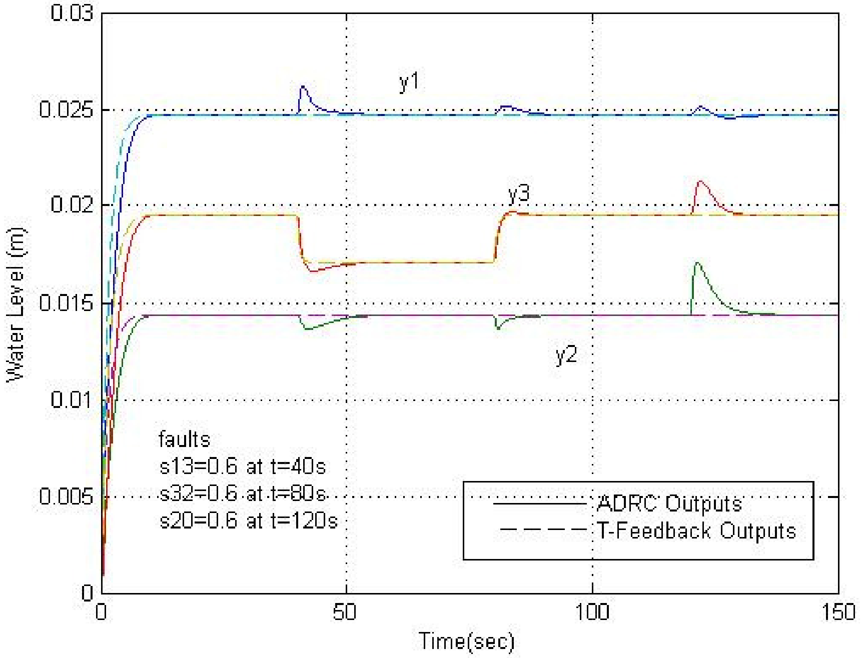


Figure40 System outputs from ADRC and tracker-based Feedback Control (triple faults)

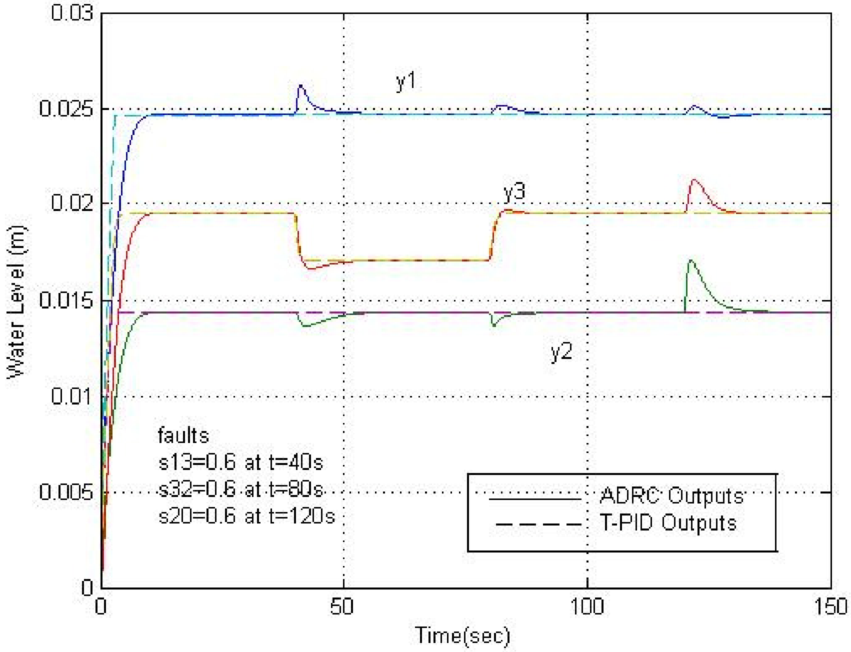


Figure 41 System outputs from ADRC and tracker-based PID (triple faults)

Figures 40 and 41 show that Tracker-based Feedback Control and Tracker-based PID perform better than ADRC did. When faults happened, they spent less time to bring the system to the predetermined stead states.

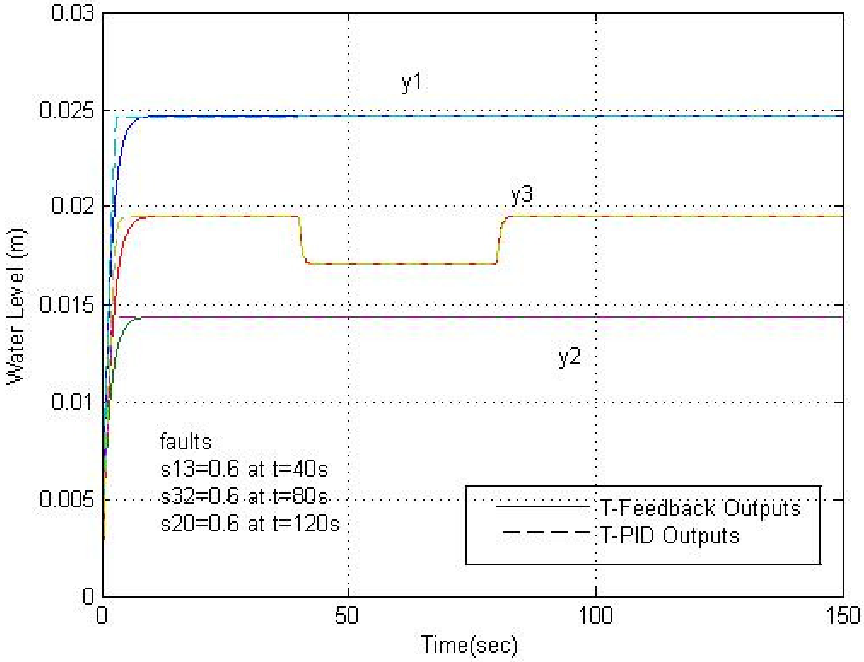


Figure 42 System outputs from tracker-based Feedback Control and tracker-based PID (triple faults)

Figure 42 shows that Tracker-based PID responded a little faster than Tracker-based Feedback Control when faults happened. The above figures show that Tracker-based PID performs better than ADRC and Tracker-based Feedback Control in terms of the low order system. However, the Tracker-based PID is more laborious for tuning.

It is observed that when the third fault occurs, the ADRC had to work hard for a while to self-reconfigure the system, while the Tracker-based Feedback Control and the Tracker-based PID almost instantly reconfigure without any effort.

