A Comparative Study of Classical and Modern Controllers

Monu Malik[1],
[1]Department of Mechanical Engineering,
Ganga Institute of Technology and Management
Kablana

Neeraj Sehgal[2]
[2]Department of Mechanical Engineering,
Ganga Institute of Technology and Management
Kablana

A controller is a device which controls the input and output quantities of a system. The device is mostly utilized in controlling vibration of a plate. For this purpose a number of controllers are available which can be utilized according to their suitability. In this field, some most preferred controllers are: PID, LQR, LQG and H-infinity controllers. The classical controllers include PID controller whereas LQR, LQG and H infinity comes in modern controllers. Each controller has its specific characteristics regarding input-output, state space, collocation and settling time criteria. In this work our main motive is to study a detailed working of these controllers and making a comparison among them.

Keywords: PID, LQR, LQG and H-infinity controller

I. INTRODUCTION

The problem of controlling input-output parameters of actuators/sensors for the control of large flexible structures is of considerable interest in engineering design where an improved performance and efficiency of response is sought from each controller. All the controllers serves the same work as they control the input-output quantities but each work is different from other by some means. Through of them PID is a basic controller which works on the parameters of proportionality-integration-differentiation. Other controllers have their own working principle and processing techniques.

Sr.No.	CLASSICAL CONTROLLERS	MODERN CONTROLLERS
1	PID - CONTROLLER	H- INFINITY CONTROLLER
2	LQR -CONTROLLER	LQG- CONTROLLER
3	FUZZY-LOGIC CONTROLLER	REAL TIME CONTROLLER
4	NEURAL NETWORK	

II. PID CONTROLLER

A proportional-integral-derivative controller (PID controller) is a control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an error value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process through use of a manipulated variable.

The PID controller algorithm involves three separate constant parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. Simply put, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve, a damper, or the power supplied to a heating element. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the setpoint, and the degree of system oscillat

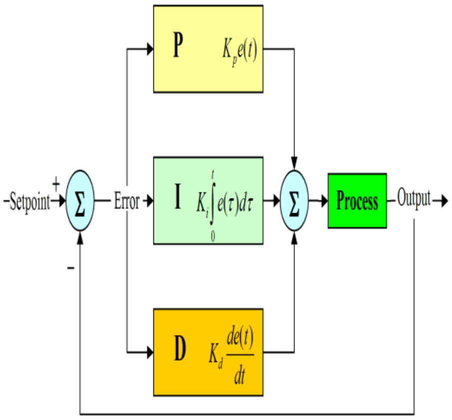

Fig. 1: A Block diagram for PID controller

Some applications may require using only one or two actions to provide the appropriate system control. This is achieved by setting the other parameters to zero. A PID controller will be called a PI, PD, P or I controller in the absence of the respective control actions. PI controllers are fairly common, since derivative action is sensitive to measurement noise, whereas the absence of an integral term may prevent the system from reaching its target value due to the control action.

PID Controller Theory:

The PID control scheme is named after its three correcting terms, whose sum constitutes the manipulated variable (MV). The proportional, integral, and derivative terms are summed to calculate the output of the PID controller. Defining $u(t)$ as the controller output, the final form of the PID algorithm is:

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t)$$

where

K_p: Proportional gain, a tuning parameter
K_i: Integral gain, a tuning parameter
K_d: Derivative gain, a tuning parameter
e: Error = $SP - PV$
t: Time or instantaneous time (the present)
τ: Variable of integration; takes on values from time 0 to the present t.

III. FUZZY LOGIC SYSTEM

In recent years, the number and variety of applications of fuzzy logic have increased significantly. The applications range from consumer products such as cameras, camcorders, washing machines, and microwave ovens to industrial process control, medical instrumentation, decision-support systems, and portfolio selection.

To understand why use of fuzzy logic has grown, you must first understand what is meant by fuzzy logic.Fuzzy logic has two different meanings. In a narrow sense, fuzzy logic is a logical system, which is an extension of multivalued logic. However, in a wider sense fuzzy logic (FL) is almost synonymous with the theory of fuzzy sets, a theory which relates to classes of objects with unsharp boundaries in which membership is a matter of degree. In this perspective, fuzzy logic in its narrow sense is a branch of FL. Even in its more narrow definition, fuzzy logic differs both in concept and substance from traditional multivalued logical systems.

