POLITECNICO DI MILANO

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING
Master of Science in Automation and Control Engineering Department

Simulation-Based Benchmark Comparison of Triggering Mechanisms in Event-Based Industrial Controllers

Supervisor: Assoc. Prof. Alberto Leva

Graduation Thesis of:

Muhammet Çakıl, matricola 859007

İsmail Onur Duygun, matricola 854905

Academic Year 2017-2018

To our family and friends,

Acknowledgements

We would like to thank our supervisor Prof. Alberto Leva, who leads and guides us with his valuable thoughts at every stage of the thesis.

We also owe a debt of gratitude to our families who helped us with moral and material supports along our education life.

December 2017 Muhammet ÇAKIL, İsmail Onur DUYGUN

Contents

List of Figures

List of Tables

Abstract

In control systems, periodically taken data and/or excessive trigger frequency could cause an increase of the energy consumption of the battery of the sensor and the actuator wear in a plant due to the structure of the periodic based triggering mechanism. To deal with this problem, among others, event based control has been proposed.

This thesis firstly introduces event based control, its application areas, motivation for the event based control and comparison with the time triggered systems. Further part focused on the event based triggering mechanisms in the literature and implementation of the triggering methods and the general framework of the entire system on LabVIEW simulation environment, followed by explanation of the benchmark processes for testing the controllers and the event based structures. Then, control problems to be taken account are specified and controller parameters are calculated with a systematic method. Lastly, the built event based structure with different triggering mechanisms are executed on LabVIEW and the results are compared and analyzed.

The desired outcomes of presented thesis are to build triggering mechanisms of event based control using the LabVIEW environment by National Instruments and compare the created triggering mechanisms based on different controlling problems regarding the varying system dynamics. Especially, the response of the systems is evaluated based on the set point trajectory and disturbance rejection properties.

Sommario

Nei sistemi di controllo, il campionamento periodico dei dati a frequenza eccessiva potrebbe causare un aumento del consumo di energia della batteria del sensore e l'usura dell'attuatore a causa della struttura del meccanismo di attivazione periodico. Per affrontare (tra altri) questo problema, è stao proposto il controllo basato sugli eventi (event-based).

Questa tesi anzitutto introduce il controllo event-based, le sue aree di applicazione, la motivazione per il controllo event-based e il confronto con i sistemi time-driven (ossia guidati dal tempo). Un'ulteriore parte si concentra sui meccanismi di trigger di eventi nella letteratura, sull'implementazione dei metodi di trigger e in generale sell'intero sistema sull'ambiente di simulazione LabVIEW, seguito dalla spiegazione dei processi di benchmark per testare i controller e le strutture basate sugli eventi. Quindi, vengono specificati i problemi di controllo da tenere in considerazione e i parametri del controllore vengono calcolati con un metodo sistematico. Infine, la struttura basata su eventi costruita con diversi meccanismi di attivazione viene eseguita su LabVIEW e i risultati vengono confrontati e analizzati.

I risultati desiderati della tesi sono (i) costruire meccanismi di trigger del controllo basato sugli eventi utilizzando l'ambiente LabVIEW di National Instruments e (ii) confrontare i meccanismi di attivazione creati sulla base di diversi problemi di controllo riguardanti le diverse dinamiche di sistema. In particolare, la risposta dei sistemi è valutatain base alle proprietà di inseguimento del set point e reiezione dei disturbi.

Chapter 1

Introduction

Event Based Control

In a control loop which consisted the sensors, the controller and the actuators, to transfer the data between these components there are several transmission strategies. One of the most popular way to send the data through the network is the periodic sampling transmission [4]. In literature this type of control is known as ‘time-triggered’ or ‘fixed rate’ control. There is a solid and strong theoretical background of periodic sampling that’s why it still dominates the design of the controllers for both linear and non-linear systems. Besides the strong sides, the theory of periodic sampling has some practical issues.

Especially in some situations periodic sampling causes some undesirable outcomes due to its structure. The sampling period must be defined before the system is set up which has to stand against uncertainties. As a result, imagine the fixed sampling rate is chosen as less than the necessary rate, CPU implemented periodic sampling controller is computing the deemed actions, because of the short period, CPU works even if there is no necessity that leads waste of CPU, also leads an increase of actuator wear.

Several alternatives to the periodic sampling could be referred. One approach is the event based control strategy which has increasing popularity especially in wired and wireless control systems [2]. In recent years, industry has been in effort to convert large-scale manual control and observing systems into fully automatic systems. The purpose of this transformation is to reduce the maintenance cost. To achieve this goal, plants have been added to sensors and actuator nodes which provides observing and controlling over the plant by transferring the data between the plant and control stations. Large scale of usage of sensors and actuators come up with extra consumption of energy, especially when the data is taken periodically. In large scale systems, sensors and actuators are generally energy restraint and applied in tough environments [34]. For instance, in smart water network more than 97% of actuation assets are located underground and powered by batteries [31]. High transmission power is needed to send the required information through long-range wireless communications which causes to fast battery reduction. Moreover, the periodic sampling, sending the data and actuation leads to decrease of the network bandwidth and increase of the energy consumption [31]. The main point is to limit the sensor and control activities. Event based control theory is developed to become a solution to those problems. The event based control is also called ‘Event-Triggered Control’, ‘Aperiodic Control’, ‘Asynchronous Control’.

In an event based control system usually, when the error of the systems exceeds the certain threshold an event is triggered [32]. In an event based system the important thing is the existence of an event rather than the elapsed time, that is the logic behind the decision of when the sample must be taken.

Accurate usage of event based control can lead several benefits, those could be sum up as follows:

- Reduction of transmission: Transmission only happens when necessary, Event Based control reduces network load, decreasing the communication delays and the packet losses

- Reduction of sensor's battery consumption: Event Based control lengthen the actuator's life due to limited and necessity control actions

- Reduction of sensor's battery consumption: Most of the time sensor source of power is battery instead of electricity on the plant, reducing the number of transmission helps to less usage of the sensor and reducing the energy consumption

There are several industrial applications for the event based sampling systems. One of them is to control of an internal combustion engine that are sampled against the engine speed. Another one is manufacturing system, sampling is based on the production rate. Relay systems with on-off control and satellite thrusters are event based too. Process industry also takes the advantage of event based sampling in statistical process control by not interfering in the system if there is not recently calculated control action [2].

Additionally, Event Based Control is the closest controller to the nature of a human, when human interferes to the system to control is could be considered as event-triggered control that is like when the output has changed enough from the set point control action takes place.

The general scheme for the event based control loop is shown in *Figure 1.1*, components are described below;

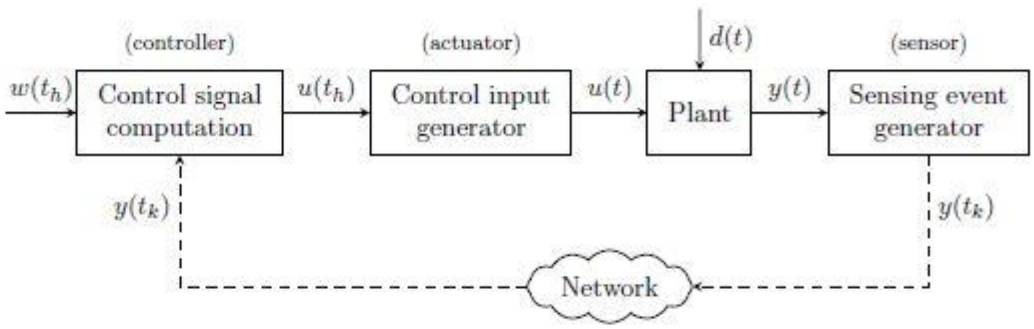

Figure 1.1: Event based control loop

- Control signal computation; the block that creates the input for the actuator that is indeed named as control signal, by computing the difference of the feedback signal and the set point signal as input

- Control input generator (Actuator)

- Plant describes continuous time dynamic system

- Sensing event generator is combination of event trigger mechanism and event generator that takes samples on a defined time and creates output regarding the last value and most recent value

- Network where transmission takes place and the data losses happens

Although for the embedded and networked control systems, the event based control is an efficient way to control regarding the flexible computation frequency and the reduced CPU usage, event-based control must decide both how to and when to actuate system. That’s why the time instants depend on an event generator or event function.

Despite all, why does the time-triggered control still dominate the literature instead of event triggered control? One of the most important reason is the difficulty of developing a strong theory with solid background for the event based system even if it has large range of implementation areas.

Table 1.1: Event based control theory with pros and cons

Event Based Control Theory	
Cons	*Pros*
The great difficulty involved with developing a system theory for event based control systems.	It closer in nature to the way a human behaves as a controller
	The reduction of the data exchange between sensors
	Extend the lifetime of battery-powered wireless sensors, to reduce the computational load in embedded devices, or to reduce the network bandwidth.
	Minimize the power consumption (and therefore to increase the battery life)
	Minimize the risk of lost data and stochastic time delays

Chapter 2

Triggering Mechanisms of Event Based Control

There are several triggering mechanisms in the literature. The most common Send on Delta method which can be called as *basic SoD* or *constant deadband*, uses a constant delta value and the reference data is the last sent one [31]. Except for *basic SoD* method, there are many techniques, such as *the relative deadband, the network-based deadband, the linear-predicted-data SoD, IAE-based SoD, Energy-error based SoD, Symmetric SoD* etc.

In this present study, it will be focused on methods of constant deadband, relative deadband, IAE based SOD and energy-error based SOD.

2.1. General Send-on-Delta Method (SoD)

One of the most wide-spread used triggering method is SoD method that has a constant delta value (δ), the reference signal is the last sent value and this strategy is named as basic SoD strategy. The model of the basic SoD strategy is shown below;

$$S(t) = \begin{cases} 1 \; if \; ||x_{ls} - x(t)|| \geq \delta \\ 0 \; if \; ||x_{ls} - x(t)|| \leq \delta \end{cases} \tag{2.1}$$

where $S(t)$ is send function, $x(t)$ is the data to be sent at instant t. In this expression the term $\left||x_{ls} - x(t)|\right| - \delta$ used as trigger function.

2.1.1. Send-on-a Delta with a Linear Prediction

Figure 2.1 shows the ordinary send on delta method. When the difference between the current value $x(t)$ and the last transmitted value $x(t')$ is greater than the threshold the value of the sensor is transmitted.

The combination of the send-on-delta concept and a linear predictor forms the new structure. A linear predictor computes the next sensor value $\hat{x}$ according to the past values if sensor transmitted the new value. One of the important point is that $\hat{x}$ is calculated both in the sensor and monitoring station. Moreover, the error in the monitoring station is always smaller than Δ leads the similar estimation performance with the classical send on delta, but the sent data number is commonly less than the classical send on delta.

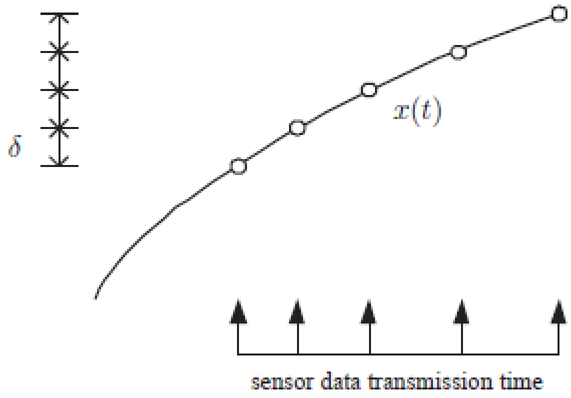

Figure 2.1: Usual send-on delta method

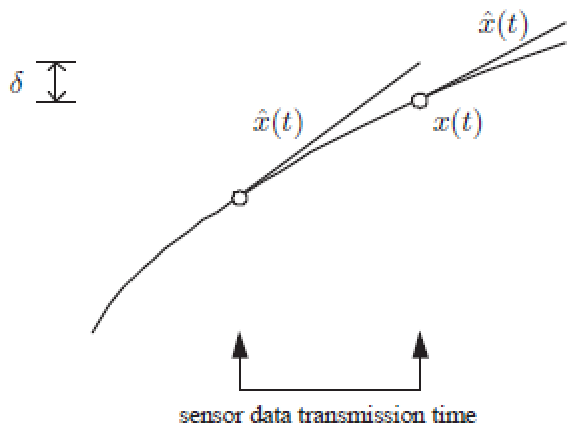

Figure 2.2: Send-on delta method with a linear prediction

2.1.2. New Send-on-Delta Algorithm with a Linear Predictor

The theory is given below in figures. In the sensor, according to the acceptance sampling period is T. A discrete-time signal x_k is defined by $x_k = x(kT)$. Admitting sampling cycle of the sensor is T, the real transmission rate is not T because all sampled sensor data are not transmitted.

In the sensor block, $\hat{x}_k = f(\hat{x}_{k-1}, \ldots., \hat{x}_{k-M-1})$ shows a linear predictor, M indicates length of the memory. If M = 1, $\hat{x}_k$ is calculated according to the $\hat{x}_{k-1}$ and $\hat{x}_{k-2}$.

When the difference between the current value x_k and the estimated value $\hat{x}_k$ is larger than the delta value, then send the all data between x_k and x_{k-M} instead of only x_k.In send on delta method it is only deemed to send x_k. Note that the transmitted data is more than the data is sent in classical send on delta.

During the monitoring phase, if there are no upcoming sensor values, the current sensor value is estimated with the help of a linear predictor, $\hat{x}_k = f(.)$. Transmitter algorithm ensures that the difference between x_k and $\hat{x}_k$ is smaller than the delta value [17].

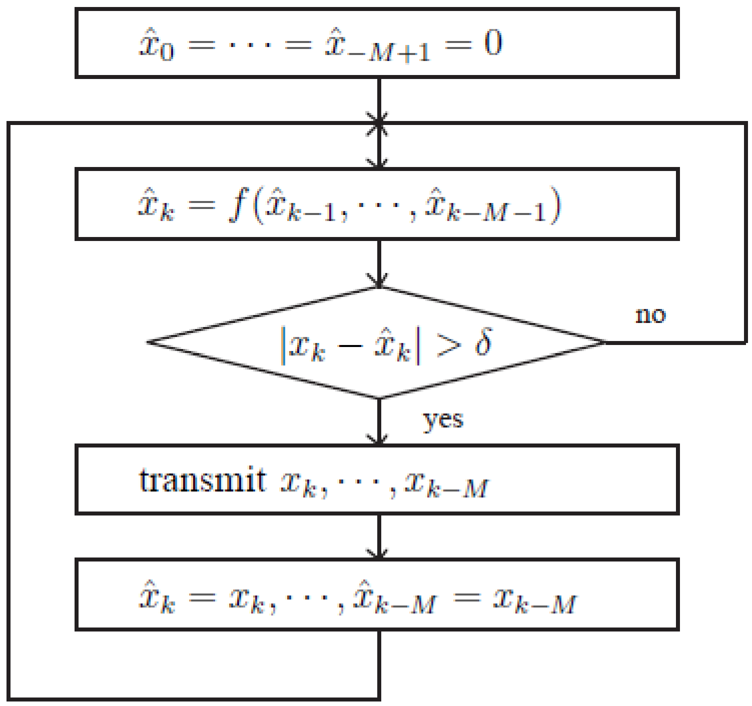

Figure 2.3: Transmitter algorithm in sensor nodes ($x_k, x(kT)$)

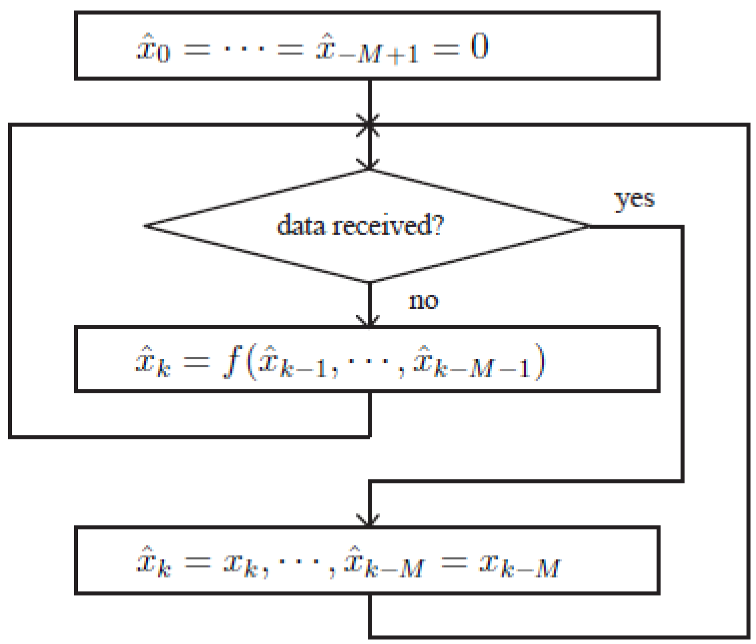

Figure 2.4: Receiver algorithm in the monitoring station

There are some more SoD structures other than basic SoD, one of them is ‘relative deadband’, which is motivated by Weber’s Law of Just Noticeable Differences which states that: ‘The Difference Threshold (or "Just Noticeable Difference") is the minimum amount by which stimulus intensity must be changed in order to produce a noticeable variation in sensory experience’ [35]. Based on this law, deadband value is employed as proportional to the most-recently sent data. The network-based deadband where the deadband value is chosen according to the status of the network, the linear-predicted-data SoD where the last sent data and the previous sampled data are used to calculate the linear prediction as reference data. Apart from those techniques other SoD structures are based on the trigger mechanism, for instance the IAE-based SoD, that checks if the absolute integral value the error is bigger than the deadband value, same works for the energy-error based SoD, that the square of the integral of the error is compared with the deadband value [4].