5.3 MEMS Gyroscope Control

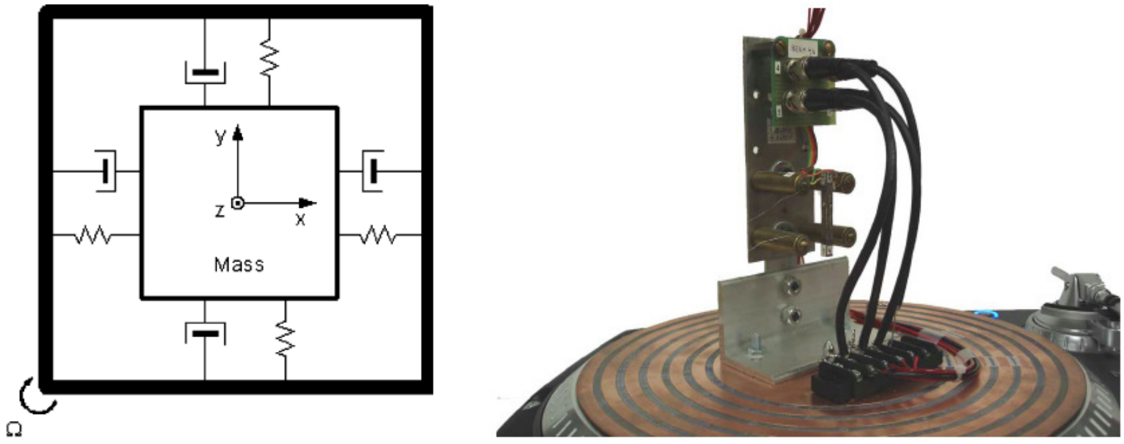


Figure 43 The MEMS gyroscope system [21]

A vibrational MEMS gyroscope is introduced, which consists of a vibrational proof mass, dampers, springs and a rigid frame [20-21]. It can be viewed as a mass suspended by elastic structures along two axes: drive axis (X axis) and sense axis (Y axis). During the vibrational movements, the Coriolis force and mechanical coupling forces transfer the energy from the drive axis to the sense axis, resulting in the vibration along the sense axis. For the two-axes driving mode, feedback control is to determine the rotation rate by measuring the vibration of the sense axis. Both controls on single-axis and two-axes are described and simulated. The effects of using three different control methods are compared.

Single-Axis Driving Mode

When only the drive axis is controlled, the vibrational MEMS gyroscope is modeled as

$$\ddot{q} + D\dot{q} + Kq + S\dot{q} = BU + CN \tag{5.12}$$

where $q(t) = \left[x(t), y(t)\right]^T$ is the displacement output vector of both axes of the gyroscope, D is the damping coefficient matrix, K represents the spring constant matrix, $S\dot{q}$ are Coriolis accelerations, in which $S(t)$ denotes the Coriolis effect matrix, B is the controller gain matrix, C is the noise gain matrix, $U(t) = \left[u_x(t), 0\right]^T$ is the control input vector, and $N = \left[N_x, N_y\right]^T$ is a mechanical-thermal noise vector. Given that the natural frequencies for both axes are matched and the sense axis is under open-loop control, we have

$$D = \begin{bmatrix} \dfrac{d_{xx}}{m} & 0 \\ 0 & \dfrac{d_{yy}}{m} \end{bmatrix},\ K = \begin{bmatrix} \dfrac{k_{xx}}{m} & \dfrac{k_{xy}}{m} \\ \dfrac{k_{xy}}{m} & \dfrac{k_{xx}}{m} \end{bmatrix},\ B = \begin{bmatrix} \dfrac{c}{m} & 0 \\ 0 & 0 \end{bmatrix},\ C = \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{c}{m} \end{bmatrix},\ S = \begin{bmatrix} 0 & -2\Omega \\ 2\Omega & 0 \end{bmatrix}$$

Define $\dfrac{d_{xx}}{m} = 2\xi\omega_n$, $\dfrac{k_{xx}}{m} = \omega_n^2$, and $\dfrac{k_{xy}}{m} = \omega_{xy}$, we rewrite the above equations and have

$$\begin{cases} \ddot{x} = -2\xi\omega_n\dot{x} - \omega_n^2 x - \omega_{xy} y + 2\Omega\dot{y} + \dfrac{c}{m}u_x \\ \ddot{y} = -2\xi\omega_n\dot{y} - \omega_n^2 y - \omega_{xy} x + 2\Omega\dot{x} + \dfrac{c}{m}N_y \end{cases} \tag{5.13}$$

where Ω is the rotation rate that the rigid frame is rotating about the rotation axis. The control objective is to force the drive axis to oscillate at a specified amplitude and the resonant frequency in the presences of parameter variations, mechanical couplings, and the mechanical-thermal noise.

Two-Axes Driving Mode

When both axes are controlled, the vibrational MEMS gyroscope is modeled as

$$\begin{cases} m\ddot{x} + d_{xx}\dot{x} + d_{xy}\dot{y} + (k_{xx} - m\Omega^2)x + k_{xy}y - 2m\Omega\dot{y} - m\dot{\Omega}y = Ku_x \\ m\ddot{y} + d_{yy}\dot{y} + d_{xy}\dot{x} + (k_{yy} - m\Omega^2)y + k_{xy}x + 2m\Omega\dot{x} + m\dot{\Omega}x = Ku_y \end{cases} \tag{5.14}$$

where x and y are the displacement outputs of drive and sense axes, respectively, Ω is the time-varying rotation rate, $2m\Omega\dot{x}$ and $2m\Omega\dot{y}$ are Coriolis forces, $m\Omega^2 x$ and $m\Omega^2 y$ are centrifugal forces, $m\dot{\Omega}x$ and $m\dot{\Omega}y$ are Euler forces, k_{xy} and d_{xy} are spring and damping coupling constants between two axes caused by mechanical imperfections, m is the mass of the MEMS gyroscope, K is the controller gain including feed-forward gain and actuator and sensor scale factors, u_x and u_y are control inputs for drive and sense axes respectively. The centrifugal force terms can be neglected or absorbed as part of the spring terms taken as unknown variations due to the reason that the rotation rate is too small comparing to the natural frequency of the system and the mass is also small (ranging from 10^{-6} kg through 10^{-10} kg).