In Fuzzy Logic Toolbox software, fuzzy logic should be interpreted as FL, that is, fuzzy logic in its wide sense. The basic ideas underlying FL are explained very clearly and insightfully in Foundations of Fuzzy Logic. What might be added is that the basic concept underlying FL is that of a linguistic variable, that is, a variable whose values are words rather than numbers. In effect, much of FL may be viewed as a methodology for computing with words rather than numbers. Although words are inherently less precise than numbers, their use is closer to human intuition. Furthermore, computing with words exploits the tolerance for imprecision and thereby lowers the cost of solution.

Another basic concept in FL, which plays a central role in most of its applications, is that of a fuzzy if-then rule or, simply, fuzzy rule. Although rule-based systems have a long history of use in Artificial Intelligence (AI), what is missing in such systems is a mechanism for dealing with fuzzy consequents and fuzzy antecedents. In fuzzy logic, this mechanism is provided by the calculus of fuzzy rules. The calculus of fuzzy rules serves as a basis for what might be called the Fuzzy Dependency and Command Language (FDCL). Although FDCL is not used explicitly in the toolbox, it is effectively one of its principal constituents. In most of the applications of fuzzy logic, a fuzzy logic solution is, in reality, a translation of a human solution into FDCL.

A trend that is growing in visibility relates to the use of fuzzy logic in combination with neurocomputing and genetic algorithms. More generally, fuzzy logic, neurocomputing, and genetic algorithms may be viewed as the principal constituents of what might be called soft computing. Unlike the traditional, hard computing, soft computing accommodates the imprecision of the real world. The guiding principle of soft computing is: Exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, and low solution cost. In the future, soft computing could play an increasingly important role in the conception and design of systems whose MIQ (Machine IQ) is much higher than that of systems designed by conventional methods.

Among various combinations of methodologies in soft computing, the one that has highest visibility at this juncture is that of fuzzy logic and neurocomputing, leading to neuro-fuzzy systems. Within fuzzy logic, such systems play a particularly important role in the induction of rules from observations. An effective method developed by Dr. Roger Jang for this purpose is called ANFIS (Adaptive Neuro-Fuzzy Inference System). This method is an important component of the toolbox

IV. NEURAL NETWORK CONTROL

Now we will train neural network according to some input and corresponding output of the PID controller and neural network will predict any input, the corresponding output in the future. An

Artificial neural network is a highly simplified model of the structure of the biological neural network. An ANN consists of interconnected processing units. These are called neuron. The diagram of neuron is as follows.

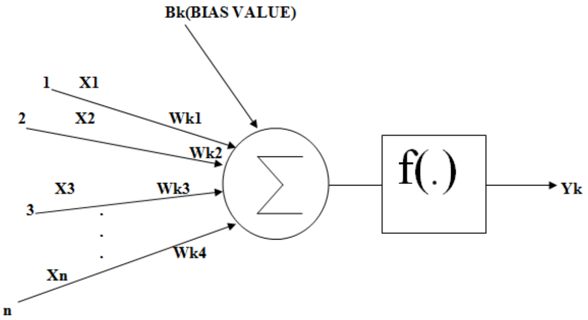

Figure 2: Diagram of a Single Artificial Neuron.

Where x1, x2, x3 and so on xn are the inputs to the neuron, wk1, wk2, wk3,. . . .wkn are corresponding weights and bk is bias. Yk is output of the neuron and f(.) is any function on which a neuron will response or will give the output. These are connected to each other to form a network. The response is obtained on the randomly selected input termed as training data. The general model of interconnected processing units consists of summing part followed by an output part. The summing part receives N input values and weights each value and compute a weighted sum. The weighted sum is called the activation value. The output part produces a signal from activation value. The sign of the weight for each input determines whether the input is excitatory (positive weight) or inhibitory (negative weight).The input could be discrete or continuous data values and likewise the outputs also could be discrete or continuous.

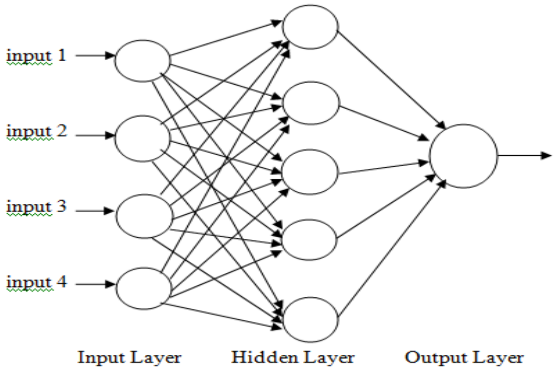

Figure 3: Diagram of a Single Artificial Neuron.