2.2. Triggering Mechanisms in The Literature

2.2.1. Constant Deadband

The aim is to decrease network traffic, deadband control is emerged for the transmission of the sampled signals. The main idea is to compare most recently sent value with the current value where they stated as $x(t')$ and $x(t)$ respectively. When the absolute value of difference between them exceeds the threshold value, also called the deadband value ‘Δ’, current value is transmitted and the new deadband is adjusted in the neighborhood of the value $x(t)$, otherwise there is no change of the output. The mathematical logic of constant deadband method is,

$$|x(t)| \in \begin{cases} 0, [|x(t')| + \Delta] & if\ |x(t')| < \Delta \\ [|x(t')| \pm \Delta] & if\ |x(t')| \geq \Delta \end{cases} \tag{2.2}$$

2.2.2. Relative Deadband

The relative deadband increase linearly by the multiples of the most recently value $x(t')$ with the proportional factor ε, the deadband value is defined by

$$\Delta_{x\,(t')} = \varepsilon * |\, x\,(t')|. \tag{2.3}$$

In the application there is a vulnerability that is if the signal $x(t')$ becomes infinitesimal the structure could become inapplicable. To prevent this deadband is defined as lower bounded $\Delta \geq \Delta_{\min}$.

$$|x\ (t)|\ \in \begin{cases} 0, [|x\ (t')|\ +\ \Delta_{min}]\ \ if\ |x\ (t')| < \Delta_{min} \\ \left[|x\ (t')|\ \pm\ \Delta_{x\ (t')}\right]\ \ if\ |x\ (t')| \geq \Delta_{min} \end{cases} \tag{2.4}$$

2.2.3. Event-Based Integral Sampling Criterion

This theorem states that, signal $x(t)$ is sampled, if the integral of the error exceeds a defined threshold,

$$\Delta = \int_{t_2}^{t_1} |x(t) - x(t_{i-1})| dt \tag{2.5}$$

i shows the number of samples. It is also considered the zero-order hold is used to store the most recently sent sample between sampling instants.

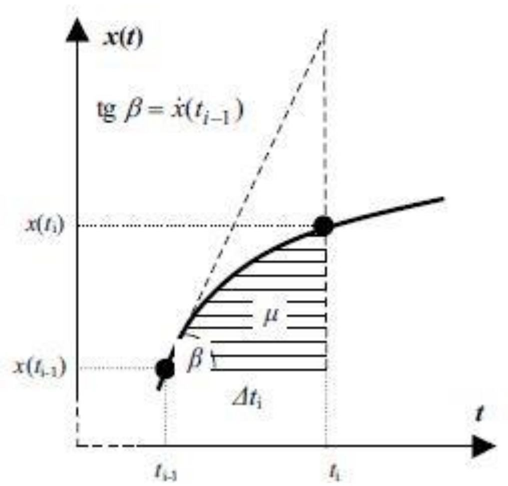

Figure 2.5: The calculation of a sampling interval in the uniform event-driven integral scheme

The integral sampling provides several advantages. The significant change is to take the integral of the absolute value of the error, but the conventional send on delta method cannot reveal the signal oscillations or steady-state error if they stay at the confidence interval. The signal tracking performance of the integrated error is better than the classical SoD system. To explain this better let us imagine the control system that the system output reaches the equilibrium, so the signal becomes almost constant. If the system is observed by the classical SoD, the sampling sometimes is not triggered for a long-time cycle, because the variation of the signal is not enough to initialize the trigger mechanism *(Figure 2.6)*.

The *Figure 2.7* below shows the behavior of the system with integral criterion trigger mechanism, which becomes more accurate with the integral sampling method. The squared error of the integral criterion has better performance on the measure of the signal tracking quality compared to the linear error of the conventional mechanisms. Besides, because of the noise, the output value of the data could go beyond the threshold. That situation could cause futile data transmissions [18].

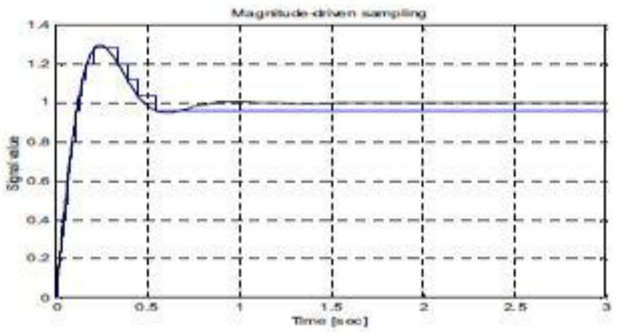

Figure 2.6: The steady state error in magnitude driven scheme

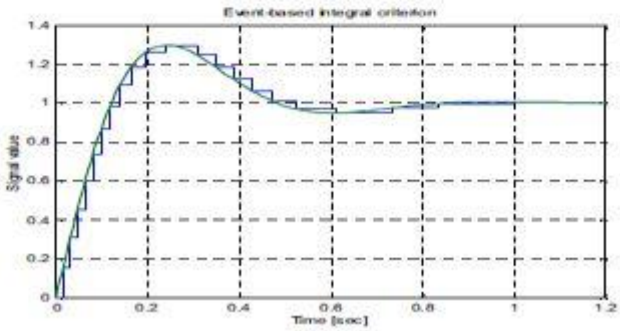

Figure 2.7: The example of sampling according to the event-based integral criterion

2.2.4. Send-on-Energy Criterion

Another version of send-on-area is the trigger mechanism based on energy of sampling error also called send-on-energy criterion.

If the energy of a difference of the current and the most recent signal, go beyond a certain limit then sample the signal x(t) regarding the energy criterion

$$\int_{t_{i-1}}^{t_i} [x(t) - x(t_{i-1})]^2 dt = \Delta \quad (2.6)$$

The main difference between the send-on-area and send-on-energy is that the error is squared before the integrating operation. This criterion has a significantly good performance on signal tracking when compared to the magnitude-driven triggering [18].

2.2.5. Symmetric Send on Delta

Basically, this method is an upgraded version of the Send on Delta sampling, that can be also generalized as a relay with hysteresis.

In symmetric-send-on-delta sampling $v(t)$ is considered as the input signal and $v^*(t)$ is the sampled output signal that is multiple of a certain threshold Δ multiplied by a gain $\beta > 0$, $v^*(t) = j\Delta\beta$. The output is sampled signal that changes its value if the input signal is more-less than the level limits. When the input is more than the upper level the sampled output changes its value to upper limit, otherwise if the lower limit is exceeded the output will be updated to the lower limit.

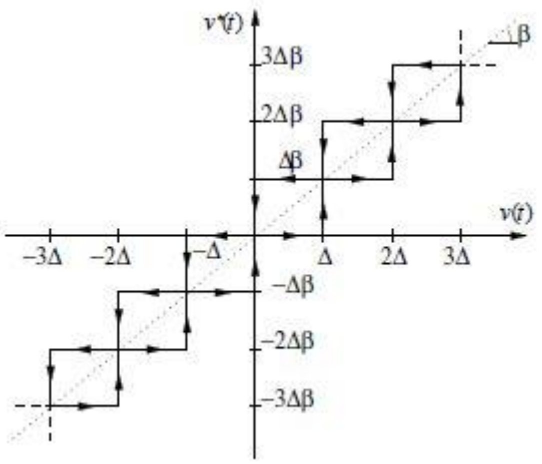

Figure 2.8: Relationship between $v(t)$ and $v^*(t)$

The behavior of the method can be modeled mathematically as:

$$v^*(t) = ssod(v(t); \Delta, \beta) = \begin{cases} (i+1)\Delta\beta & if\ v(t) \geq (i+1)\Delta\ and\ v^*(t^-) = i\Delta \\ i\Delta\beta\ if\ v(t) \in [(i-1)\Delta, (i+1)\Delta]\ and\ v^*(t^-) = i\Delta \\ (i-1)\Delta\beta & if\ v(t) \leq (i-1)\Delta\ and\ v^*(t^-) = i\Delta \end{cases} \quad (2.7)$$

The sampled signal does not depend on the idle conditions. The parameter Δ does not affect the stability so could be selected to manage to the limitation of the data transmission and decreasing of the steady state error [33].

Chapter 3

Labview Library

3.1. Labview Graphical System Design Platform

To meet the requirements of the users, many modeling, and simulation tools are available in the market. Labview programming tool has been chosen and used for the simulation part of this thesis.

Labview is software for the applications which require test, simulation, measurement and control. The Labview programming environment is compatible with other tools and software which provides to models of the complex systems [36].

3.2. Labview Library of The Event Based Control Structure

3.2.1. General Framework of The Event Based Control Structure

As shown in *Figure 3.2*, the entire system consists of a setpoint variable, a controller, a plant, a sensor and an event generator. The setpoint gives a reference signal to the system. The plant is the one of the benchmark processes will be taken into account in the next chapter. Event generator generates events. Sensor includes triggering mechanism inside. Output of the sensor is the value of output at every event.

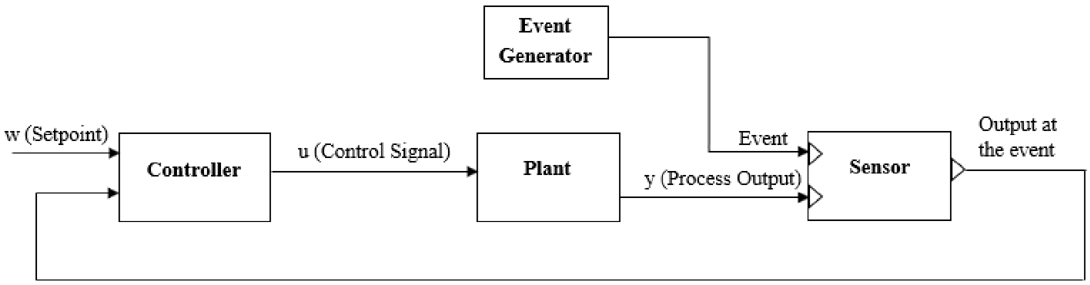

Figure 3.1: General framework of event based control

The idea is that the event generator which generates an event signal, gives the generated signal to the sensor as input. The output of the plant continuously generates controlled variable. The sensor related with triggering mechanism gives the current value of output at the event as output. The controller takes the difference of setpoint and this output and then, generates a control signal.

General framework of the event based control structure built in Labview is shown in *Figure 3.2.*

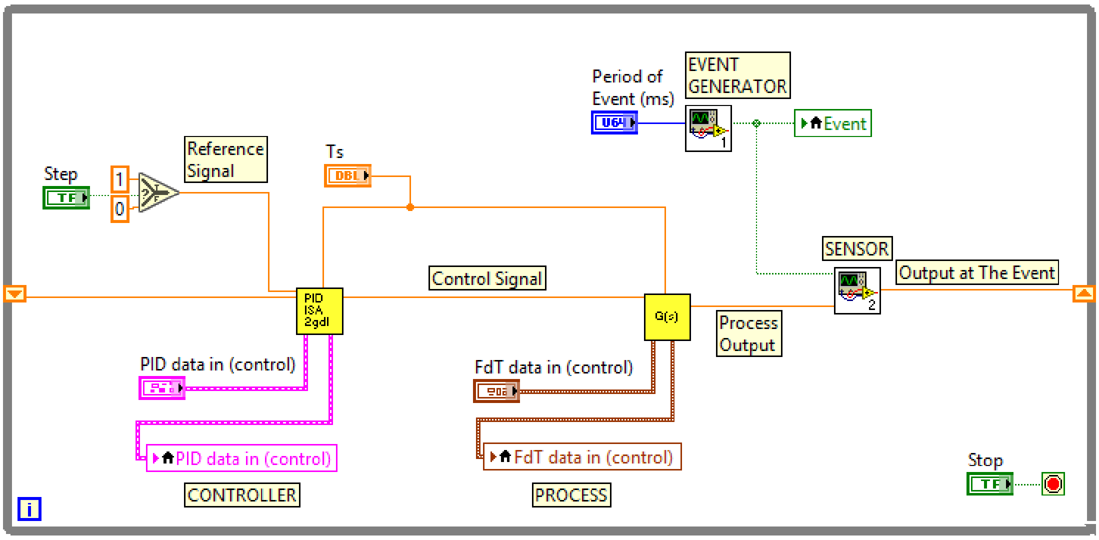

Figure 3.2: LABVIEW implementation of general framework

When the simulation is activated, firstly the system output is waited to set the zero, followed by giving a step value to the PI controller as a reference. Benchmark processes as mentioned in the next chapter forms the plant of the simulation. The output of the plant is the input of the sensor. Whenever there is an event, the sensor samples the input signal as an output based on its own structure which consists triggering mechanism and event detector.

Labview implementation of general framework includes several sub VI structures such as event generator, sensor and triggering mechanism. These sub VI structures will be described in the following.

3.2.2. Process Transfer Function

To realize benchmark processes in simulation, system transfer function VI has been used. Inputs and outputs of the transfer function VI is shown *Figure 3.3*. The control signal is connected to "Ingresso (u)" and the output of transfer function is "Uscita (y)".

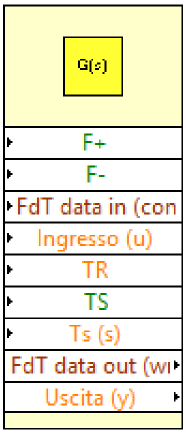

Figure 3.3: Process transfer function VI

The control panel of the transfer function is in *Figure 3.4*. System transfer function till third order can be controlled with presented VI.

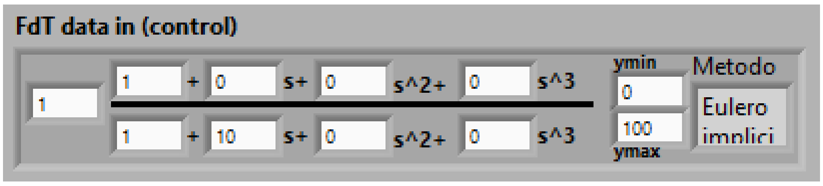

Figure 3.4: Process transfer function control panel

3.2.3. Delay Time Transfer Function

First order process with delay time transfer function model requires a delay time VI in simulation. Benchmark processes 4 and 5 to be introduced in Chapter 4, need delay time. The structure of delay time VI is shown in *Figure 3.5*.

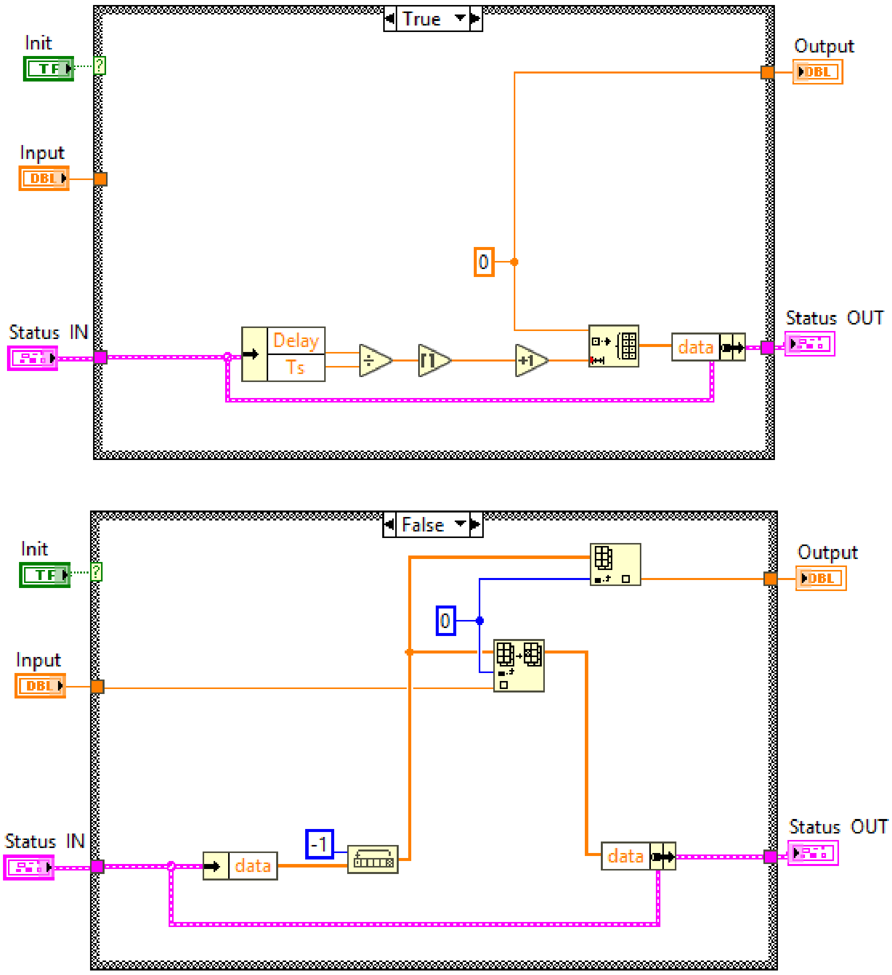

Figure 3.5: Delay time VI

3.2.4. Controller Transfer Function

To control the event based system, a PID controller transfer function VI has been used. Inputs and outputs of the VI and control panel are shown in *Figure 3.6.* The feedback node is connected to “PV” and control signal is output “CS”.

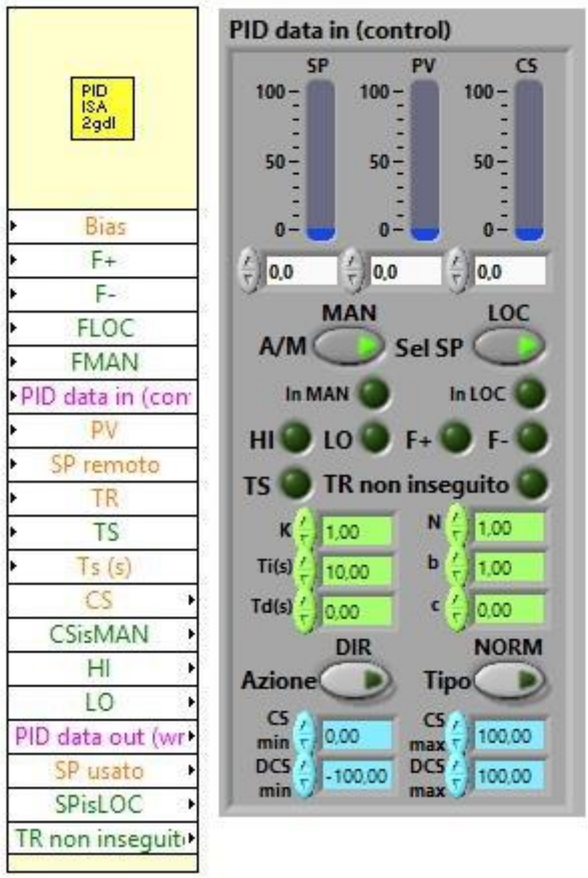

Figure 3.6: Controller VI and control panel

Reference signal can be given remotely with input “SP remote” and reference signal also can be changed manually in control panel. Process value and control signal value can be observed in control panel.