By defining $\omega = \frac{k}{m}$, and ξ_x and ξ_y as damping coefficients of two axes respectively, we rewrite the equations as

$$\begin{cases} \ddot{x} = -2\xi_x\omega_x\dot{x} - \omega_x^2 x - \omega_{xy}y + 2\Omega\dot{y} + \dot{\Omega}y + b_x u_x \\ \ddot{y} = -2\xi_y\omega_y\dot{y} - \omega_y^2 y - \omega_{xy}x - 2\Omega\dot{x} - \dot{\Omega}x + b_y u_y \end{cases} \tag{5.15}$$

or
$$\begin{cases} \ddot{x} = f_x + b_x u_x \\ \ddot{y} = f_y + b_y u_y \end{cases} \tag{5.16}$$

where $b_x = b_y = b = \frac{k}{m}$

$$f_x = -2\xi_x \omega_x \dot{x} - \omega_x^2 x - \omega_{xy} y + 2\Omega \dot{y} + \dot{\Omega} y$$

$$f_y = -2\xi_y \omega_y \dot{y} - \omega_y^2 y - \omega_{xy} x - 2\Omega \dot{x} - \dot{\Omega} x$$

5.3.1 By Means of ADRC

Dong and Avanesian [20] successfully applied ADRC to the MEMS gyroscope system for drive mode control. Dong, Zheng and Gao [21] applied ADRC to the MEMS gyroscope system for both drive mode and sense mode control. Their control is repearted here. In addition, Tracker-based Feedback Control is presented.

The drive mode controller is $u_x = \frac{1}{b}(-\hat{f}(\dot{x}, x, d) + u_0)$. We choose $k_p = \omega_c^2$ and $k_d = 2\omega_c$, where $\omega_c > 0$. Then we have the final controller, which is

$$u_0 = \omega_c^2 (r - \hat{x}_1) - 2\omega_c \hat{x}_2 + \hat{f} \tag{5.17}$$

For the sake that MEMS gyroscopes are small, the faults are not like those in three-tank system. We will classify them as system disturbances and noises. In MEMS gyroscopes, we determine the control effect by observing the system outputs.

When there is only drive axis under the control, the system output is the displacement of drive axis. $\omega_0 = 2.5 \times 10^6$, and $\omega_c = \frac{1}{5}\omega_0$. Figures 44 and 45 show the comparisons between reference signals and system outputs of two axes from the simulation.

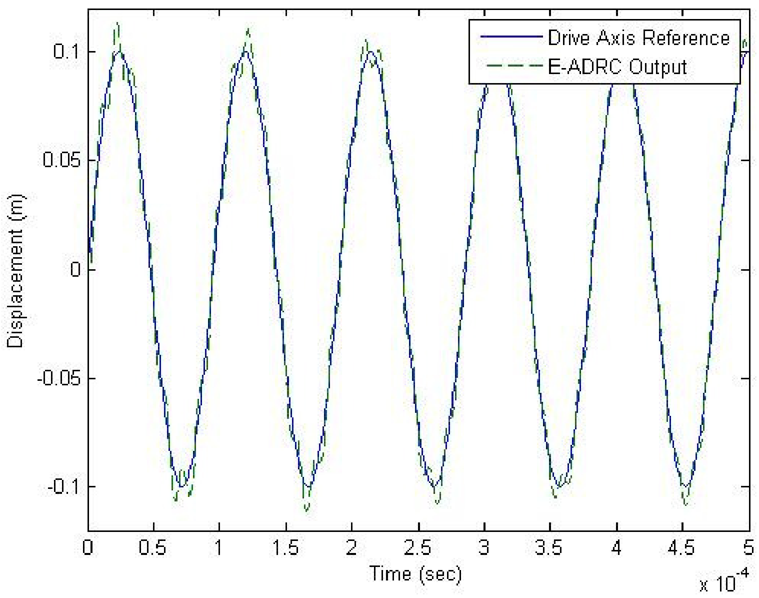


Figure 44 Drive axis (x) displacement between referenced and ADRC (Single-Axis Driving Mode)

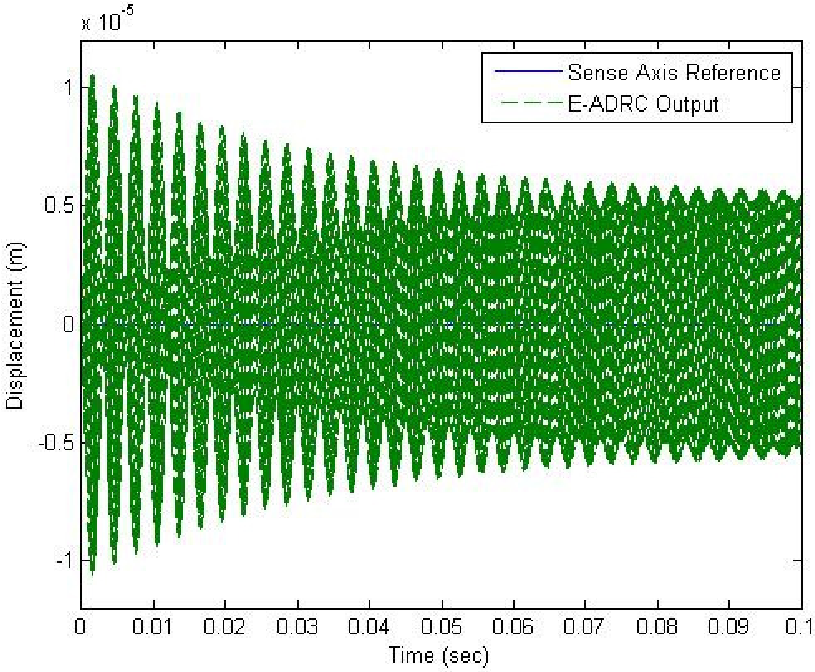


Figure 45 Sense axis (y) displacement between referenced and ADRC (Single-Axis Driving Mode)

When both axes are closed-loops, the system outputs are displacements of drive axis and sense axis. For the drive axis, the controller is same as single axis control, which is

$$u_x = \frac{\omega_c^2}{b_x}(r - \hat{x}_1) + \frac{2\omega_c}{b_x}(\dot{r} - \hat{x}_2) - \frac{1}{b_x}\hat{f} \tag{5.18}$$

In the sense axis, the output is aimed to be zero in order to calculate the rotation rate of the system. Hence, the vibration of sense axis is zero. The controller is designed as

$$u_y = \frac{\omega_c^2}{b_y}\hat{x}_1 + \frac{2\omega_c}{b_y}\hat{x}_2 - \frac{1}{b_y}\hat{f} \; . \tag{5.19}$$

Here the observer bandwidth is $\omega_0 = 2.5\times10^6$, and $\omega_c = \frac{1}{5}\omega_0$. Figures 46 and 47 show the comparisons between reference signals and system outputs of two axes from the simulation.

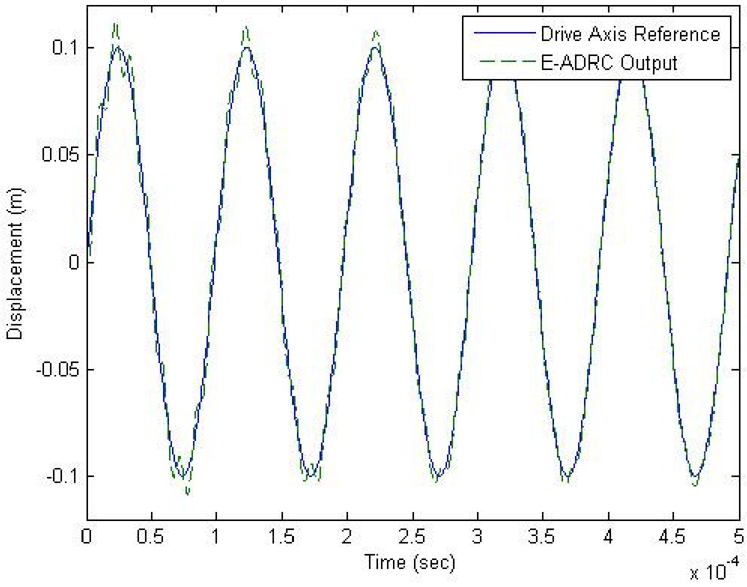


Figure 46 Drive axis (x) displacement between referenced and ADRC (Two-Axes Driving Mode)

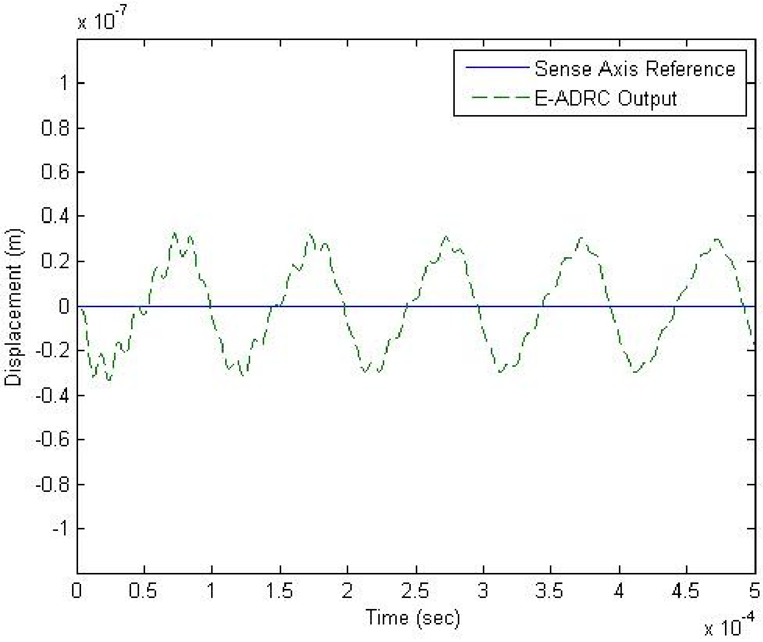


Figure 47 Sense axis (y) displacement between referenced and ADRC (Two-Axes Driving Mode)

5.3.2 By Means of Tracker-based Feedback Control

The drive mode controller is $u_x = \frac{1}{b}(-\hat{f}(\dot{x}, x, d) + u_0)$. We choose $k_p = \omega_c^2$ and $k_d = 2\omega_c$, where $\omega_c > 0$. Then we have the final controller, which is

$$u_0 = \omega_c^2(r - \hat{x}_1) - 2\omega_c\hat{x}_2 + \hat{f} \tag{5.20}$$

When there is only drive axis under the control, the system output is the displacement of drive axis. We choose $\omega_0 = 2.5\times10^6$, and $\omega_c = \frac{1}{5}\omega_0$. Figures 48 and 49 show the comparisons between reference signals and system outputs of two axes from the simulation.

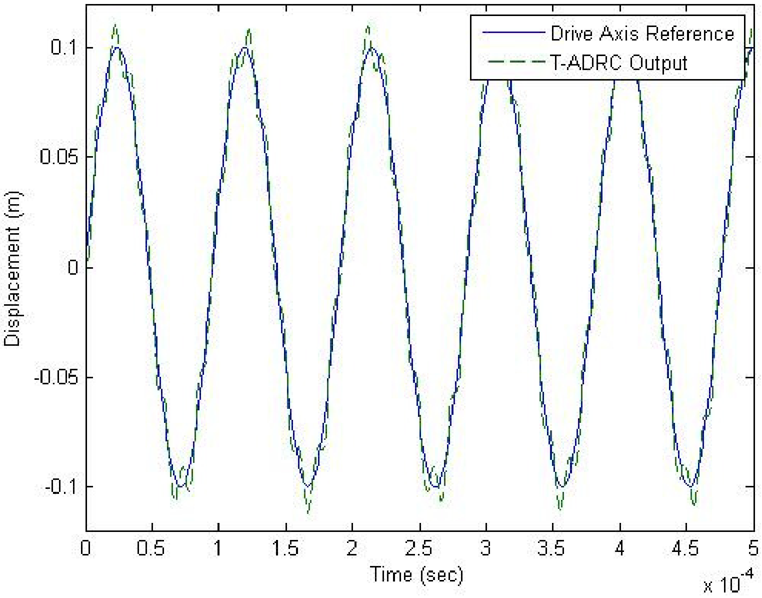


Figure 48 Drive axis (x) displacement between referenced and tracker-based Feedback Control (Single-Axis Driving Mode)

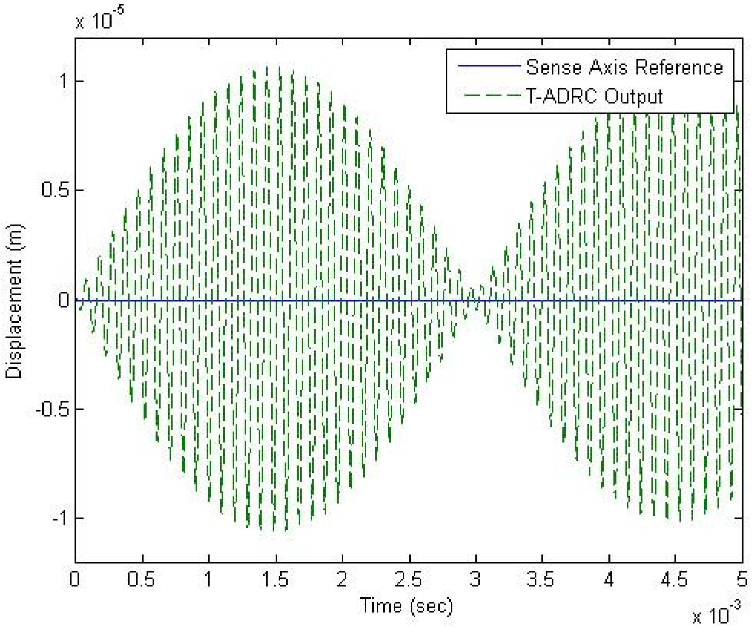


Figure 49 Sense axis (y) displacement between referenced and tracker-based Feedback Control (Single-Axis Driving Mode)