3.2.5. Periodic Event Generator

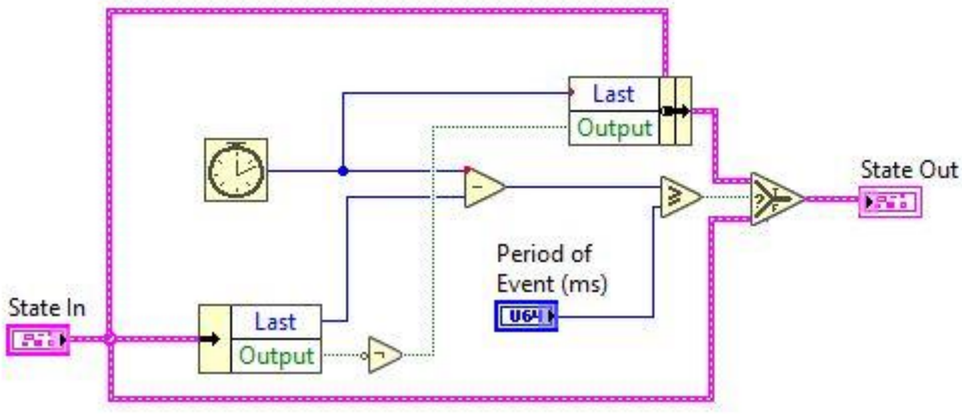

Figure 3.7: Periodic event generator implementation on LABVIEW

This structure continuously toggles the output that creates falling and rising edges synchronously. The elapsed time between two consecutive edges is defined by the period of the system.

3.2.6. Event Generator

The Event Generator VI consists of the sub VI of The Periodic Event Generator and edge detection mechanism which compares the current and the new values of the output of the periodic event generator.

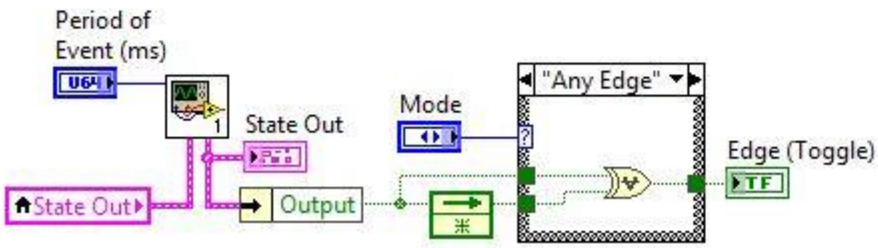

Figure 3.8: Event generator implementation on LABVIEW

3.2.7. Sensor

Input of the sensor runs continuously, so the new value is different from the old one. The idea of sensor is that when an event which means every toggling of event generator output occurs, the output of sensor is the sampled input.

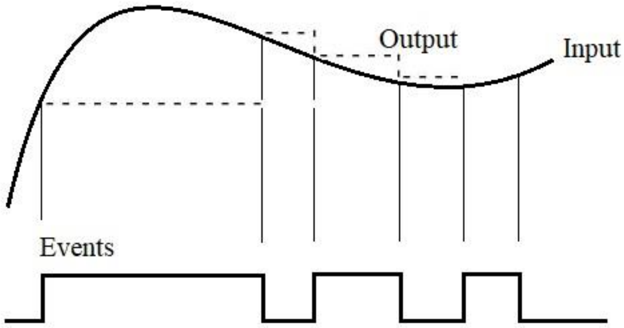

Figure 3.9: The idea of sensor

The sensor implementation contains the sub VI of the event triggering mechanism and the block for the update of the system output when there is an event. Basically, this structure creates system output at each event.

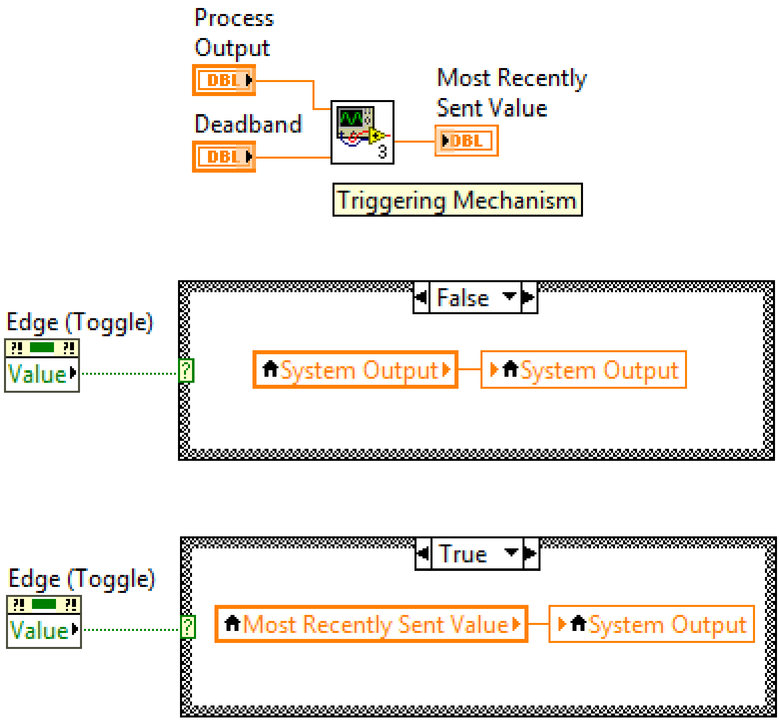

Figure 3.10: Sensor implementation on LABVIEW

3.3. Labview Library of Triggering Mechanisms

3.3.1. Constant Deadband

The mathematical expression of constant deadband triggering method is,

$$|x\,(t)| \in \begin{cases} 0, [|x\,(t')| + \Delta] & if\ |x\,(t')| < \Delta \\ [|x\,(t')| \pm \Delta] & if\ |x\,(t')| \geq \Delta \end{cases} \tag{2.8}$$

where $x(t)$ is the current value and $x(t')$ is the most recently sent value. As mentioned in *Chapter 2*, constant deadband method checks if the difference between the most recently sent value and the current value exceeds the threshold sample the input value.

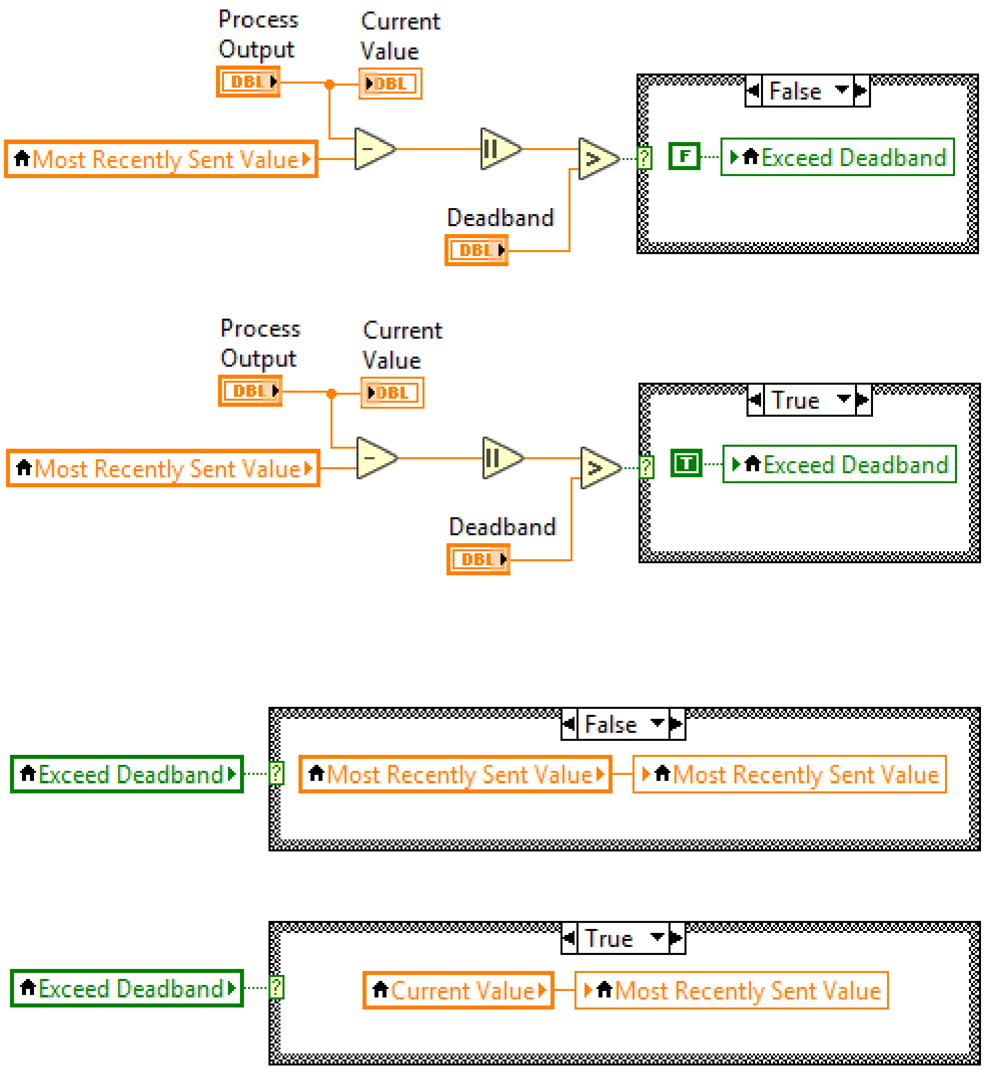

Figure 3.11: Constant deadband triggering mechanism implementation on LABVIEW

The process output which the output of the plant in the framework is the input signal for the deadband structure that is indeed considered as current value. At the upper part of the Software VI Block takes the difference of the current value and the most recently value, then the system compares the absolute value of the difference of the signals and the deadband value. If the result is true it activates the Boolean named as exceed deadband which indicates the deadband is exceed. The Boolean used as selection variable and allows us to update the output value regarding the result of the comparison between the current value and the most recent value. If the difference goes beyond the deadband most recently sent value is updated by current value otherwise the local variable of the most recently sent value keeps holding the same data inside.

3.3.2. Relative Deadband

As mentioned in *Chapter 2* – (2.3), The deadband value changes linearly according to the multiple of the proportional factor and the most recently sent value.

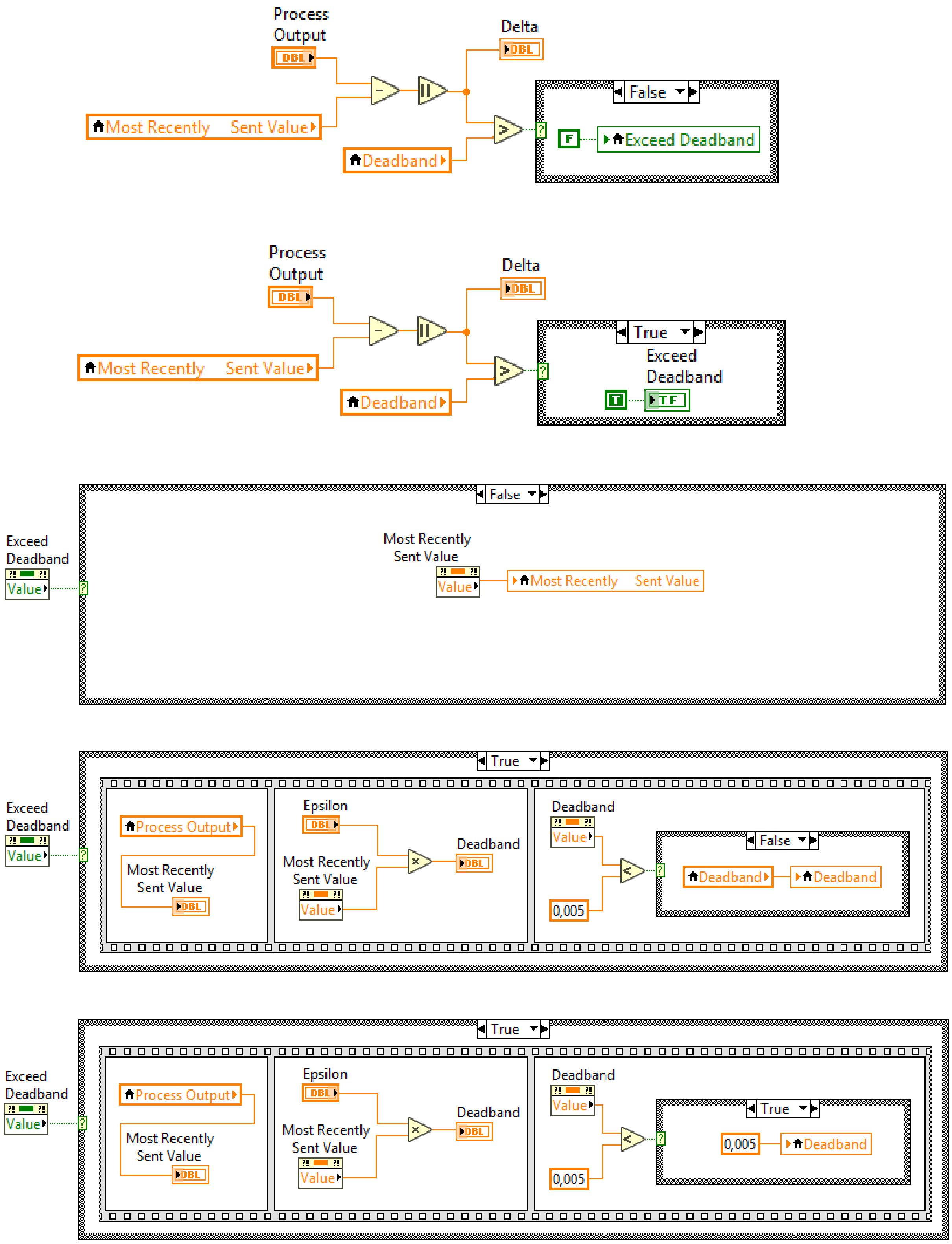

Figure 3.12: Relative deadband triggering mechanism implementation on LABVIEW

The very similar logic of the constant deadband application is integrated on the relative deadband mechanism. If the difference of the new value and most recently sent value exceeds the deadband, this condition activates the case structure below and consequent actions inside the structure are realized. First action is to update the most recently sent value with the process output, then after deadband value is computed again based on the multiplication of epsilon and the updated most recently sent value, followed by the analysis of the deadband value. In practical applications to prevent the deadband value becomes close to the origin, the deadband value predefined as lower bounded so last part of the block provides to compare the deadband with the lower bound value, if it becomes less than the limit changes the value with the lower limit.

3.3.3. Integral SoD

According to the integral criterion, current value of process output is sampled, if the integral of the error exceeds a defined threshold. Definition of integral of the error is

$$\int_{t_2}^{t_1} |y(t) - y(t_{i-1})| dt \tag{2.9}$$

where $y(t)$ is the current value of process output and $y(t_{i-1})$ is most recently sent value.

The algorithm of the Integral SoD is

$$\Delta = y - y_{last}$$

$$I = I + |\Delta| * T_s$$

$$\begin{cases} y_{last} = y, I = 0 & if\ I > threshold \\ y = y_{last} & if\ I < threshold \end{cases} \tag{2.10}$$

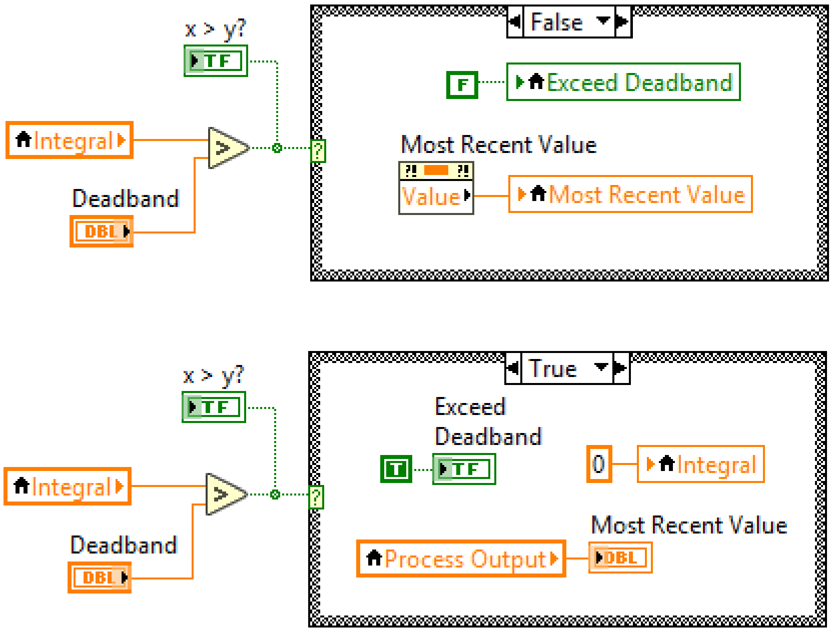

Figure 3.13: Integral SoD triggering mechanism implementation on LABVIEW (1)