When both axes are closed-loops, the system outputs are displacements of drive axis and sense axis. For the drive axis, the controller is same as single axis control, which is

$$u_x = \frac{\omega_c^2}{b_x}(r - \hat{x}_1) + \frac{2\omega_c}{b_x}(\dot{r} - \hat{x}_2) - \frac{1}{b_x}\hat{f} \tag{5.21}$$

In the sense axis, the output is aimed to be zero in order to calculate the rotation rate of the system. Hence, the vibration of sense axis is also to be zero. The controller is designed as

$$u_y = \frac{\omega_c^2}{b_y}\hat{x}_1 + \frac{2\omega_c}{b_y}\hat{x}_2 - \frac{1}{b_y}\hat{f} \,. \tag{5.22}$$

Still, we choose $\omega_0 = 2.5\times10^6$, and $\omega_c = \frac{1}{5}\omega_0$. Figures 50 and 51 will show the comparisons between reference signals and system outputs of two axes from the simulation.

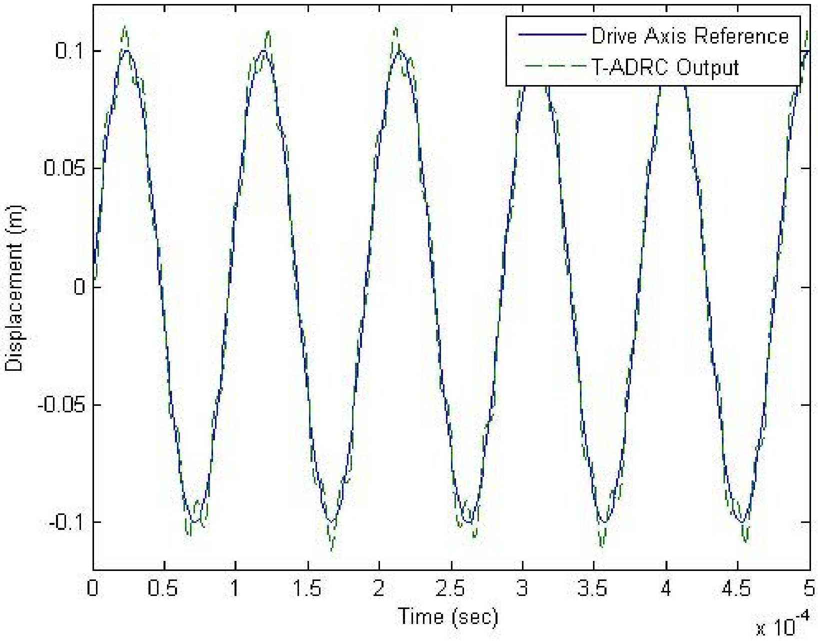


Figure 50 Drive axis (x) displacement between referenced and tracker-based Feedback Control (Two-Axes Driving Mode)

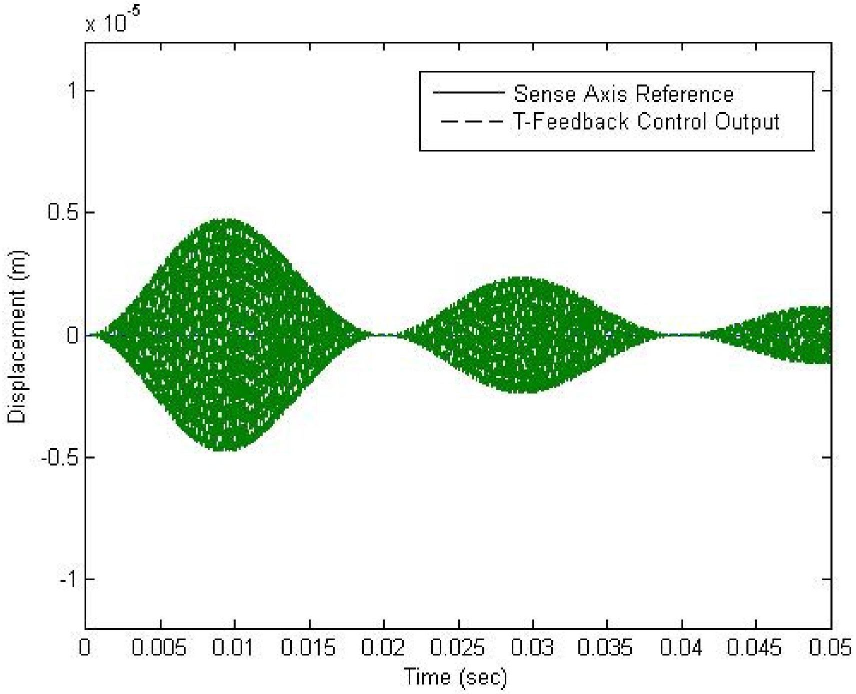


Figure 51 Sense axis (y) displacement between referenced and tracker-based Feedback Control (Two-Axes Driving Mode)

As Figure 51 shows, the sense axis displacement is getting larger, but as time goes on, it gets smaller. In contrast, the ADRC keeps the displacement of sense axis more under control.

5.3.3 By Means of Tracker-based PID

When only drive axis is closed-loop, for conventional PID, the feedback control loop is quite simple. The controller is

$$u_x = k_p(r - \hat{x}_1) + k_i(r - \hat{x}_1) + k_d(r - \hat{x}_1) \tag{5.23}$$

When both axes are closed-loops, the system outputs are displacements of drive axis and sense axis. The controller of drive axis is still

$$u_x = k_p(r - \hat{x}_1) + k_i(r - \hat{x}_1) + k_d(r - \hat{x}_1) \tag{5.23}$$

However, the controller for sense axis is

$$u_x = k_p\hat{x}_1 + k_i\hat{x}_1 + k_d\hat{x}_1\text{'} \tag{5.24}$$

For that the reference of sense axis is zero. Figures 52, 53, 54 and 55 show the simulation results of tracker-based PID control.

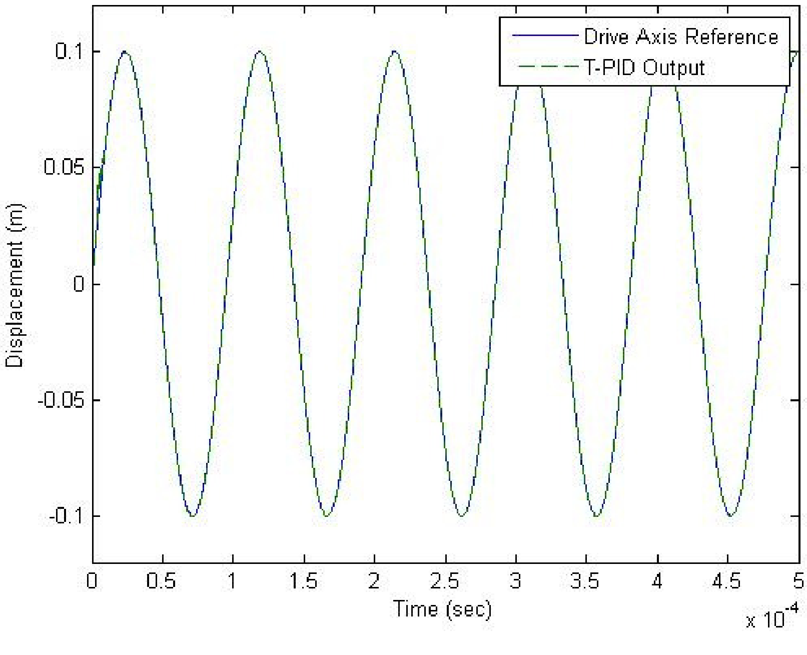


Figure 52 Drive axis (x) displacement between referenced and Tracker-based PID (Single-Axis Driving Mode)

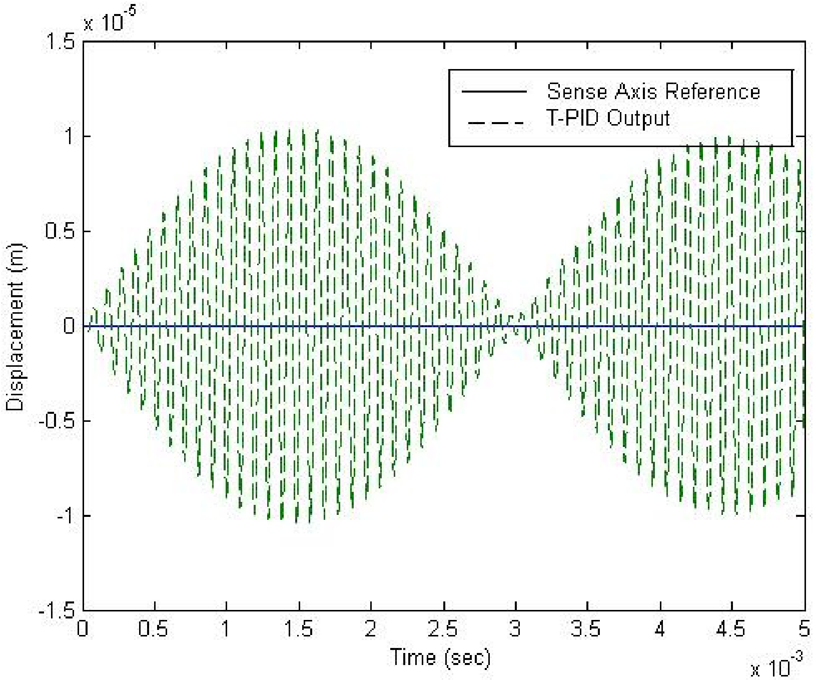


Figure 53 Sense axis (y) displacement between referenced and Tracker-based PID

(Single-Axis Driving Mode)

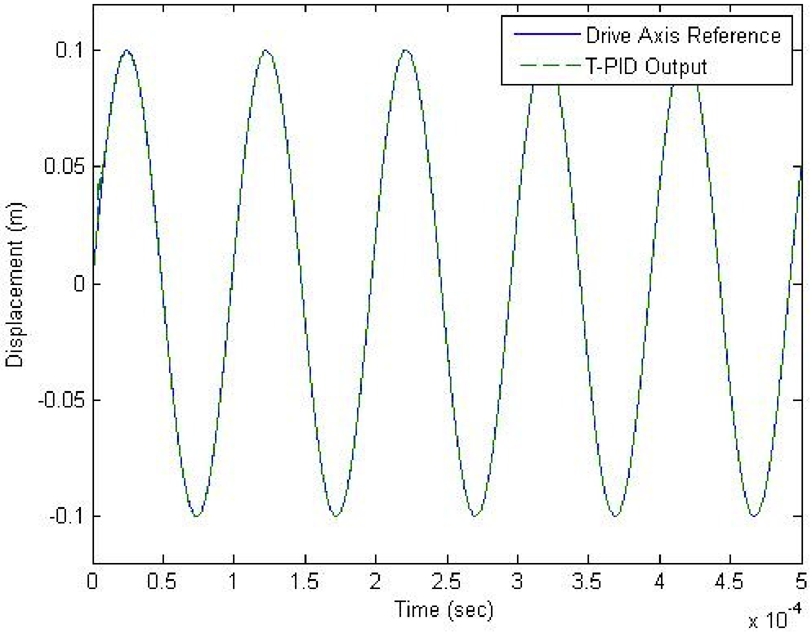


Figure 54 Drive axis (x) displacement between referenced and Tracker-based PID (Two-Axes Driving Mode)

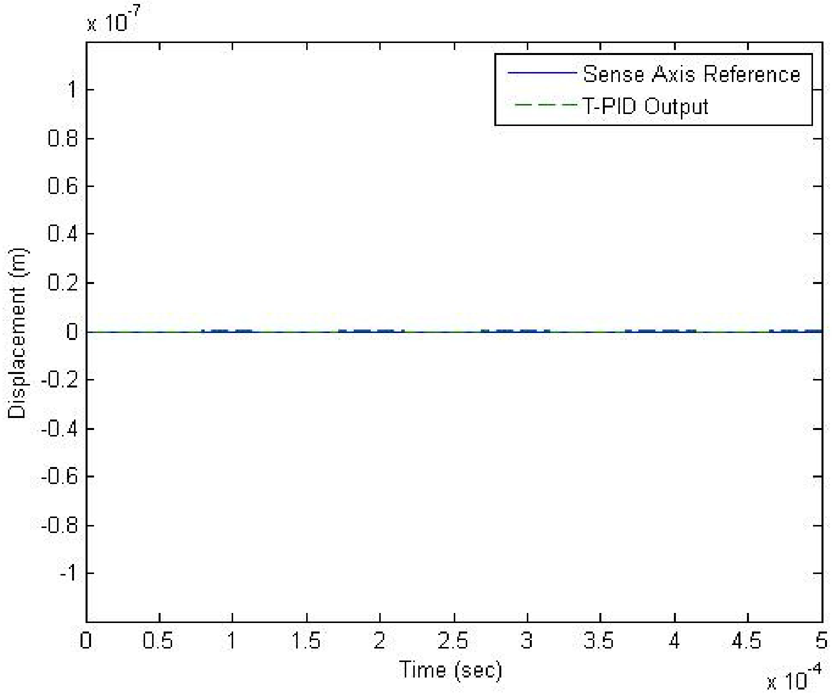


Figure 55 Sense axis (y) displacement between referenced and Tracker-based PID (Two-Axes Driving Mode)

In the MEMS gyroscope, ADRC gives a much more stabilized control result than the other two, and shows the advantage in high order system.