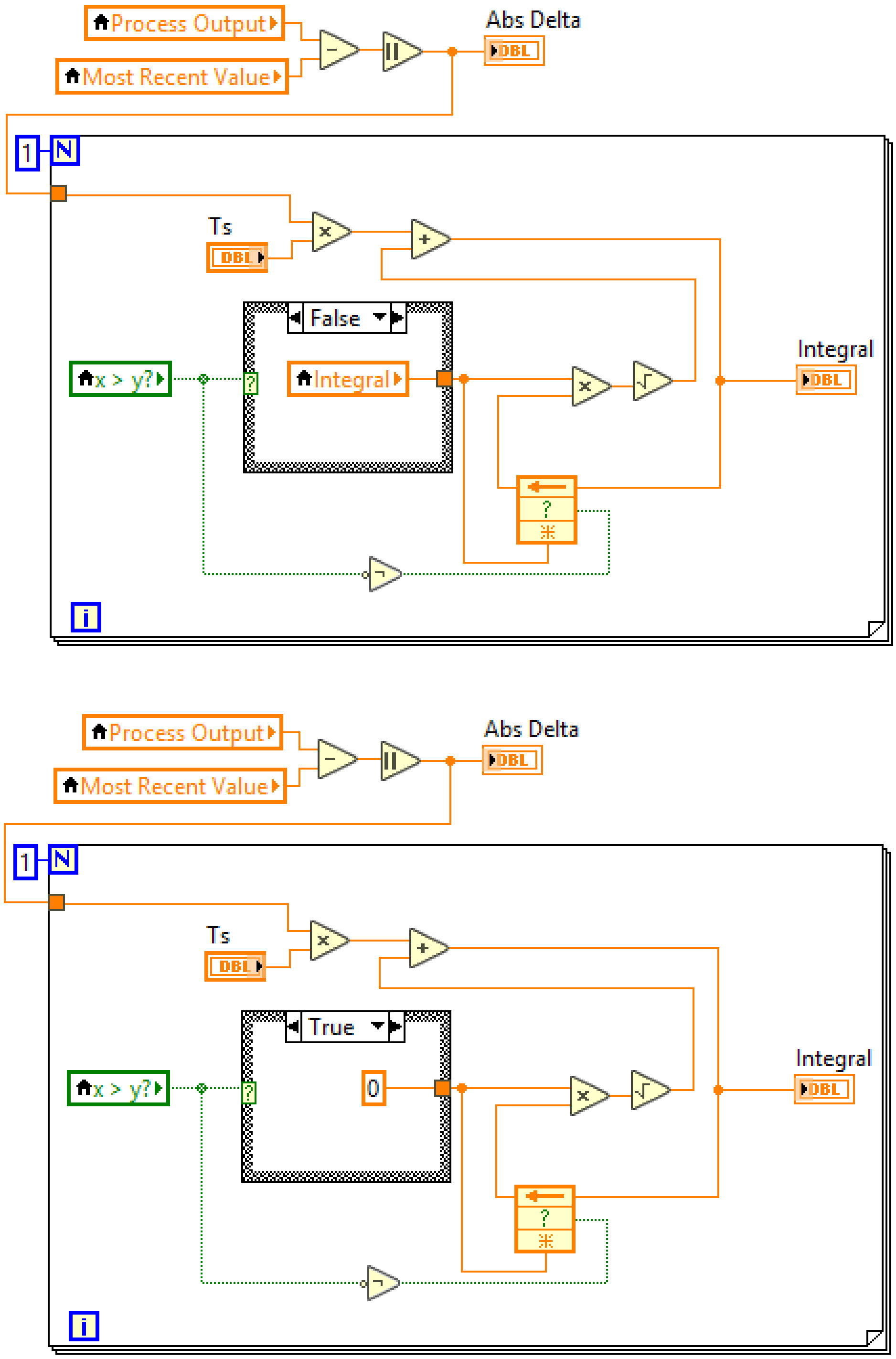

Figure 3.14: Integral SoD triggering mechanism implementation on LABVIEW (2)

The for loop in the structure computes the integral value based on the mathematical expression in the theory above. It takes the difference of the most recent and new input signals, multiples it with sampling time (T_s), then adds this value to the previous integral value. Every cycle of the program it checks whether the integral value reaches the deadband value or not. Regarding the comparison result the same update actions hold place as the previous deadband structures. Additionally, if the difference exceeds the deadband local variable of the integral is set to zero.

3.3.4. Energy SoD

The library has almost the same working logic except the square of the absolute value of the error before the integrating operation. The integral of the squared error is,

$$\int_{t_{i-1}}^{t_i} [y(t) - y(t_{i-1})]^2 dt \qquad (2.11)$$

where $y(t)$ is the current value of process output and $y(t_{i-1})$ is the most recently sent value.

The expression of Energy SoD is,

$$\Delta = y - y_{last}$$

$$I = I + \Delta^2 * T_s$$

$$\begin{cases} y_{last} = y \, , I = 0 & if \ I > threshold \\ y = y_{last} & if \ I < threshold \end{cases} \qquad (2.12)$$

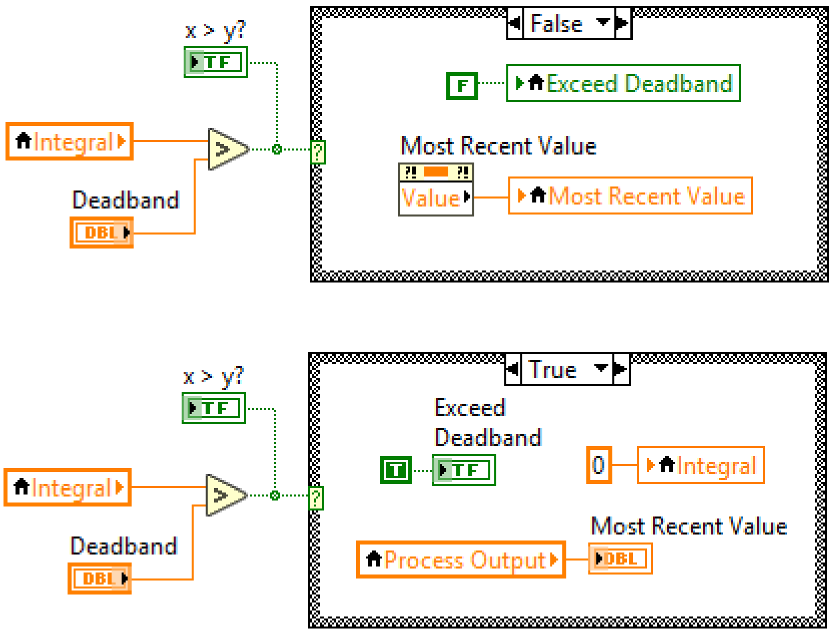

Figure 3.15: Energy SoD triggering mechanism implementation on LABVIEW (1)

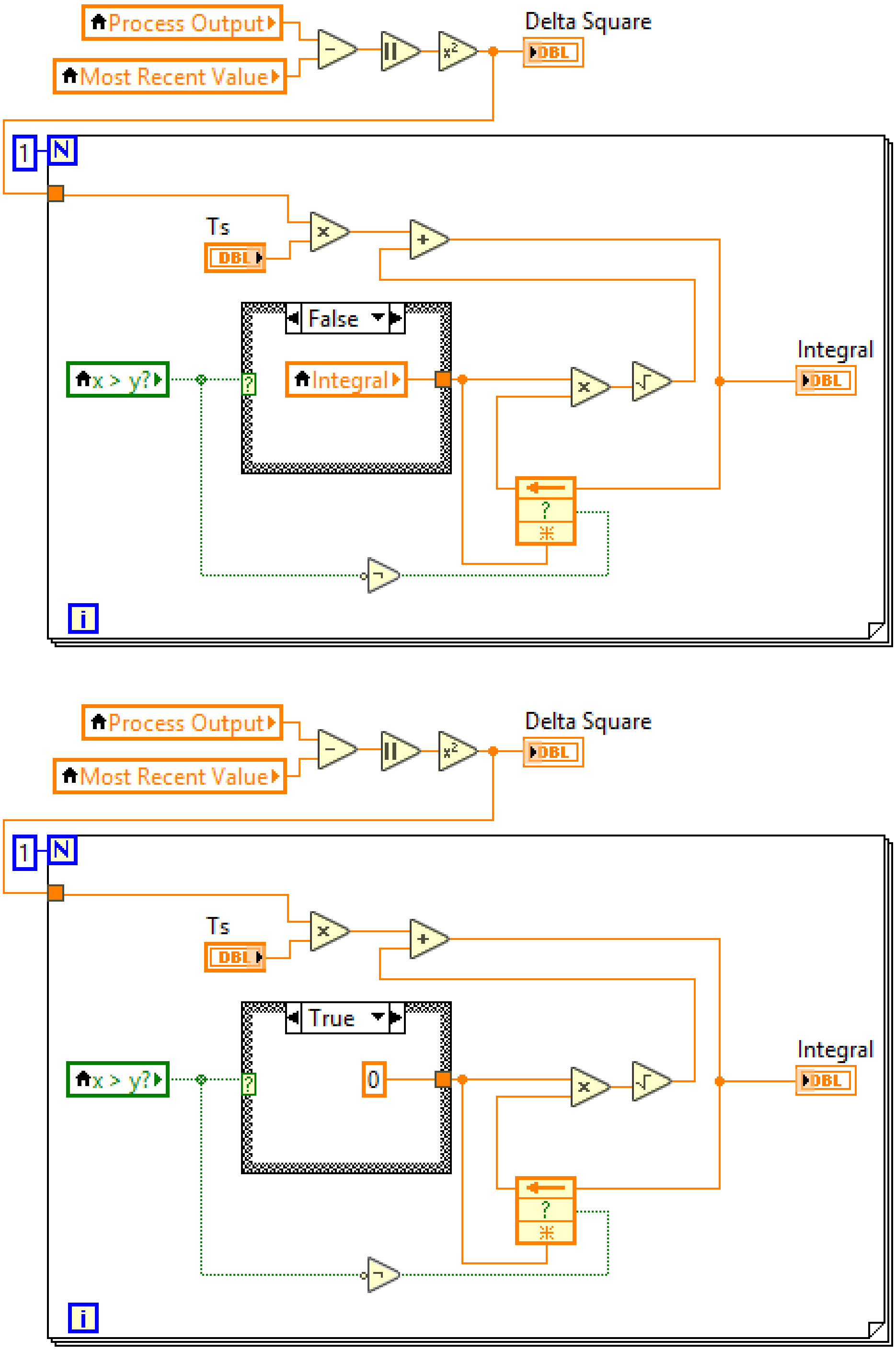

Figure 3.16: Energy SoD triggering mechanism implementation on LABVIEW (2)

Chapter 4

Benchmark Processes

In this chapter, it will be introduced benchmark processes were considered in the thesis. To test controllers and event based control structures, benchmark processes 1-5 of K. J. Åström and T. Hägglund have been taken account. The systems 1-5 are standard systems that are well suited to parametric studies in process control applications. Their properties can easily be changed by varying a parameter [22]. However, the other benchmark processes 6-10 are not so relevant for the typical process control applications.

The benchmark systems which are regarded in the thesis are below.

4.1. Benchmark Processes of Åström and Hägglund

4.1.1. System with Multiple Equal Poles

Transfer function

$$G(s) = \frac{1}{(s+1)^n} \qquad n = 1, 2, 3, 4, 8 \tag{3.1}$$

These systems are quite prevalent. The system behaves as system with long deadtime for the large values of n. Controller producers have used the system for many years.

4.1.2. Fourth Order System

Transfer function

$$G(s) = \frac{1}{(s+1)\,(1+\alpha s)\,(1+\alpha^2 s)\,(1+\alpha^3 s)} \qquad \alpha = 0.1, 0.2, 0.5, 1.(\tag{3.2}$$

The system has four poles which are locating dependent on a parameter α. This system for $\alpha = 1.0$ is equal to *System 4.1.1* - (4.1) for $n = 4$.

4.1.3. System with Right Half Plane Zero

Transfer function

$$G(s) = \frac{1 - \alpha s}{(s+1)^3} \qquad \alpha = 0.1, 0.2, 0.5, 1, 2, 3 \tag{3.3}$$

The system has 3 equal poles in -1 and one right half plane zero. Location of right half zero depends on parameter α. The control of the system is more difficult with increasing α value. In [23], the last α value is 5 instead of 3. In this thesis, it is considered as 3 since, a high zero time constant makes many rules fail. Moreover, it is improbable to be encountered in practice.

4.1.4. First Order System with Dead Time

Transfer function

$$G(s) = \frac{1}{1 + s\tau} e^{-s} \qquad \tau = 0.1, 0.2, 0.5, 2, 5, 10 \tag{3.4}$$

This structure is classic first order system with dead time. The magnitude of the delay is equal to 1. A drawback with the model is that it has slow roll-off at high frequencies [22].

In [22], there is also parameter $\tau = 0$. The system with $\tau = 0$ is totally nonphysical in some sense. We cannot apply the same method – *method of areas (it will be introduced in the current chapter)* – that we used for all rest of the benchmark. Nevertheless, we can approximate that case with an arbitrary precision by taking an arbitrarily small but non-zero value of τ. However, the test has no relevance from practical point of ours. Therefore, $\tau = 0$ value has been omitted with these justifications in the study.

4.1.5. Second Order System with Dead Time

Transfer function

$$G(s) = \frac{1}{(1 + s\tau)^2} e^{-s} \qquad \tau = 0.1, 0.2, 0.5, 2, 5, 10 \tag{3.5}$$

The model is nearly the same with (4.4), but it has more high frequency roll-off. In [23], there is also parameter $\tau = 0$. However, that particular case has not been included in the thesis due to reasons above.

4.2. Benchmark Processes to FOPDT

In this section, benchmark processes 1 to 5 of K. J. Åström and T. Hägglund have been obtained as first order models with delay time by using method of areas since, it is difficult to design controllers without systematic approach. To obtain FOPDT, at first, it has been generated step responses of benchmark processes. Then, it has been computed the areas A_0 and A_1. By using computed areas, first order models with delay time of benchmark processes have been acquired.

FOPDT (First order plus dead time) process model,

$$P(s) = \frac{\mu}{1+sT}e^{-sD}, \quad T > 0, D \geq 0. \tag{3.6}$$

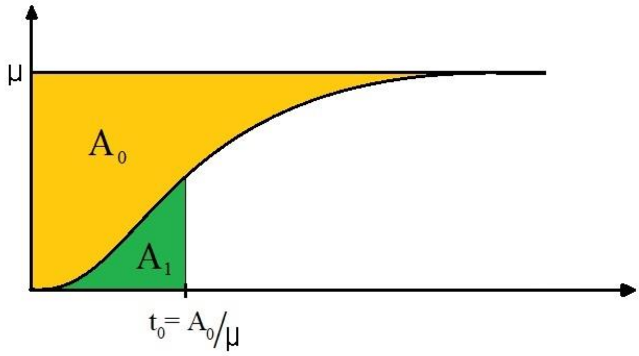

Figure 4.1: Areas A_0 and A_1

All benchmark processes have been transformed to FOPDT process model described in (4.6) by means of the method of areas. Accordingly, the unit step response $y_{us}(t)$ of each process has been obtained, then areas A_0 and A_1 have been computed,

$$A_0 := \int_0^{\infty} (y_{us}(\infty) - y_{us}(t))\, dt, \quad A_1 := \int_0^{A_0/y_{us}(\infty)} y_{us}(t) dt. \tag{3.7}$$

Eventually, the model gain, time constant and time delay have been determined respectively,

$$\mu = y_{us}(\infty), \quad T = \frac{eA_1}{\mu}, \quad D = \frac{A_0}{\mu} - T. \tag{3.8}$$

Hereunder, analytical application of the method of areas for each process are presented below.

The unit step responses of the five benchmark processes are

Process P_1:

$$y_{us,P_1}(t) = 1 - e^{-t}\sum_{k=0}^{n-1}\frac{t^k}{k!} \tag{3.9}$$

Process P_2:

$$y_{us,P_2}(t) = 1 - \frac{\alpha e^{-t/\alpha} - \alpha^3 e^{-t/\alpha^2}}{(\alpha+1)(\alpha-1)^3} + \frac{e^{-t} - \alpha^6 e^{-t/\alpha^3}}{(\alpha+1)(\alpha-1)^3(\alpha^2+\alpha+1)} \tag{3.10}$$

Process P_3:

$$y_{us,P_3}(t) = 1 - e^{-t}\left(1 + t + \frac{t^2}{2}(1+\alpha)\right) \tag{3.11}$$

Process P_4:

$$y_{us,P_4}(t) = \begin{cases} 0, & t < 1 \\ 1 - e^{-(t-1)/\tau}, & t \geq 1 \end{cases} \tag{3.12}$$

Process P_5:

$$y_{us,P_5}(t) = \begin{cases} 0, & t < 1 \\ 1 - e^{-(t-1)/\tau}\left(1 + \frac{t-1}{\tau}\right), & t \geq 1 \end{cases} \tag{3.13}$$

This allows us to express the areas A_0 and A_1 as follows.

Process P_1:

$$A_{0,P_1} = n, \quad A_{1,P_1} = \frac{e^{-n} n^n}{(n-1)!} \tag{3.14}$$

Process P_2:

$$A_{0,P_2} = (\alpha+1)(\alpha^2+1),$$

$$A_{1,P_2} = (\alpha+1)(\alpha^2+1) - \frac{\beta_1(2)\alpha^5 - \beta_1(1)\alpha^2}{\beta_2} + \frac{\beta_1(3)\alpha^9 - \beta_1(0)}{\beta_2(\alpha^2+\alpha+1)} \tag{3.15}$$

where

$$\beta_1(k) \coloneqq e^{-\frac{(\alpha+1)(\alpha^2+1)}{\alpha^k}} - 1,$$

$$\beta_2 \coloneqq (\alpha-1)^3\,(\alpha+1) \tag{3.16}$$

Process P_3:

$$A_{0,P_3} = \alpha + 3, \quad A_{1,P_3} = \frac{(\alpha+3)^3}{2} e^{-(\alpha+3)} \tag{3.17}$$

Process P_4:

$$A_{0,P_4} = \tau + 1, \quad A_{1,P_4} = e^{-1}\tau \tag{3.18}$$

Process P_5:

$$A_{0,P_5} = 2\tau + 1, \quad A_{1,P_5} = 4e^{-2}\tau \tag{3.19}$$

The derived FOPDT model parametrization for the five classes is indicated below.