5.4 Comparisons

In the active engine vibration isolation system, Fuzzy Logic Control, ADRC and conventional PID are used. In the three-tank system, four control schemes are applied. In the MEMS gyroscope system, ADRC, Tracker-based Feedback Control and Tracker-based PID are used. Response time, steady-state error and noise filtering will be compared in details.

5.4.1 In Terms of Response Time and Steady-State Error

The system response time is defined a time needed to reach the system's steady state. Two kinds of responding time will be compared here, one is from the beginning to reaching the steady state with no fault occurrence, and the other is from the time of fault occurrence to the time the system becomes stabilized.

In the active engine vibration isolation system, the ADRC, the Fuzzy Logic Control and the Tracker-based PID were applied. Observing from Figure 24 to Figure 26, simulation results are obvious, that the Fuzzy Logic Control has the fastest response time, while the Tracker-based PID has the slowest one. The ADRC has a larger steady state error than the other two.

In the three-tank system, the four control methods were all applied, and three faults were assumed to occur at $t = 40$ second, $t = 80$ second, and $t = 120$ second, which were the same as $s_{13} = s_{32} = s_{20} = 0.6$, respectively. Observing from Figures 28, 31, 34 and 37, the Fuzzy Logic Control and the Tracker-based PID have almost the same fastest response time, and the Tracker-based Feedback Control has a little faster response time than the ADRC. However, when a fault occurred, Tracker-based PID and Tracker-based Feedback Control showed their ability of self-reconfiguration faster than the other two. Since any PID control takes significant time to fine tune the three control gains, Tracker-based Feedback Control is the best choice for this application.

In MEMS gyroscope system, ADRC and Tracker-based Feedback Control are applied. From Figure 44 to 55, we find that the ADRC is doing as well as the Tracker-based Feedback Control. For a complicated system like MEMS gyroscope, the two control methods are all capable of being applied. The only difference is that the ADRC has a little higher steady state error than Tracker-based Feedback Control does. But the difference is insignificant.

All in all, the Tracker-based Feedback Control and the Fuzzy Logic Control have demonstrated excellent control abilities and faster response time. They show some advantages for controlling a simple system. However, for controlling a complex system, the ADRC and the Tracker-based Feedback Control are highly recommended.

5.4.2 In Terms of Noise Filtering

In real world, disturbances exist in every system. Among the four control methods, only ADRC has the ability to filter noise and reject the disturbance. By properly choose the ESO's observer bandwidth, noise can be filtered.

In contrast, each of the other three control methods requires using a low-pass filter to the input signals. Among the three methods, the Tracker-based Feedback Control performed far better than the other two. Thus only the ADRC and the Tracker-based Feedback Control are given for comparing the ability of noise filtering.

Figures 56 and 57 show the close look of outputs when a fault occurred at $t = 40s$, when 5% noise was added to the measured data.

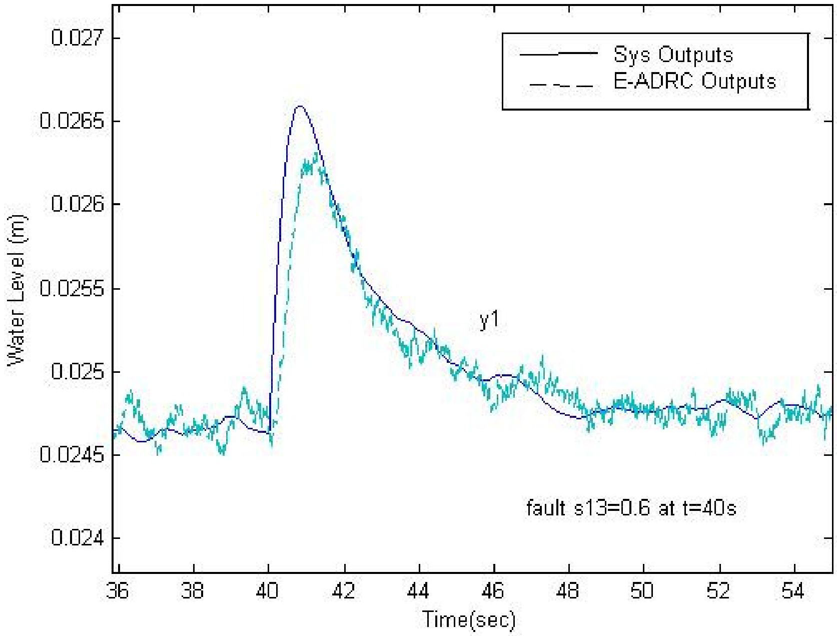


Figure 56 System output from reference and ADRC

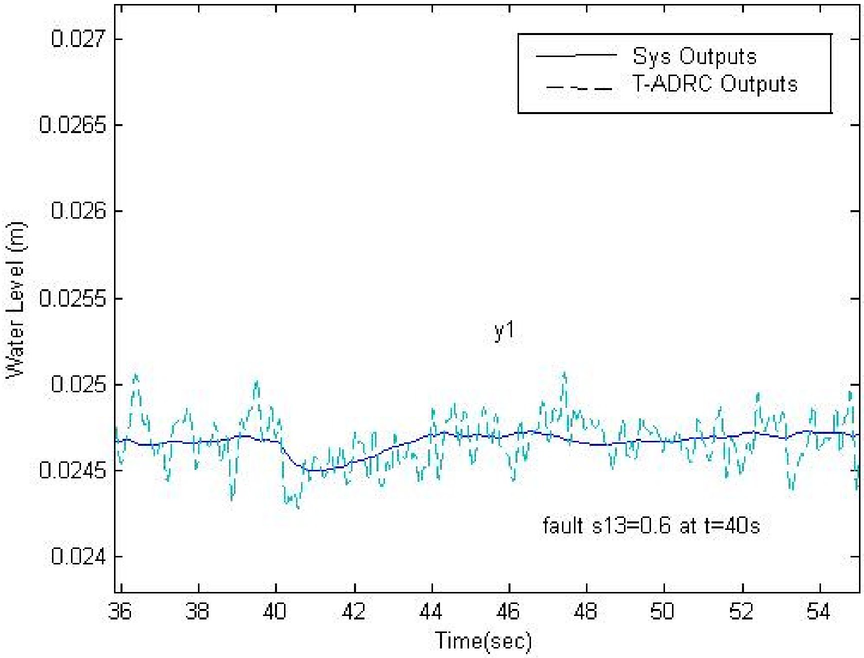


Figure 57 System output from reference and Tracker-based Feedback Control

Figures 56 and 57 show the control effects under the condition of single fault occurrence with the presence of 5% noise. During the tuning process, different filters interrupted the Tracker-based Feedback Control and caused the inaccurate control effects. Thus, although it may be observed that the Tracker-based Feedback Control has potential to be a better choice over the ADRC if the tracker can properly filter the noise, it is too laborious for choosing the filter and beaten by the ADRC on noise filtering.