Process P_1:

$$\mu_{P_1} = 1, \quad T_{P_1} = \frac{e^{1-n}n^n}{(n-1)!}, \quad D_{P_1} = n - \frac{e^{1-n}n^n}{(n-1)!} \tag{3.20}$$

Process P_2:

$$\mu_{P_2} = 1,$$

$$T_{P_2} = e\left((\alpha+1)(\alpha^2+1) - \frac{\beta_1(2)\alpha^5 - \beta_1(1)\alpha^2}{\beta_2} + \frac{\beta_1(3)\alpha^9 - \beta_1(0)}{\beta_2(\alpha^2+\alpha+1)}\right),$$

$$\begin{aligned} D_{P_2} = {} & (1-e)(\alpha+1)(\alpha^2+1) \\ & + e\left(\frac{\beta_1(2)\alpha^5 - \beta_1(1)\alpha^2}{\beta_2} - \frac{\beta_1(3)\alpha^9 - \beta_1(0)}{\beta_2(\alpha^2+\alpha+1)}\right) \end{aligned} \tag{3.21}$$

Process P_3:

$$\mu_{P_3} = 1,$$

$$T_{P_3} = \frac{1}{2}e^{-\alpha-2}(\alpha+3)^3,$$

$$D_{P_3} = \alpha + 3 - \frac{1}{2}e^{-\alpha-2}(\alpha+3)^3 \tag{3.22}$$

Process P_4:

$$\mu_{P_4} = 1, \quad T_{P_4} = \tau, \quad D_{P_4} = 1 \tag{3.23}$$

Process P_5:

$$\mu_{P_5} = 1, \quad T_{P_5} = 4e^{-1}\tau, \quad D_{P_5} = 1 + 2\tau(1 - 2e^{-1}) \tag{3.24}$$

Lastly, based on (4.6), final form of the processes list below,

Table 4.1: Parameters of FOPDT models

PROCESS CLASS	Parameter	μ	T	D
1 System with Multiple Equal Poles	n = 1	1	1	0
	n = 2	1	1,4715	0,5285
	n = 3	1	1,8270	1,1730
	n = 4	1	2,1242	1,8758
	n = 8	1	3,0355	4,9645
2 Fourth Order System	alfa = 0,1	1	1,0054	0,1056
	alfa = 0,2	1	1,0239	0,2241
	alfa = 0,5	1	1,1855	0,6895
	alfa = 1	1	2,1242	1,8758
3 System with Right Half Plane Zero	alfa = 0,1	1	1,8240	1,2760
	alfa = 0,2	1	1,8154	1,3846
	alfa = 0,5	1	1,7597	1,7403
	alfa = 1	1	1,5932	2,4068
	alfa = 2	1	1,1447	3,8553
	alfa = 3	1	0,7277	5,2723
4 First Order System With Deadtime	tau = 0,1	1	0,1	1
	tau = 0,2	1	0,2	1
	tau = 0,5	1	0,5	1
	tau = 2	1	2	1
	tau = 5	1	5	1
	tau = 10	1	10	1
5 Second Order System With Deadtime	tau = 0,1	1	0,1472	1,0528
	tau = 0,2	1	0,2943	1,1057
	tau = 0,5	1	0,7358	1,2642
	tau = 2	1	2,9430	2,0570
	tau = 5	1	7,3576	3,6424
	tau = 10	1	14,7152	6,2848

Chapter 5

Control Problems

There are various control problems in the literature. The main target of a feedback controller is commonly either disturbance rejection or setpoint tracking. Another objective of closed loop control system is fast as open loop or faster than open loop. All problems encountered in control systems have some advantages and disadvantages. In this thesis, it has been considered as combinations of control problems: setpoint tracking and closed loop system fast as open loop, setpoint tracking and closed loop system faster than open loop, disturbance rejection and closed loop system fast as open loop, disturbance rejection and closed loop system faster than open loop.

5.1. Setpoint Tracking vs Disturbance Rejection

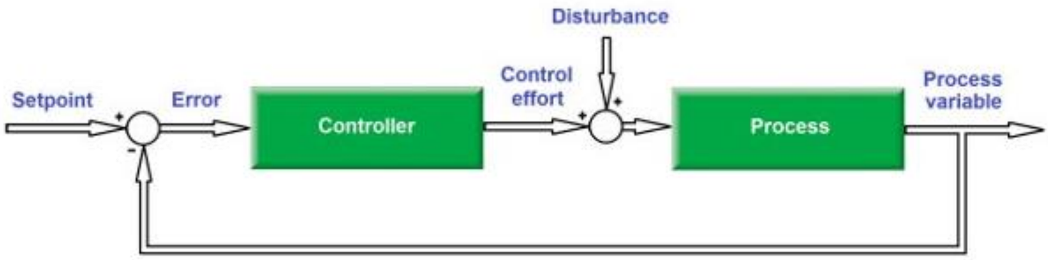

Figure 5.1: Control loop with feedback

A controller designed to reject the disturbances acts to bring the process variable back to the desired setpoint when a breakdown or load in the process causes any deviation. When the setpoint value changes often and the controller needs to increase or decrease the process variable correspondingly, setpoint tracking controller is suitable choice. It is frequently not possible to acquire good setpoint tracking and fast disturbance rejection at the same time.

In industrial process control applications, it is required that a good load-disturbance rejection since, the setpoint usually remain constant. However, in servo control, depending process operation conditions, set-point might ultimately be changed, then it is required that a good transient response to this change [27].

In analytical calculated controller tuning methods, it is included that a design parameter related with the control closed loop control system speed of response. It affects the system performance. In control systems, it is considered that load disturbance attenuation, robustness of the closed loop system and set point response by many controller manufacturers. Disturbance rejection is a primary concern in process control, while set point tracking is a main concern in motion control [30].

5.1.1. Open-Loop Operations

If it is considered that feedback node is removed which means that controller is operating in open loop mode. After a disturbance, process variable changes related with amplitude of the load and characteristics of process.

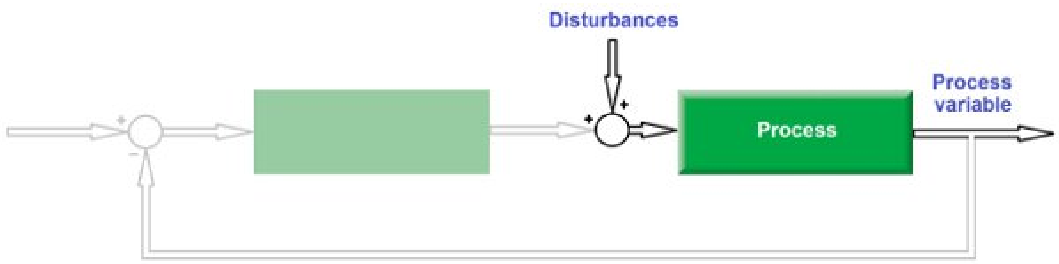

Figure 5.2: Open-loop operation

An open loop controller does not influence in determining how the process reacts to a disturbance, so that controller tuning is unreasonable when feedback is disabled. Vice versa, a set point change pass through both controller and process, without any feedback.

Consequently, to give a process response to a setpoint variation slower than a response to a sudden disturbance, the mathematical inertia of the controller merges with the physical inertia of the process. It is particularly true when the controller has integral action. The integral part of the PI controller filters the effects of a setpoint change by presenting a time lag.

5.1.2. Closed-Loop Operations

The mathematical inertia of the controller can be reduced without vanishing its ability to remove errors between the process variable and setpoint.

If it is desired a fast setpoint tracking controller, it necessitates an aggressive tuning. However, it should not be a problem to reject disturbances. On the other hand, if an abrupt load disturbance effects into the system, designed fast setpoint tracking controller shows an aggressive response, so an oscillated process output variable is determined unnecessarily.

On the contrary side, if a controller is tuned to reject disturbances, the controller mostly will be much slower to execute a setpoint change. As stated previously, industrial applications are managed at a constant setpoint for lengthy periods. Therefore, the only time that a controller designed to reject disturbance in the industry is exposed to a delay due to setpoint change is start-up.

5.2. CL as Fast as Open Loop vs CL Faster than Open Loop

The other control problem is desiring closed loop as fast as open loop vs closed loop faster than open loop. Tight or smooth control is related with the closed loop time constant. In tuning controller for fast response, it requires to have good robustness. Conversely, in tuning controller for slow response, it is critical point to have acceptable disturbance rejection.

To measure control performance, there are several performance indexes are used in academic papers and simulation studies such as IAE (Integral Absolute Error), ISE (Integral Squared Error), ITAE (Integral Time-weighted Absolute Error) etc. In this thesis, to measure control performance IAE and ISE have been used. Control performance can be qualified by the Integral Absolute Error (IAE) and the Integral Squared Error (ISE),

$$IAE = \int_0^\infty |e(t)|dt \tag{4.1}$$

$$ISE = \int_0^\infty e^2(t)dt \tag{4.2}$$

where $e(t)$ is the control error between the process variable and setpoint value.

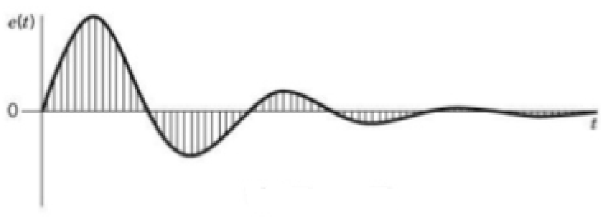

Figure 5.3: Disturbance input

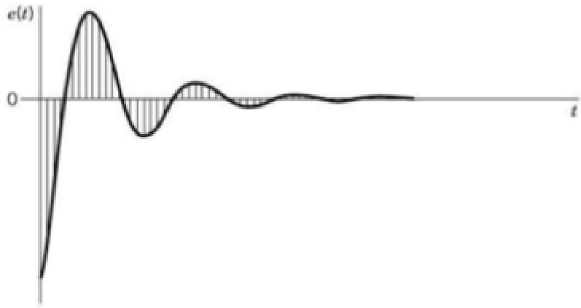

Figure 5.4: Setpoint change

ISE integrates the square of the error in term of time. ISE, penalizes large errors more than smaller ones, since the square of a large error is much bigger. Control systems modified to minimize ISE, tends to remove large errors instantly. However, it tolerates small errors persisting for an extended period. This results in fast responses with significant, low amplitude, oscillation.

IAE integrates the absolute error in term of time. In a system's response, it does not add weight to any of the errors. It tends to produce slower response than ISE optimal systems, but mostly with less sustained oscillation.

A controller tuning rule that allows to decide ratio between the speed of closed loop system response and the speed of open loop system response is investigated in the next section.

Chapter 6

Controller Tuning

Tuning process provides a convenient way for the system to determine the consequences of adjusting different controller parameters. To control the processes, PI controller has been chosen as controller type. Internal Model Control (IMC) method by Skogestad [29] has been adopted to derive PI controller parameters. In this section, it has been also chosen a tuning rule that allows to decide the ratio between closed loop response speed and process dynamics time scale and then, PI controller parameters has been derived.

Since the proportional-integral (PI) controller has two parameters, it is not easy to find good values without a systematic process. The reasons of selection IMC-PI(D) method are listed below,

- It is simple and easy to memorize.
- It is FOPDT model-based and analytically derived. First order with delay time processes were already obtained in *Chapter 4.*
- It works well on a wide range of processes.

It has been used the model parameters (μ, D, T) to tune the PI controllers.

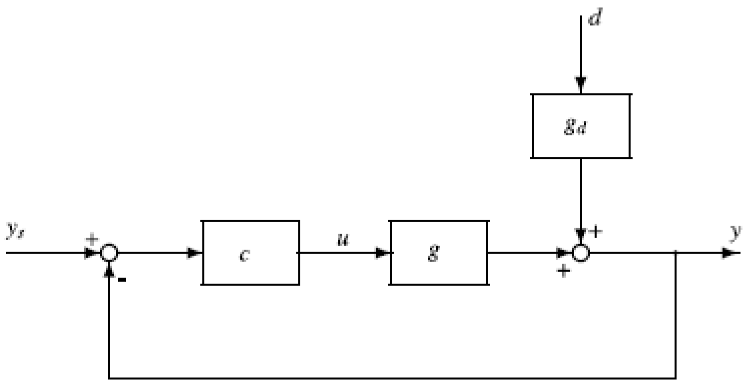

Figure 6.1: Block diagram of feedback control system

FOPDT process structure is

$$G(s) = \frac{\mu}{1 + Ts} e^{-sD} \tag{5.1}$$

where μ is gain of the system, T is time constant of the system and D is time delay.

PI controller structure is

$$C(s) = K_c \left(1 + \frac{1}{T_i\, s}\right) \tag{5.2}$$

where K_c is the controller gain and T_i is the integral time.

For the system in *Figure 6.1,* the closed loop set point response is

$$\frac{y}{y_s} = \frac{C(s)\, G(s)}{1 + C(s)\, G(s)} \tag{5.3}$$

where it is assumed that the measurement of the output y is perfect. The aim of direct synthesis is to derive the desired closed loop response and solve for the corresponding controller. From (6.3)

$$C(s) = \frac{1}{G(s)} \frac{1}{\frac{1}{(y/ys)_{desired}} - 1}. \tag{5.4}$$

It is considered the first order time delay model $G(s)$ in (6.1) and it is specified a desired smooth first order response with time constant and time delay,

$$\left(\frac{y}{y_s}\right)_{desired} = \frac{1}{1 + \lambda s}\, e^{-sD} \tag{5.5}$$

where λ is closed loop desired time constant and D is time delay. D (Time delay) is kept in the desired response due to unavoidability. Then, it is substituted (6.1) and (6.5) into (6.4), it gives a “Smith Predictor” controller

$$C(s) = \frac{1 + Ts}{\mu} \frac{1}{(\lambda s + 1 - e^{-sD})} \tag{5.6}$$

λ is the desired time constant and it is the only tuning parameter for the PI controller. To obtain PI settings, it is introduced in (6.6) a first order Taylor series approximation of delay,

$$e^{-sD} \cong 1 + Ds. \tag{5.7}$$

The new form of the controller structure is

$$C(s) = \frac{1 + Ts}{\mu} \frac{1}{(\lambda + D)s}. \tag{5.8}$$

Equalizing (6.2) and (6.8), it is obtained that

$$K_c = \frac{1}{\mu} \frac{T}{(\lambda + D)}, \qquad T_i = T \tag{5.9}$$

According to Skogestad [29], these settings are derived by considering the setpoint response. However, for lag dominant process with $T \gg D$, the choice of $T_i = T$ gives long settling time for load disturbances. To get better load disturbance response performance, it may reduce that the integral time, but not too much, because otherwise response will give slow oscillations and robustness performance. Hereunder, Skogestad [37] offers that a good trade-off between disturbance response and robustness by choosing integral time,

$$T_i = 4(\lambda + D). \tag{5.10}$$

However, in this study, it has been considered as $T_i = T$ to find controller parameters.

Additionally, another control problem which is "Closed loop fast as open loop versus closed loop faster than open loop" has been applied easily into this method.

Open loop settling time is

$$t_{settling-openloop} = D + 5T \tag{5.11}$$

to settle into band %1.

Closed loop desired settling time is

$$t_{settling-desired} = 5\lambda \tag{5.12}$$

to settle into band %1.

To compute closed loop response is "a" times faster than open loop, by using (6.11) and (6.12), we get

$$\lambda = \frac{1}{5a}\ (D + 5T) \tag{5.13}$$

Closed loop response is as fast as open loop response for parameter $a = 1$ and closed loop response is faster than open loop response for parameter $a = 4$.

By using $a = 1$, it is obtained that good setpoint tracking and poor load disturbance performance. Conversely, by using $a = 4$, it is observed that fast remove in load disturbance and oscillated setpoint tracking.

To sum up, by using (6.9) and (6.13), two PI controllers for each process have been derived: one for closed loop as fast as open loop, the other for closed loop faster than open loop.

PI controllers with determined parameters K_c and T_i are listed below.

Table 6.1: Parameters of PI controllers

PROCESS CLASS	Parameter	a=1		a=4	
		K	Ti	K	Ti
1 System with Multiple Equal Poles	n = 1	1,0000	1,0000	4,0000	1,0000
	n = 2	0,6988	1,4715	1,5946	1,4715
	n = 3	0,5648	1,8270	1,0821	1,8270
	n = 4	0,4855	2,1242	0,8495	2,1242
	n = 8	0,3375	3,0355	0,5083	3,0355
2 Fourth Order System	alfa = 0,1	0,8881	1,0054	2,7758	1,0054
	alfa = 0,2	0,7920	1,0239	2,0844	1,0239
	alfa = 0,5	0,5889	1,1855	1,1618	1,1855
	alfa = 1	0,4855	2,1242	0,8495	2,1242
3 System with Right Half Plane Zero	alfa = 0,1	0,5437	1,8240	1,0158	1,8240
	alfa = 0,2	0,5221	1,8154	0,9516	1,8154
	alfa = 0,5	0,4573	1,7597	0,7761	1,7597
	alfa = 1	0,3555	1,5932	0,5446	1,5932
	alfa = 2	0,1984	1,1447	0,2641	1,1447
	alfa = 3	0,1032	0,7277	0,1273	0,7277
4 First Order System with Dead Time	tau = 0,1	0,0769	0,1000	0,0930	0,1000
	tau = 0,2	0,1429	0,2000	0,1818	0,2000
	tau = 0,5	0,2941	0,5000	0,4255	0,5000
	tau = 2	0,6250	2,0000	1,2903	2,0000
	tau = 5	0,8065	5,0000	2,1739	5,0000
	tau = 10	0,8929	10,0000	2,8169	10,0000
5 Second Order System with Dead Time	tau = 0,1	0,1043	0,1472	0,1288	0,1472
	tau = 0,2	0,1815	0,2943	0,2384	0,2943
	tau = 0,5	0,3266	0,7358	0,4868	0,7358
	tau = 2	0,5439	2,9430	1,0164	2,9430
	tau = 5	0,6273	7,3576	1,2990	7,3576
	tau = 10	0,6611	14,7152	1,4317	14,7152

All MATLAB codes to compute PI controller parameters are attached to *Appendix*.

Chapter 7

Simulation on Labview

In this section, it has been run a simulation campaign on Labview with the problems on the benchmark with all triggering rules. To evaluate the results, a specific threshold and sampling time have been determined for each step.

To achieve the big comparison table of triggering methods, simulations have been executed by using The Labview Library defined in *Chapter 3*. To find which triggering mechanism is the best option for a specific benchmark process introduced in *Chapter 4* and a specific control problem defined in *Chapter 5*, each triggering methods have been tested.

To observe setpoint response, step reference signal with amplitude 1 has been given to all systems. To settle on the refence signal, threshold value generally has been chosen as 0.01. In some cases, the threshold value 0.01 has been considered very large for Energy SoD. Therefore, it has been changed to 0.0001 to be targeted the sample value almost equals with other triggering mechanisms.

The sampling time of the process and the controller has been chosen as at least ten times smaller than the smallest time constant to be safe. To simplification, it has been desired that event generator generates periodic events with period 500 millisecond.

To observe load disturbance response, when the system was in steady state, a load disturbance with amplitude 2 has been added to control signal for a very small-time interval. Then, recovery time and sample number have been observed till reaching steady state again.

To compare the triggering mechanism, control measurement indexes described in *Chapter 5* have been used. ISE and IAE values have been calculated for each step and then compared.

For instance, LABVIEW library and the control panel for Process 3 with parameter $\alpha = 3$ are shown in *Figure 7.1* and *Figure 7.2*.

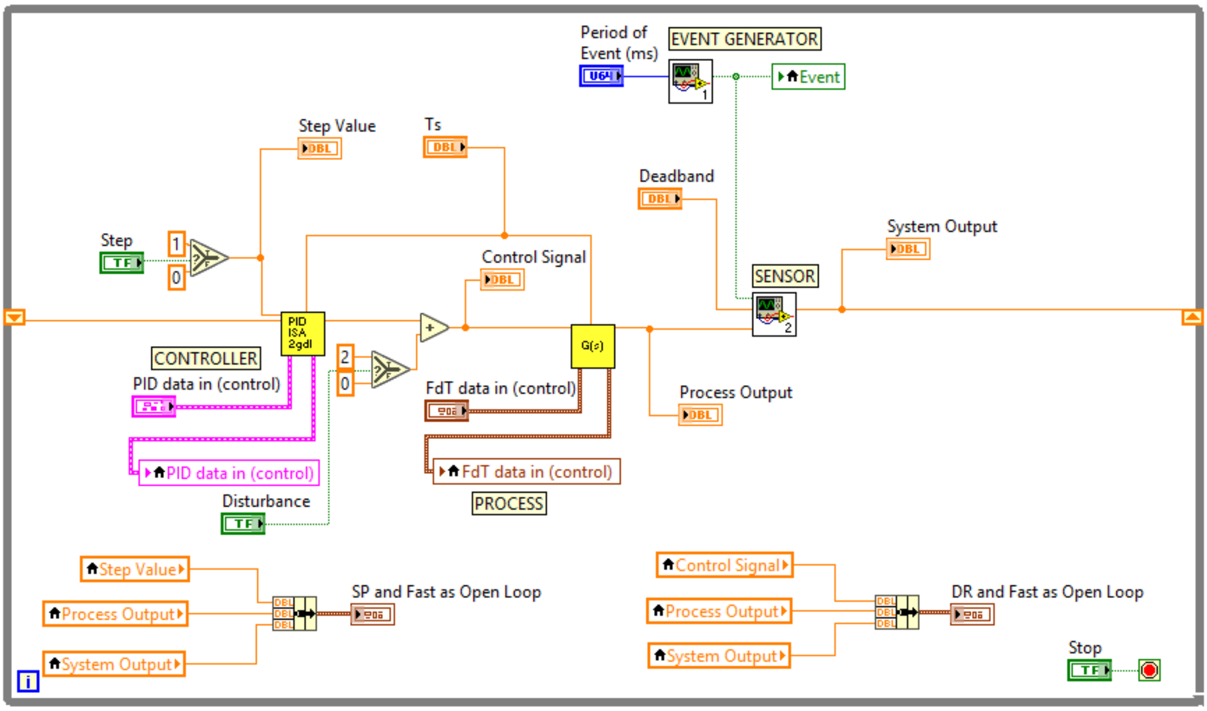

Figure 7.1: LABVIEW library for process 3 with $\alpha = 3$

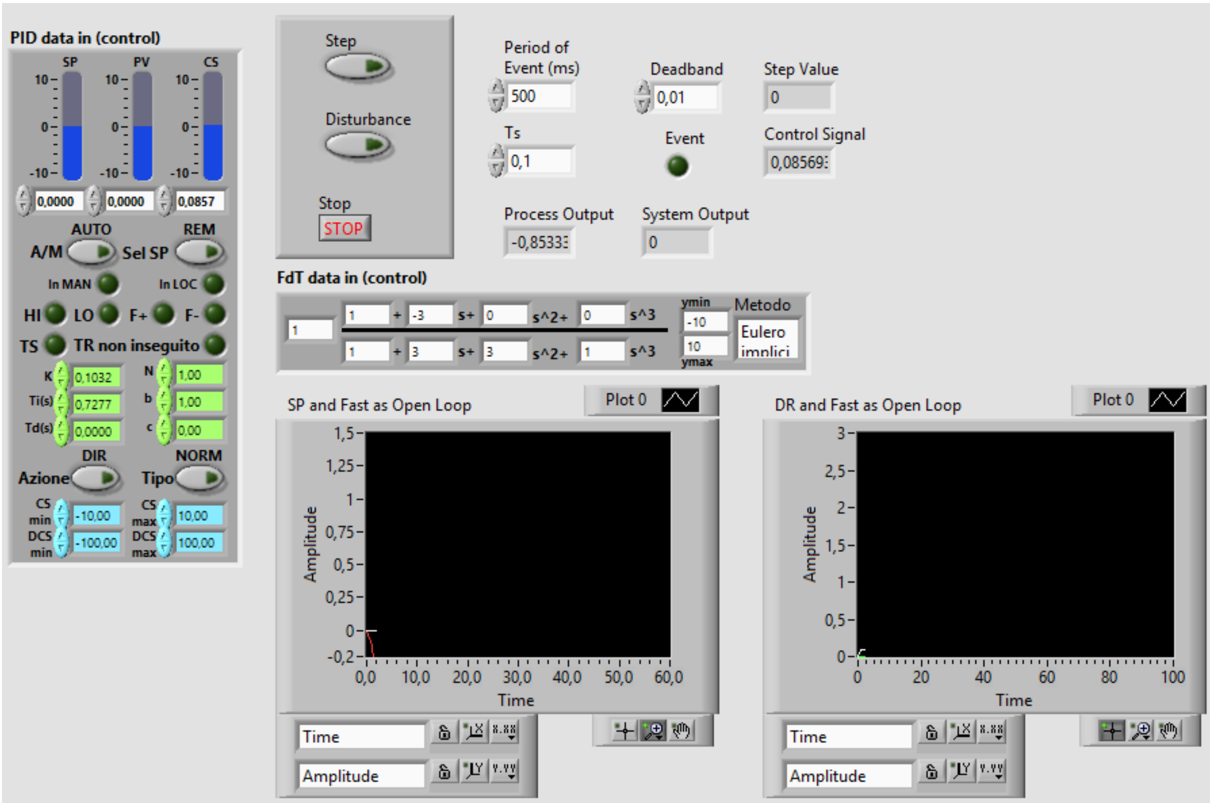

Figure 7.2: Control panel for process 3 with $\alpha = 3$

For Process 4 with parameter $\tau = 5$ and control problem setpoint tracking and closed loop as fast as open loop, the simulation results and specifications regarding the simulation are shown in *Figure 7.3*, *7.4*, *7.5*, *7.6* and *Table 7.1*.

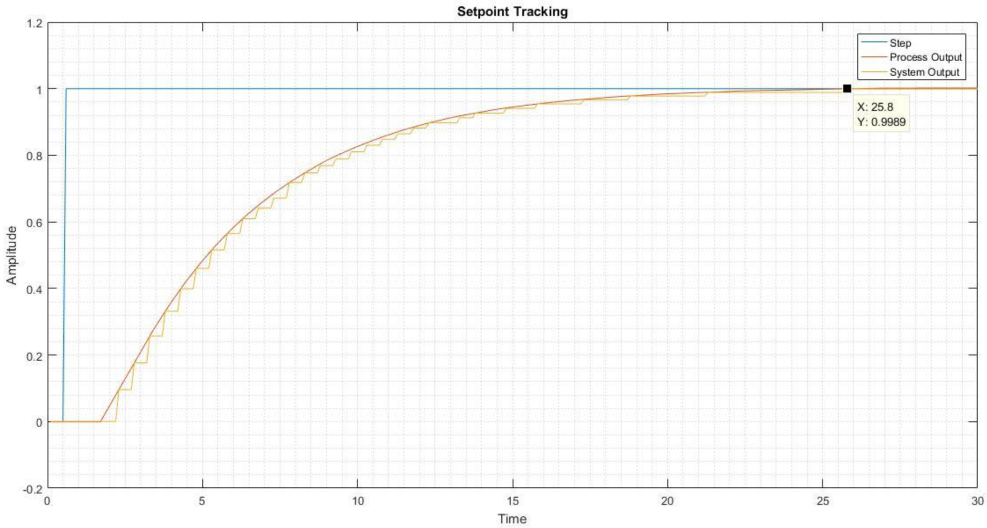

Figure 7.3: Constant deadband triggering mechanism simulation result for Process 4 with parameter $\tau = 5$ and control problem setpoint tracking and closed loop as fast as open loop

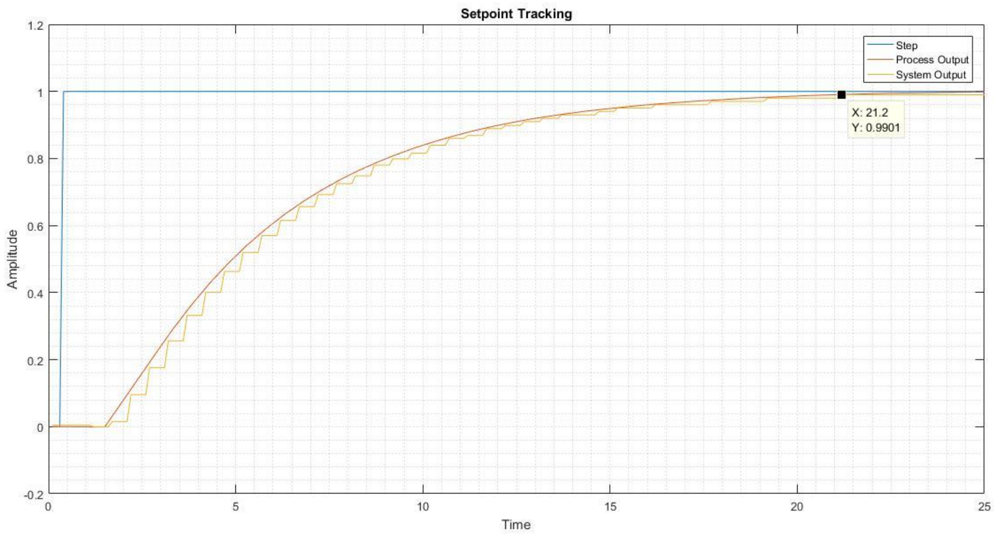

Figure 7.4: Relative deadband triggering mechanism simulation result for Process 4 with parameter $\tau = 5$ and control problem setpoint tracking and closed loop as fast as open loop

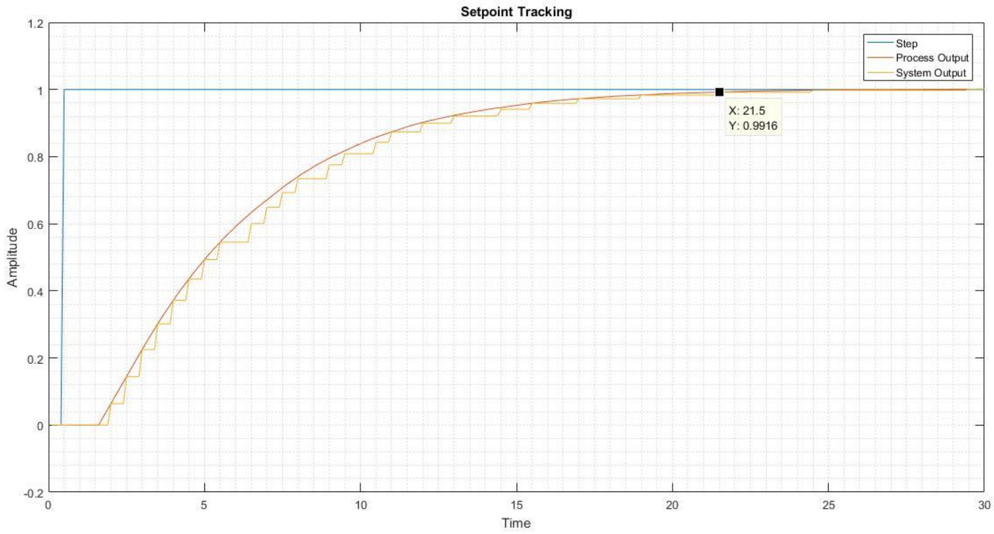

Figure 7.5: Integral SoD triggering mechanism simulation result for Process 4 with parameter $\tau = 5$ and control problem setpoint tracking and closed loop as fast as open loop

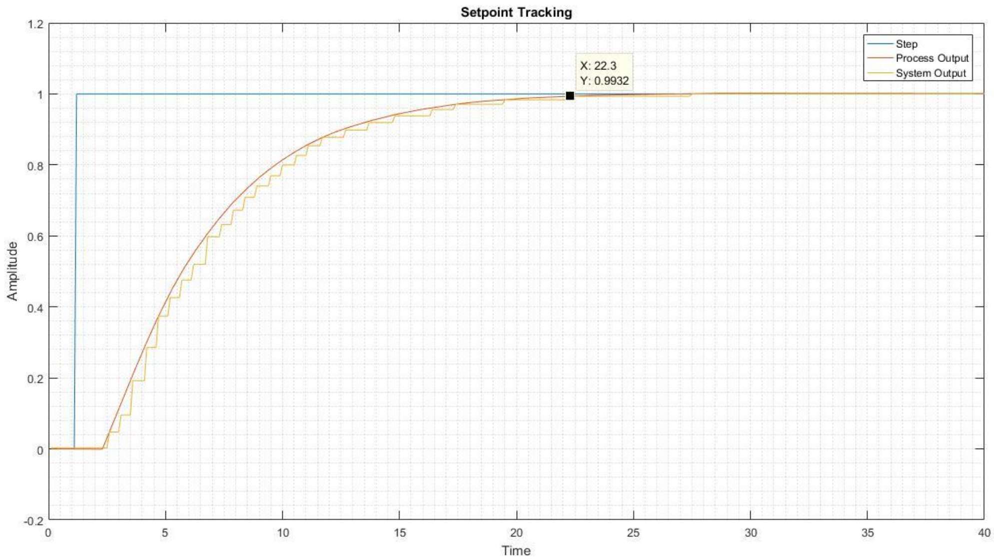

Figure 7.6: Energy SoD triggering mechanism simulation result for Process 4 with parameter $\tau = 5$ and control problem setpoint tracking and closed loop as fast as open loop

Table 7.1: Specifications of simulation result for Process 4 with parameter $\tau = 5$ and control problem setpoint tracking and closed loop as fast as open loop

Specifications	Triggering Mechanisms			
	Constant	Relative	Integral	Energy
Deadband	0,01	0,01	0,01	0,0001
IAE	6,2187	6,2086	6,2132	6,2225
ISE	3,8892	3,9343	3,9166	3,9595
Settling Time	25,3	31	21,1	21,2
# of Samples	29	28	23	25

Based on *Table 7.1* and *Figures 7.3*, *7.4*, *7.5*, *7.6*, the best triggering mechanism has been chosen as Integral SoD for this specific benchmark process with control problem setpoint tracking and closed loop as fast as open loop since, performance indexes of integral triggering method are better than the other ones.

This method of comparison has been fulfilled to all other processes with a specific control problem.

Control performance specifications of all processes to depend on a parameter are indicated in *Table 7.2, 7.3, 7.4* and *7.5.*

Table 7.2: Process 1 control performance indexes

	Specifications		Triggering Mechanisms				Specifications		Triggering Mechanisms			
			Constant	Relative	Integral	Energy			Constant	Relative	Integral	Energy
n = 1	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,0041	1,0442	1,1985	1,4008		IAE	N.S	1,6778	N.S	N.S
		ISE	0,6671	0,6529	0,7294	0,7844		ISE	N.S	1,1916	N.S	N.S
		Settling Time	4,1	4	7,6	6		Settling Time	N.S	7,5	N.S	N.S
		# of Samples	7	10	7	6		# of Samples	N.S	14	N.S	N.S
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	0,9845	1,809	1,918	1,7848		IAE	N.S	4,1311	N.S	N.S
		ISE	1,0308	1,5782	1,9691	1,0426		ISE	N.S	6,612	N.S	N.S
		Recovery Time	5,5	6,5	5	5,1		Recovery Time	N.S	5,2	N.S	N.S
		# of Samples	8	11	8	6		# of Samples	N.S	9	N.S	N.S
n = 2	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	2,1449	2,2089	2,4397	2,4342		IAE	1,9129	2,1747	2,3398	2,4504
		ISE	1,5325	1,1834	1,6412	1,6115		ISE	1,0459	1,194	1,2096	1,9995
		Settling Time	8	13,3	9,5	7,2		Settling Time	14,2	13	19,4	15,1
		# of Samples	10	21	13	9		# of Samples	19	21	24	20
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,6428	1,7523	1,3643	3,2116		IAE	1,6247	1,5924	2,3844	3,8909
		ISE	1,4336	1,0825	0,946	2,7983		ISE	1,0435	1,0521	1,1748	2,5884
		Recovery Time	10	12	11,3	9,5		Recovery Time	8,6	11,4	26,2	25,4
		# of Samples	14	5	10	14		# of Samples	13	15	18	21
n = 3	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	3,5565	3,7308	3,7458	3,7812		IAE	3,7357	4,1012	4,1169	3,9574
		ISE	2,5636	2,6341	2,6889	2,6372		ISE	2,0701	2,237	2,221	2,0489
		Settling Time	12	19	17	18		Settling Time	28,5	27,9	31,1	33,6
		# of Samples	17	21	17	18		# of Samples	31	35	31	33
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,8477	1,626	1,383	2,0911		IAE	1,8624	2,8932	3,2945	2,4236
		ISE	1,4241	0,9029	0,8993	1,4456		ISE	0,9735	1,6547	1,7423	1,0659
		Recovery Time	13,1	12,7	13,5	13		Recovery Time	25,1	20,7	23	20
		# of Samples	18	15	12	15		# of Samples	17	28	20	18
n = 4	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	5,0563	5,2399	5,4045	5,3755		IAE	5,6535	6,0962	6,7213	6,5437
		ISE	3,6088	3,6969	3,7914	3,7586		ISE	3,1458	3,4529	3,6223	3,5415
		Settling Time	26,6	25	26	23		Settling Time	31,3	39,1	46	45,2
		# of Samples	22	29	22	24		# of Samples	38	49	45	46
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,3038	2,1088	1,4724	1,4698		IAE	2,2506	3,004	2,5834	2,729
		ISE	0,8802	1,402	0,8867	0,8976		ISE	0,9774	1,5431	1,0307	1,0256
		Recovery Time	16,8	17	17	15,4		Recovery Time	26,8	27,8	32,4	33
		# of Samples	14	22	13	13		# of Samples	26	30	22	23
n = 8	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	11,61	11,5152	12,2582	11,9338		IAE	14,27	14,6052	15,5121	14,9748
		ISE	8,44	8,5337	8,7139	8,4681		ISE	8,36	8,5178	8,8514	8,4868
		Settling Time	52,4	48,5	47,6	47,8		Settling Time	84,5	84,1	98,7	83
		# of Samples	44	47	40	42		# of Samples	76	78	74	86
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	2,4448	1,4995	1,7347	2,4692		IAE	2,3	2,7277	3,839	3,9851
		ISE	1,349	0,8522	0,8641	1,349		ISE	0,88	0,9207	1,462	1,5147
		Recovery Time	33,2	31,5	29	46,8		Recovery Time	43,4	67,2	55	57
		# of Samples	21	13	13	17		# of Samples	20	27	28	28