CHAPTER VI

OBSERVATIONS AND CONCLUSIONS

This thesis conducted a comparative study on rate of convergence and estimation accuracy between the ESO and the $\alpha-\beta-\gamma$ tracker, while the tuning parameters are specifically chosen. The simulation results showed that the tracker outperformed the ESO under the condition that noise is free. The study was then extended to fault detection and control for self-reconfiguration. It is worthwhile to note that the ESO turned out to be a more effective method than the tracker for fault detection.

In terms of control design for self-reconfiguration, four control methods were investigated. They are Active Disturbance Rejection Control, Tracker-based Feedback Control, Fuzzy Logic Control and Tracker-based PID. To compare the control performance, the control methods were applied to three real-world applications. (1) ADRC and fuzzy logic control were applied to an active engine vibration isolation system. (2) ADRC, Tracker-based Feedback Control, fuzzy logic control and tracker-based PID were applied to a three-tank system. (3) ADRC, Tracker-based Feedback Control and tracker-based PID were applied to a MEMS gyroscope.

It has been observed and concluded that, for the active engine vibration isolation system, fuzzy logic is more effective than ADRC with specific chosen tuning parameters. For the three-tank system, Tracker-based PID performs better than the others, but more laborious for tuning. For the MEMS gyroscope, ADRC performs better than the others in controlling the high order system.

The future work will include the following recommendations. (1) Fuzzy Logic Control may be combined with Tracker-based Feedback Control to make the control more intelligence. (2) A better noise filter may be incorporated into the Tracker-based Feedback Control.

REFERENCES

[1] R. V. Beard, “Failure accommodation in linear system through self-reconfiguration,” Man Vehicle Laboratory, MIT, Cambridge, MA Rep. MTV-71-1, 1971.

[2] H. L. Jone, “Failure detection in linear system,” Ph.D. dissertation, Dept. Aero. And Astro. , MIT, Cambridge, MA, 1973.

[3] R. N. Clark, D. C. Fosth and W. M. Walton, “Detecting instrument malfunctions in control systems,” *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-11, pp. 465-473, July 1975.

[4] D. Tenne and T. Singh, “Characterizing Performance of $\alpha-\beta-\gamma$ Filters,” IEEE Transactions on Aerospace and Electronic Systems, 2002, vol. 38, No.3 pp. 1072-1087

[5] P.P. Lin, H. Singh, “Intelligent Model-Free Diagnosis for Multiple Faults in a Nonlinear Dynamic System,” IEEE/ASME Conference on Advanced Intelligent Mechatronics (AIM), Zurich, Switzerland, 2007.

[6] D. Ye, P.P. Lin and Z. Gao, “Research on Fault Diagnosis Using Extended State Observer,” *Chinese Journal of Scientific Instrument*, Vol. 29, No. 4, April 2008.

[7] A. Radke, “On Disturbance Estimation and Its Application on Health Monitoring,” Doctoral Dissertation, Department of Electrical and Computer Engineering, Cleveland State University, 2006.

[8] M. Ye, “Road Surface Condition Detection and Identification and Vehicle Anti-Skid Control,” Master Thesis, Department of Mechanical Engineering, Cleveland State University, 2008.

[9] J. Han, “A Class of Extended State Observers for Uncertain Systems,” Control and Decision, vol. 10, No.1, pp. 85-88, 1995, (in Chinese).

[10] Z. Gao, “Scaling and Parameterization Based Controller Tuning,” Proc. Of the 2003 American Control Conference, vol. 6, 4-6 June 2003, pp. 4989-4996.

[11] C. Wu, P. P. Lin, Z. Han and S. Li, “Simulation-based Optimal Design of $\alpha-\beta-\gamma-\delta$ Filter,” International Journal of Automation and Computing, 7(2), May 2010, pp. 247-253.

[12] K. Ogata, “Discrete-Time Control Systems,” Prentice-Hall, 1987.

[13] P. P. Lin and H. Singh, “Intelligent Model-Free Diagnosis for Multiple Faults in a Nonlinear Dynamic System,” IEEE/ASME international conference on Advanced intelligent mechatronics, 2007, pp. 1-6.

[14] D. Ye and P. P. Lin, “Fault Detection with Little Knowledge of System Model,” IEEE International Conference on Systems, Man and Cybernetics, 2008, pp. 1972-1977.

[15] J. Chen and R. J. Patton, “Robust Model-Based Fault Diagnosis for Dynamic Systems,” Kluwer Academic Publishers, 1999, pp.251-295.

[16] J. Han, “Nonlinear State Error Feedback Control,” Control and Decision, vol. 10, No. 3, 1995, pp. 221-225, (in Chinese).

[17] J. Han, “Auto-disturbance Rejection Control and its Applications,” Control and Decision, vol. 13, no. 1, 1998, pp. 19-23, (in Chinese).

[18] Z. Gao, “Active Disturbance Rejection Control: A Paradigm Shift in Feedback Control System Design,” Pro. Of the 2006 American Control Conference, Minneapolis, Minnesota, USA, 14-16 June, 2006.

[19] G. Sun, Y. Tian and W. Wu, “Fuzzy Control for Automobile Active Engine Vibration Isolation System,” Proceedings of 2003 Chinese Intelligent Automation Conference (I), 2003, (in Chinese).

[20] L. Dong and D. Avanesian, “Drive-Mode Control for Vibrational MEMS Gyroscopes,” IEEE transactions on industrial electronics, vol. 56, No. 4, April 2009, pp. 956-963.

[21] L. Dong, Q. Zheng and Z. Gao, “On Control System Design for the Conventional Mode of Operation of Vibrational Gyroscopes,” IEEE sensors journal, vol. 8, No. 11, November 2008, pp. 1871-1878.