N.S.: System output does not settle to reference signal, since oscillations do not remain within the threshold

Table 7.3: Process 2 control performance indexes

Specifications			Triggering Mechanisms: Constant	Relative	Integral	Energy	Specifications		Triggering Mechanisms: Constant	Relative	Integral	Energy
α = 0,1	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	11,3574	11,825	11,57	11,3049		IAE	4,1581	3,9826	N.S	3,9705
		ISE	6,4178	6,4208	6,87	6,629		ISE	2,5738	2,6122	N.S	2,646
		Settling Time	48,2	48	38,5	42,5		Settling Time	10,2	9	N.S	9,4
		# of Samples	48	55	12	20		# of Samples	17	16	N.S	12
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,3222	1,1753	1,5352	1,4062		IAE	2,6725	1,7758	N.S	1,8077
		ISE	0,826	0,8214	0,8582	0,8418		ISE	1,3228	0,8668	N.S	0,9224
		Recovery Time	14	8,5	27,8	41		Recovery Time	8	6,1	N.S	20,3
		# of Samples	5	3	3	4		# of Samples	5	3	N.S	4
α = 0,2	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	6,7748	6,5609	6,6756	6,5442		IAE	3,2868	3,4463	4,2771	3,5513
		ISE	3,9897	4,0635	4,255	4,2439		ISE	2,0842	2,1592	2,4295	2,3023
		Settling Time	26,2	23	20	21,3		Settling Time	16,9	17,5	25	16,4
		# of Samples	32	33	12	19		# of Samples	21	21	13	15
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,6977	1,9121	1,4214	1,3196		IAE	2,1796	1,5382	3,041	2,3352
		ISE	1,3021	1,271	0,8558	0,8354		ISE	0,8751	0,8937	0,9459	1,4587
		Recovery Time	28,1	23,8	30	23		Recovery Time	14	12	20,7	14,7
		# of Samples	13	14	4	5		# of Samples	7	11	5	7
α = 0,5	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	4,2392	4,4093	4,5546	4,807		IAE	4,0705	4,282	4,8768	5,542
		ISE	3,0633	3,1083	3,2051	3,2249		ISE	2,3592	2,5133	2,6835	2,8728
		Settling Time	14,7	15	19,2	27,8		Settling Time	19,6	20	26	41,7
		# of Samples	18	19	16	13		# of Samples	26	33	27	24
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,6512	1,4957	1,5351	1,4467		IAE	2,0029	2,608	2,6002	3,3285
		ISE	0,8858	0,881	0,9055	0,8961		ISE	0,9892	1,5394	1,0499	1,0927
		Recovery Time	15	14	13,4	15,1		Recovery Time	15,7	16,5	23	43,3
		# of Samples	15	14	8	4		# of Samples	16	22	13	10
α = 1	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	5,0563	5,2399	5,4045	5,3371		IAE	5,6535	6,0962	6,7213	6,5725
		ISE	3,6088	3,6969	3,7914	3,815		ISE	3,1458	3,4529	3,6223	3,4945
		Settling Time	26,6	25	26	22,8		Settling Time	31,3	39,1	46	53
		# of Samples	22	29	22	22		# of Samples	38	49	45	49
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,3038	2,1088	1,4724	2,2609		IAE	2,2506	3,004	2,5834	3,508
		ISE	0,8802	1,402	0,8867	1,4267		ISE	0,9775	1,5431	1,0307	1,6557
		Recovery Time	16,8	17,1	17,25	16,4		Recovery Time	26,8	27,8	32,4	32,7
		# of Samples	14	22	13	18		# of Samples	26	30	22	27

N.S.: System output does not settle to reference signal, since oscillations do not remain within the threshold

Table 7.4: Process 3 control performance indexes

	Specifications		Triggering Mechanisms				Specifications		Triggering Mechanisms			
			Constant	Relative	Integral	Energy			Constant	Relative	Integral	Energy
α = 0,1	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	34,5	34,2552	35,6196	35,47		IAE	32,8523	32,3799	34,013	34
		ISE	25,0982	25,2936	25,7133	25,63		ISE	20,2458	19,2364	19,8419	19
		Settling Time	106,8	71	108	119,7		Settling Time	162,4	223	216,8	222,3
		# of Samples	109	90	21	29		# of Samples	164	220	33	48
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	3,2597	5,7377	3,33	2,9503		IAE	4,1511	5,5322	5,4321	6,03
		ISE	2,0159	3,7562	2,04	1,6532		ISE	2,1381	4,5984	4,8313	2,18
		Recovery Time	113,3	106,5	108,4	123		Recovery Time	87,7	89	136,5	164
		# of Samples	24	11	5	2		# of Samples	6	3	8	9
α = 0,2	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	18,29	18,5816	19,1149	19,0476		IAE	17,8263	18,1094	20,5787	18,65
		ISE	15,35	13,6918	13,7238	14,1233		ISE	10,4262	10,7207	11,7084	11,03
		Settling Time	58,1	55,7	90	59,5		Settling Time	114,5	90	137	89,2
		# of Samples	59	64	23	28		# of Samples	90	87	39	45
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,1524	1,5964	1,5776	2,0963		IAE	1,9305	1,9123	2,0801	3,3228
		ISE	0,8263	0,8339	0,831	1,263		ISE	0,817	0,8636	0,8601	1,2942
		Recovery Time	22	21,1	25	20,3		Recovery Time	48	20,5	22,3	77,7
		# of Samples	2	2	3	3		# of Samples	3	4	2	10
α = 0,5	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	8,8509	9,0233	9,1963	9,1209		IAE	9,4734	9,7797	10,8413	10,5275
		ISE	6,4832	6,4761	6,6505	6,7215		ISE	5,6735	5,8653	6,0525	6,1711
		Settling Time	28	36,4	42	42,7		Settling Time	51,9	54,3	78,8	65
		# of Samples	30	38	24	29		# of Samples	56	63	46	49
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	2,1961	1,6926	2,2833	1,5301		IAE	2,3565	2,0379	2,5511	2,5138
		ISE	1,3421	0,8653	1,332	0,8605		ISE	0,939	0,8895	0,9405	0,9196
		Recovery Time	27,1	27	30,3	24,7		Recovery Time	36,2	32,1	34,6	47,3
		# of Samples	21	14	11	9		# of Samples	24	20	13	15
α = 1	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	5,7945	6,0099	6,0891	6,0738		IAE	6,7173	7,2881	7,8672	7,6048
		ISE	4,2069	4,2957	4,3796	4,3361		ISE	4,1413	4,3903	4,6261	4,4069
		Settling Time	26	30,2	27,5	26		Settling Time	41,1	42,3	47,8	48,2
		# of Samples	29	33	26	25		# of Samples	50	54	47	47
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,7716	2,5468	1,8318	2,7835		IAE	3,6626	3,7788	2,9971	3,9509
		ISE	0,9518	1,4768	0,9452	1,5288		ISE	1,8055	1,7231	1,12	1,7887
		Recovery Time	25,2	17,1	17	22,5		Recovery Time	28,3	34,2	34	34,3
		# of Samples	23	26	16	21		# of Samples	32	39	29	32
α = 2	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	8,7931	9,0367	9,1963	9,14		IAE	10,7078	11,3303	11,8037	12,1066
		ISE	6,3845	6,5579	6,5354	6,67		ISE	6,9151	7,2415	7,4146	7,5756
		Settling Time	42,3	32,9	41	43,8		Settling Time	56,5	57	69	80,2
		# of Samples	45	45	41	42		# of Samples	65	70	63	67
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	2,3378	2,2276	3,4311	3,5439		IAE	4,1622	2,9448	3,4573	N.S
		ISE	1,0005	0,984	1,6624	1,6823		ISE	1,8465	1,0889	1,1437	N.S
		Recovery Time	30	31,2	31,1	44,6		Recovery Time	37,3	27,3	37,2	N.S
		# of Samples	26	28	28	31		# of Samples	38	34	32	N.S
α = 3	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	12,2804	12,5324	13,2203	12,9269		IAE	15,0541	15,8473	16,3984	16,5286
		ISE	9,1544	9,3577	9,4512	6,4277		ISE	10,0302	10,7324	10,8619	10,9752
		Settling Time	54	51	66,5	69,3		Settling Time	69,3	85,4	87	85,3
		# of Samples	57	62	55	55		# of Samples	80	78	80	81
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	3,8775	2,6483	2,7424	3,9542		IAE	3,3524	5,3518	3,7161	4,23
		ISE	1,7499	1,0303	1,0584	1,7081		ISE	1,1046	2,0154	1,1853	1,22
		Recovery Time	36,9	41,2	35,7	50		Recovery Time	47,3	52,2	60,1	76
		# of Samples	36	30	26	35		# of Samples	39	57	38	40

Table 7.5: Process 4 control performance indexes

Specifications			Triggering Mechanisms				Specifications		Triggering Mechanisms			
			Constant	Relative	Integral	Energy			Constant	Relative	Integral	Energy
τ = 0,1	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	13,094	12,9864	13,0298	13,14		IAE	10,8592	10,7772	10,7614	11,05
		ISE	7,2562	7,2751	7,8909	7,62		ISE	6,1589	6,1873	6,614	6,45
		Settling Time	56,7	57,3	47,1	49,5		Settling Time	46,3	48,1	39,2	36,4
		# of Samples	50	58	13	21		# of Samples	45	51	12	19
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,7191	1,0835	1,533	2,3351		IAE	1,3149	1,58	1,4631	2,3634
		ISE	1,3984	0,9311	1,0016	1,9604		ISE	0,8871	1,4764	0,96	2,1119
		Recovery Time	26,8	22,4	29	36		Recovery Time	24	27	27,7	27,8
		# of Samples	12	9	6	11		# of Samples	10	13	6	10
τ = 0,2	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	7,0404	7,0042	7,0631	7,161		IAE	5,5671	5,5191	5,5882	5,5929
		ISE	4,2919	4,3168	4,6062	4,457		ISE	5,5792	3,6101	3,8868	3,6449
		Settling Time	30,6	25,3	23,7	25		Settling Time	18,4	20,3	14,8	18,7
		# of Samples	34	35	13	20		# of Samples	25	27	10	17
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,1753	1,0697	2,4539	1,9149		IAE	1,0494	1,1398	1,5336	1,514
		ISE	0,9537	0,9035	2,1641	1,4437		ISE	0,8953	0,924	1,0485	0,9747
		Recovery Time	19,3	14,3	18	18,7		Recovery Time	13,6	15,8	13,9	12,9
		# of Samples	12	12	9	10		# of Samples	11	14	7	8
τ = 0,5	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	3,4386	3,4756	3,5737	3,5505		IAE	2,8764	3,1174	3,3339	3,2121
		ISE	2,5787	2,6471	2,6311	2,678		ISE	2,1857	2,2615	2,3688	2,2722
		Settling Time	8,2	7,6	12,7	7,1		Settling Time	8,3	13,2	14,8	13
		# of Samples	13	13	12	11		# of Samples	13	18	13	14
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	0,6871	1,2556	1,5218	2,1455		IAE	1,3539	2,9287	1,886	2,1284
		ISE	0,4533	0,9887	1,067	1,6508		ISE	1,0062	2,5973	1,127	1,1877
		Recovery Time	8,8	10,5	9,9	11,3		Recovery Time	8,5	8,5	19,5	20
		# of Samples	11	14	9	12		# of Samples	14	15	13	15
τ = 2	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	3,2567	3,329	3,3761	3,3902		IAE	3,4447	3,8912	4,2123	4,053
		ISE	2,5053	2,5702	2,5981	2,5332		ISE	2,1449	2,3891	2,3704	2,336
		Settling Time	6,8	6,2	12,4	11,5		Settling Time	17,2	17,3	26,1	22
		# of Samples	11	10	13	13		# of Samples	27	28	30	26
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,1882	1,2609	1,2927	1,4158		IAE	1,852	2,2282	3,115	3,2994
		ISE	0,9249	0,93359	0,9557	1,0055		ISE	1,121	1,2412	1,9211	1,3704
		Recovery Time	11,6	12	10,8	11,1		Recovery Time	15	14,4	24,1	44
		# of Samples	13	13	10	12		# of Samples	21	22	20	32
τ = 5	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	6,2187	6,2086	6,2132	6,2225		IAE	2,8865	3,0742	3,1498	3,1369
		ISE	3,8892	3,9343	3,9166	3,9595		ISE	2,1567	2,2116	2,2778	2,2147
		Settling Time	25,3	20,9	21,1	21,2		Settling Time	8,2	12,2	13,7	12,2
		# of Samples	29	31	23	25		# of Samples	13	17	15	14
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,223	1,6696	1,6548	1,75		IAE	1,2516	2,5291	1,5278	2,1222
		ISE	0,867	1,3316	1,3305	1,3266		ISE	0,9559	2,1535	0,999	1,5343
		Recovery Time	18,5	23,5	21,2	13,7		Recovery Time	10,7	11,3	10,8	10
		# of Samples	9	14	11	10		# of Samples	10	15	8	9
τ = 10	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	11,2366	11,1767	11,1942	11,2		IAE	3,5457	3,6033	3,6187	3,6522
		ISE	6,3899	6,3611	6,5031	6,41		ISE	2,606	2,6846	2,7344	2,7422
		Settling Time	49,1	46,9	43,6	47		Settling Time	9,7	8,3	8,9	8
		# of Samples	46	53	35	38		# of Samples	15	14	12	13
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,3284	1,7078	1,5209	1,6361		IAE	1,5077	1,6833	1,6518	1,4413
		ISE	0,8437	1,2693	1,2616	1,2679		ISE	1,3728	1,3807	1,3667	0,8589
		Recovery Time	39,7	35	28,6	20,5		Recovery Time	6,8	17,7	15,8	4,5
		# of Samples	7	9	9	8		# of Samples	6	9	7	2

Table 7.6: Process 5 control performance indexes

	Specifications		Constant	Relative	Integral	Energy	Specifications		Constant	Relative	Integral	Energy
			Triggering Mechanisms						Triggering Mechanisms			
τ = 0,1	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	14,253	13,013	14,154	14,26		IAE	11,518	10,4393	11,5473	11,5156
		ISE	8,903	7,004	8,66	8,44		ISE	6,817	5,8497	7,4282	7,0545
		Settling Time	61,9	53,7	46,7	49		Settling Time	42,3	49,7	43,6	46,3
		# of Samples	55	58	13	21		# of Samples	47	50	12	20
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	2,7837	1,2142	1,9445	1,7691		IAE	1,2708	1,3014	1,8452	1,8889
		ISE	2,3618	0,8593	1,3819	1,3556		ISE	0,8617	0,8659	1,3626	1,3745
		Recovery Time	25,4	29,2	31,6	35,1		Recovery Time	20,2	26,4	28,9	26,5
		# of Samples	14	14	16	8		# of Samples	12	13	6	8
τ = 0,2	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	8,4952	7,5513	8,2078	8,1461		IAE	6,2793	6,2703	6,3145	6,2867
		ISE	5,1932	4,4028	5,5298	5,2921		ISE	4,2473	4,2933	4,5257	4,3359
		Settling Time	32	31	25,3	25,9		Settling Time	20	18,2	18,7	16,1
		# of Samples	38	36	13	20		# of Samples	26	28	11	17
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,3644	1,4481	1,4023	1,7757		IAE	2,1143	1,9201	1,8844	1,3092
		ISE	0,8757	0,8805	0,8954	1,3591		ISE	1,3808	1,3756	1,3954	0,8884
		Recovery Time	18,9	21,2	19,4	23,5		Recovery Time	17,8	16,9	16,7	16,8
		# of Samples	14	14	6	9		# of Samples	18	17	7	8
τ = 0,5	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	4,6438	4,9048	4,8421	4,8066		IAE	4,1983	4,5567	5,0082	4,5692
		ISE	3,44	3,5331	3,5569	3,5724		ISE	2,8877	3,048	3,2982	3,1013
		Settling Time	10	22,3	17,2	15,3		Settling Time	16	17,6	24,2	26,1
		# of Samples	17	18	15	16		# of Samples	21	23	20	21
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,5414	2,3091	1,5204	1,4527		IAE	1,734	1,718	2,7675	2,5658
		ISE	0,9099	1,4427	0,9273	0,9207		ISE	0,963	0,9689	1,5972	1,613
		Recovery Time	13	14,8	14,8	13,3		Recovery Time	11,6	17,2	25,5	18,7
		# of Samples	15	20	10	11		# of Samples	16	18	15	17
τ = 2	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	6,0357	6,1012	5,9481	6,0278		IAE	5,9631	6,5026	6,9669	6,4455
		ISE	4,294	4,3839	4,3249	4,4546		ISE	3,537	3,7626	3,8746	3,6762
		Settling Time	19	19	20,1	19,2		Settling Time	37,8	36,2	51,1	43,5
		# of Samples	22	24	21	22		# of Samples	41	45	44	42
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,5917	1,5384	1,9607	1,4016		IAE	2,1452	2,218	3,2842	3,3231
		ISE	0,8652	0,8904	1,3548	0,8756		ISE	0,9331	0,9421	1,5777	1,599
		Recovery Time	19,2	17,8	19	18,7		Recovery Time	21	22,8	29,8	38,3
		# of Samples	14	13	14	12		# of Samples	20	19	22	25
τ = 5	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	11,8627	11,7594	11,7662	11,8494		IAE	9,4962	9,5583	9,8347	9,83
		ISE	8,3124	8,4115	8,5681	8,5125		ISE	5,8374	5,8378	5,9545	6,0062
		Settling Time	29	29,4	27	27,3		Settling Time	54	43	54,1	53,2
		# of Samples	44	45	30	36		# of Samples	56	54	46	45
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,5744	1,3357	1,6993	1,7213		IAE	2,1192	2,2906	1,6172	2,48118
		ISE	0,8294	0,8263	1,2484	1,2422		ISE	0,8631	1,2937	0,8568	1,3558
		Recovery Time	38,5	33,4	37	38,8		Recovery Time	30	31,2	26	28,8
		# of Samples	7	5	12	9		# of Samples	30	10	9	12
τ = 10	SP and as Fast as OL	Deadband	0,01	0,01	0,01	0,0001	SP and Faster than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	22,55	22,3865	21,9196	22,7144		IAE	15,8084	15,7082	16,0241	15,9152
		ISE	15,5245	15,1115	15,2352	15,9337		ISE	9,97967	9,629	9,8786	9,8044
		Settling Time	71	63	55,7	58,6		Settling Time	81,2	77,7	74,7	75,2
		# of Samples	72	79	44	48		# of Samples	64	73	53	59
	DR and Fast as OL	Deadband	0,01	0,01	0,01	0,0001	DR and Faster Than OL	Deadband	0,01	0,01	0,01	0,0001
		IAE	1,1931	1,2141	0,9985	2,4005		IAE	2,0608	2,2607	2,0125	2,007
		ISE	0,8132	0,8173	0,8149	1,6291		ISE	1,236	0,8259	1,2505	1,2451
		Recovery Time	70,8	24,9	17,4	67,3		Recovery Time	58,7	52	47,3	46,1
		# of Samples	6	2	4	10		# of Samples	5	3	11	7

Chapter 8

Analysis of The Results

The simulation results on Labview were compared for each process generated by using all triggering methods. In event based structures, the best triggering mechanism was identified for each process and each control problem.

In consequence of many simulation, the comparison of triggering mechanisms in event based control is shown *Table 8.1*. To achieve same performance with other triggering mechanisms, deadband value of triggering method has been decreased sometimes. For instance, in some cases of the energy send-on delta triggering method simulation, it has been observed that the system output never settles to reference signal value with band %1, since the square of error value gives a smaller value than the error. Furthermore, in some tests, the steady state error or the oscillations are detected by the sampling operations, since these oscillations do not remain within the threshold.

As a result, the constant deadband triggering method appears as dominant triggering mechanism, but the results also reveal that for some unusual cases other triggering mechanisms show better performance.

In the future, the study can be extended by inserting the other triggering mechanisms presented in the literature.

Table 8.1: Comparison table of triggering mechanisms in event based control

PROCESS CLASS	Parameter	CONTROL		PROBLEMS	
		Setpoint Tracking & Fast as Open Loop	Setpoint Tracking & Faster Than Open Loop	Disturbance Rejection & Fast as Open Loop	Disturbance Rejection & Faster Than Open Loop
1	n = 1	Basic SoD (Constant Deadband)	Relative Deadband	Basic SoD (Constant Deadband)	Relative Deadband
System with	n = 2	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Integral SoD	Basic SoD (Constant Deadband)
Multiple	n = 3	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Integral SoD	Basic SoD (Constant Deadband)
Equal	n = 4	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)
Poles	n = 8	Energy SoD	Basic SoD (Constant Deadband)	Relative Deadband	Basic SoD (Constant Deadband)
2	alfa = 0.1	Integral SoD	Relative Deadband	Relative Deadband	Relative Deadband
Fourth	alfa = 0.2	Integral SoD	Basic SoD (Constant Deadband)	Integral SoD	Relative Deadband
Order	alfa = 0.5	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Integral SoD	Basic SoD (Constant Deadband)
System	alfa = 1	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)
3	alfa = 0.1	Integral SoD	Integral SoD	Basic SoD (Constant Deadband)	Integral SoD
System	alfa = 0.2	Energy SoD	Relative Deadband	Basic SoD (Constant Deadband)	Relative Deadband
with	alfa = 0.5	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Energy SoD	Relative Deadband
Right	alfa = 1	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Integral SoD	Integral SoD
Half Plane	alfa = 2	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Relative Deadband	Relative Deadband
Zero	alfa = 3	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Integral SoD	Basic SoD (Constant Deadband)
4	tau = 0.1	Integral SoD	Integral SoD	Relative Deadband	Basic SoD (Constant Deadband)
First	tau = 0.2	Integral SoD	Integral SoD	Relative Deadband	Basic SoD (Constant Deadband)
Order	tau = 0.5	Integral SoD	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)
System	tau = 2	Relative Deadband	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)
With	tau = 5	Integral SoD	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)
Deadtime	tau = 10	Integral SoD	Integral SoD	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)
5	tau = 0.1	Integral SoD	Integral SoD	Relative Deadband	Relative Deadband
Second	tau = 0.2	Relative Deadband	Integral SoD	Basic SoD (Constant Deadband)	Integral SoD
Order	tau = 0.5	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)	Basic SoD (Constant Deadband)
System	tau = 2	Integral SoD	Basic SoD (Constant Deadband)	Relative Deadband	Basic SoD (Constant Deadband)
With	tau = 5	Integral SoD	Relative Deadband	Relative Deadband	Integral SoD
Deadtime	tau = 10	Integral SoD	Integral SoD	Integral SoD	Integral SoD

Bibliography

[1] Leva, A.; Papadopoulos, A.V. (2013). Tuning of event-based industrial controllers with simple stability guarantees. *Journal of Process Control, 23*(9), 1251-1260. doi: 10.1016/j.jprocont.2013.07.010

[2] Arzén, K.E. A simple event-based PID controller. *Proceedings of IFAC World Congress* **1999**, *18*, 423-428.

[3] Otanez, P.; Moyne, J.; Tilbury, D. Using deadbands to reduce communication in networked control systems. *Proceedings of American Control Conference ACC'2002,* Anchorage, USA, **2002**, *4*, 3015-3020.

[4] Diaz-Cacho, M., Delgado, E., Barreiro, A., & Falcón, P. (2017). Basic Send-on-Delta Sampling for Signal Tracking-Error Reduction. Sensors, 17(2), 312. doi:10.3390/s17020312

[5] Miskowicz, M. Send-On-Delta Concept: An Event-Based Data Reporting Strategy. Sensors **2006**, 6, 49–63.

[6] Guinaldo, M.; Dimarogonas, D.V.; Johansson, K.H.; Sánchez, J.; Dormido, S. Distributed event-based control strategies for interconnected linear systems. IET Control Theory Appl. **2013**, 7, 877–886.

[7] Åström, K.J.; Bernhardsson, B. Comparison of Periodic and Event Based Sampling for First-Order Stochastic Systems; Citeseer: Forest Grove, OR, USA, 1999.

[8] W. Heemels, K. Johansson, P. Tabuada, An introduction to event-triggered and self-triggered control, in: 2012 IEEE 51st Annual Conference on Decision and Control (CDC), 2012, pp. 3270–3285.

[9] Vasyutynskyy, V.; Kabitzsch, K. Implementation of PID controller with send-on-delta sampling. *International Conference Control*, Glasgow, Scotland, UK, 2006.

[10] Soo, Y. Send-on-delta sensor data transmission with a lineal predictor. *Sensors* **2007**, *7*, 537–547.

[11] J. Sánchez, M.A. Guarnes, S. Dormido, "On the application of different event-based sampling strategies to the control of a simple industrial process", *Sensors*, vol. 9, pp. 6795-6818, 2009.

[12] V. Vasyutynskyy, K. Kabitzsch, “Towards comparison of deadband sampling types”, *Proc. of IEEE International Symposium on Industrial Electronics ISIE 2007*, pp. 2899-2904, 2007.

[13] *Contributions to networked and event-triggered control of linear systems = Contribuciones al control en red basado en eventos para sistemas lineales* (Unpublished master's thesis). Thesis / Dissertation ETD. (2013).

[14] Miskowicz, M. Event-based sampling strategies in networked control systems. In Proceedings of the 10th IEEEWorkshop on Factory Communication Systems (WFCS), Toulouse, France, 5–7 May 2014; pp. 1–10.

[15] Sandra, H.; Peter, H.; Eckehard, S.; Martin, B. Network Traffic Reduction in Haptic Telepresence Systems by Deadband Control. In *Proceedings IFAC World Congress*; International Federation of Automatic Control: Prague, Czech Republic, 2005.

[16] Miśkowicz, M. The event-triggered integral criterion for sensor sampling. *Proceedings of IEEE International Symposium on Industrial Electronics* **2005**, 1061-1066.

[17] Miśkowicz, M. Asymptotic Effectiveness of the Event-Based Sampling according to the Integral Criterion. *Sensors* **2007**, *7*, 16-37.

[18] M. Miskowicz. Sampling of signals in energy domain. In *ETFA - 10th IEEE Int. Conf. on Emerging Technologies and Factory Automation*, volume 1, pages 263–266, Catana, Italy, 19.–20. Sept. 2005.

[19] Ploennigs, J.; Vasyutynskyy, V.; Kabitzsch, K. Comparison of Energy-Efficient Sampling Methods for WSNs in Building Automation Scenarios. In *Proceedings of IEEE International Conference on Emerging Techonologies and Factory Automation ETFA*, Palma de Mallorca, Spain, 2009; pp. 1-8.

[20] M. Miśkowicz, “Efficiency of event-based sampling according to error energy criterion”, *Sensors*, vol. 10, no.3, pp. 2242-2261, 2010.

[21] Nguyen, V.H.; Suh, Y.S. Networked Estimation with an Area-Triggered Transmission Method. *Sensors* **2008**, *8*, 897-909.

[22] K. Åström, T. Hagglund, Benchmark systems for PID control, in: IFAC Workshop on Digital Control – Past, Present, and Future of PID Control, Terrassa, Spain, 2000.

[23] Grimholt, C.; Skogestad, S. (2012). Optimal PI-Control and Verification of the SIMC Tuning Rule. *IFAC Proceedings Volumes, 45*(3), 11-22. doi:10.3182/20120328-3-it-3014.00003”

[24] Skogestad, S. (2004). Simple analytic rules for model reduction and PID controller tuning. *Modeling, Identification and Control: A Norwegian Research Bulletin, 25*(2), 85-120. doi:10.4173/mic.2004.2.2

[25] A. O'Dwyer. Handbook of PI and PID Controller Tuning Rules. Imperial College Press, London, UK, 2009.

[26] Skogestad, S. (2001). Probably the best simple PID tuning rules in the world, *AIChE Annual Meeting* (Reno, Nevada, U.S.A.) [Online]. Available at http://www.nt.ntnu.no/users/skoge/publications/2001/skogestad_reno/tuningpaper.pdf [accessed 24 September 2004].

[27] Alfaro, V. M., & Vilanova, R. (2012). Performance/Robustness Trade-off Design Framework for 2DoF PI Controllers. *Studies in Informatics and Control, 21*(1). doi:10.24846/v21i1y201209

[28] B, D., M, G. A., R, B., R, J. D., & P, A. (2015). IMC Based Design of PI Controller for Real Time Pressure Process. *Ijireeice, 3*(10), 12-15. doi:10.17148/ijireeice.2015.31003

[29] Skogestad, S., & Grimholt, C. (2012). The SIMC Method for Smooth PID Controller Tuning. *PID Control in the Third Millennium Advances in Industrial Control*, 147-175. doi:10.1007/978-1-4471-2425-2_5

[30] Garpinger, O., Hägglund, T., & Åström, K. J. (2012). Criteria and Trade-offs in PID Design. *IFAC Proceedings Volumes, 45*(3), 47-52. doi:10.3182/20120328-3-it-3014.00008

[31] Fu, A., Kartakis, S., Mazo, M., & McCann, J.A. (2016). Evaluation of Decentralized Event-Triggered Control Strategies for Cyber-Physical Systems. *CoRR, abs/1611.04366*.

[32] Dimarogonas, D.V., Dormido, S., Guinaldo, M., Johansson, K.H., & Moreno, J.S. (2011). Distributed event-based control for interconnected linear systems. *CDC-ECE*.

[33] Dormido, S., Jiménez, J.E., Ruiz, A.B., & Sánchez, J. (2015). Characterization and tuning of predictive SSOD-PI controllers. *2015 23rd Mediterranean Conference on Control and Automation (MED)*, 551-557.

[34] SWIG, "Data standards and protocols for real-time communications in the water industry," http://www.swig.org.uk/past-swig-events/2014-2/, 2015, online; accessed 22-June-2015.

[35] Weber's Law of Just Noticeable Differences. (n.d.). Retrieved November 18, 2017, from http://apps.usd.edu/coglab/WebersLaw.html

[36] LabVIEW. (n.d.). Retrieved November 18, 2017, from http://www.ni.com/en-tr/shop/labview/buy-labview.html

[37] Skogestad S., “Simple analytic rules for model reduction and PID controller tuning,” *Journal of Process Control*, vol. 13, pp. 291–309, 2003.

Appendix

Appendix A.1: To compute PI controller parameters, Matlab code for Process 1

Appendix A.2: To compute PI controller parameters, Matlab code for Process 2

Appendix A.3: To compute PI controller parameters, Matlab code for Process 3

Appendix A.4: To compute PI controller parameters, Matlab code for Process 4

Appendix A.5: To compute PI controller parameters, Matlab code for Process 5

Appendix A.1

```
%SYSTEM WITH MULTIPLE EQUAL POLES

s = tf('s');

n = 1; %System order depends on value "n"
P1 = 1/(s+1)^n %System transfer function

%To obtain first order model with delay time
mu = 1
T = (exp(1-n)*n^n)/(factorial(n-1))
D = n-((exp(1-n)*n^n)/(factorial(n-1)))

FOPDT = exp(-s*D)*mu/(1+s*T)

%To obtain PI controller parameters
oloopst = D+5*T; %Open loop settling time for 1% band
a = 1 %Closed loop is "a" time faster than open loop
lambda = (D+5*T)/(5*a); %Closed loop time constant
cloopst = 5*lambda; %Closed loop settling time

%PI controller parameters
Ti = T %Integral time
K = T/(mu*(lambda+D)) %Controller gain

C = K*(1+(1/(Ti*s))) %Controller transfer function

Kp = K %Proportional gain of controller
Ki = K/Ti %Integral gain of controller
```

Appendix A.2

```
function bf = beta1(alfa,k)
bf = exp(-(alfa+1)*(alfa^2+1)/(alfa^k)) - 1;
end
```

```
%FOURTH ORDER SYSTEM

s = tf('s');

alfa = 0.1; %System poles depend on value "alfa"
P2 = 1/((1+s)*(1+alfa*s)*(1+alfa^2*s)*(1+alfa^3*s))

%To obtain first order model with delay time
mu = 1

if alfa ~= 1
A0 = (alfa+1)*(alfa^2+1);
beta2 = (alfa-1)^3*(alfa+1);

A1 = A0-((beta1(alfa,2)*alfa^5 - beta1(alfa,1)*alfa^2)/beta2)...
    +((beta1(alfa,3)*alfa^9 ...
    -beta1(alfa,0))/(beta2*(alfa^2 + alfa + 1)));
else
    A0 = 4;
    A1= 128/3*exp(-4);
end

T = exp(1)*A1/mu
D = A0/mu - T

FOPDT = exp(-s*D)*mu/(1+s*T)

%To obtain PI controller parameters
oloopst=D+5*T; %Open loop settling time for 1% band
a=4; %Closed loop is "a" time faster than open loop
lambda=(D+5*T)/(5*a); %Closed loop time constant
cloopst=5*lambda; %Closed loop settling time

%PI controller parameters
Ti=T %Integral time
K=T/(mu*(lambda+D)) %Controller gain

C=K*(1+(1/(Ti*s))) %Controller transfer function

Kp=K %Proportional gain of controller
Ki=K/Ti %Integral gain of controller
```

Appendix A.3

```
%SYSTEM WITH RIGHT HALF PLANE ZERO

s=tf('s');

alfa = 3; %System zero depends on value "alfa"
P3 = (1-alfa*s)/(s+1)^3

%To obtain first order model with delay time
mu = 1;
T = 1/2*exp(-alfa-2)*(alfa+3)^3
D = alfa + 3 - 1/2*exp(-alfa-2)*(alfa+3)^3

FOPDT = exp(-s*D)*mu/(1+s*T)

%To obtain PI controller parameters
oloopst = D+5*T; %Open loop settling time for 1% band
a = 4 %Closed loop is "a" time faster than open loop
lambda = (D+5*T)/(5*a); %Closed loop time constant
cloopst = 5*lambda; %Closed loop settling time

%PI controller parameters
Ti = T %Integral time
K = T/(mu*(lambda+D)) %Controller gain

C = K*(1+(1/(Ti*s))) %Controller transfer function

Kp = K %Proportional gain of controller
Ki = K/Ti %Integral gain of controller
```

Appendix A.4

```
%FIRST ORDER SYSTEM WITH DEAD TIME

s = tf('s');
tau = 0.5; %System time constant depends on value "tau"
P4 = (exp(-s))/(1+s*tau) %System transfer function

%To obtain first oreder model with delay time
mu = 1;
T = tau
D = 1

FOPDT = exp(-s*D)*mu/(1+s*T)

%To obtain PI controller parameters
oloopst = D+5*T; %Open loop settling time for 1% band
a = 4 %Closed loop is "a" time faster than open loop
lambda = (D+5*T)/(5*a); %Closed loop time constant
cloopst = 5*lambda; %Closed loop settling time

%PI controller parameters
Ti = T %Integral time
K = T/(mu*(lambda+D)) %Controller gain

C = K*(1+(1/(Ti*s))) %Controller transfer function

Kp = K %Proportional gain of controller
Ki = K/Ti %Integral gain of controller
```

Appendix A.5

```
%SECOND ORDER SYSTEM WITH DEAD TIME

s = tf('s');
tau = 10; %System time constant depends on value "tau"
P5 = exp(-s)*1/(1+s*tau)^2 %System transfer function

%To obtain first oreder model with delay time
mu = 1;
T = 4*exp(-1)*tau
D = 1 + 2*tau*(1-2*exp(-1))

FOPDT = exp(-s*D)*mu/(1+s*T)

%To obtain PI controller parameters
oloopst = D+5*T; %Open loop settling time for 1% band
a = 4 %Closed loop is "a" time faster than open loop
lambda = (D+5*T)/(5*a); %Closed loop time constant
cloopst = 5*lambda; %Closed loop settling time

%PI controller parameters
Ti = T %Integral time
K = T/(mu*(lambda+D)) %Controller gain

C = K*(1+(1/(Ti*s))) %Controller transfer function

Kp = K %Proportional gain of controller
Ki = K/Ti %Integral gain of controller
```