The neural networks used in the current research are known as multilayer perceptron (MLP). The MLPs comprise a layer of input signals, one or more hidden layers of neurons, and an output layer of neurons. A layer consists of a single or multiple neurons. Differences between the desired outputs (targets) and the network outputs give the errors. The connection strengths ('weights') and 'biases' are updated during training (or learning) such that the network produces the desired output for the given input.

Mathematically, neuron *j* having *n* inputs is described as follows:

$$\mathrm{y_j} = \varphi_j(\mathrm{v_j})$$

$$\text{Where } v_j = \left(\sum_{i=1}^{m} w_{ji}x_i + b_j\right) \text{ and}$$

$$x_i - inputsignals, w_{ji} - weightsandb_jisbias$$

In this way we design neural network controller for active vibration control of a cantilever plate with piezo-patches as sensor and actuator. For a given set of inputs to the network, outputs are computed for each neuron in the first layer and forwarded to the next layer. The signals propagate on a layer-by-layer basis until the output layer is reached. The weights and biases remain unchanged during the 'forward pass'. The output of the network is compared with the desired value (t_j) and difference gives the error.

$$e_j = t_j - y_j$$

The total error is defined by given equation, where *c* is the number of neurons in the output layer.

$$\mathrm{E} = \frac{1}{2}\sum_{\mathrm{j=1}}^{\mathrm{c}} \mathrm{e_j^2}$$

The error *E* represents the cost function, and the weights and biases are updated to minimize it.

The computed partial derivatives (sensitivity) $\partial E/\partial w_{ji}$determine the search direction for updating the weights w_{ji}as

$$w_{ji}(k+1) = w_{ji}(k) - \eta\frac{\partial E(k)}{\partial w_{ji}(k)}$$

*k*is the current time and (*k* + 1) is the next time step. The above update formula is refined using a 'momentum term' that has a stabilizing effect on the back-propagation algorithm. The training of the NN is complete when the error (or change in the error) reduces to a predetermined small value. ANN is robust controller. It can work even in the condition when environmental condition changes like temperature and pressure. When the NN controller is trained with inputs and outputs with a very less error then we will get the same output as before changing condition. It is also useful in the conditions like when initial conditions are changing and forces are changing.

V. H_∞ CONTROLLER

The control inputs were collocated with disturbances and the control outputs were collocated with the performance in case of LQG controller design. This assumption showing the limited utilization of LQG controller, resulting in lesser possibilities and applications. Where the system performance is evaluated the locations of control inputs do not always coincide with the disturbance locations, and the locations of controlled outputs are not necessarily collocated with the location. This was discussed earlier, when the generalized structure was introduced. The H_∞ controller addresses the controller design problem in its general configuration of non-collocated disturbance and control inputs as well non-collocated performance and control outputs. The H_∞ method addresses a large range of the control problems with combining the frequency- and time-domain approaches. By the minimization of the H norm of the closed-loop transfer function the design becomes an optimal one. The H_∞ controller includes colored measurement and process noise, addresses the issues of robustness due to model uncertainties. It is applicable to the single-input–single-output systems as well as to the multiple-input–multiple output systems.

In this article we present the H_∞ controller design for flexible structures. We chose the modal approach to H_∞ controller design, which allows for the determination of a stable reduced-order H_∞ controller with performance close to the full-order controller.

DEFINITION AND GAINS (WODEK K. GAWRONSKI [95])

In Fig.5 a closed-loop system architecture is shown In which *G* is the transfer function of a plant (or structure), *K* is the transfer function of a controller, *w* is the exogenous input (such as commands, disturbances), *u* is the actuator input, *z* is the regulated output (at which performance is evaluated), and *y* is the sensed (or controlled) output. This system is different from the LQG control system besides the actuator input and controlled output it has disturbance input and the regulated output. So we can say that it represents a broader class of systems than the LQG control system.

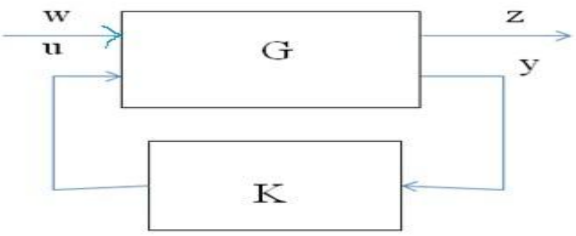

Fig: 4 H∞ Closed-loop system

The H∞ closed-loop system configuration:
G-plant,
K-controller,
u- actuator input,
w-exogenous input,
y-sensed output, and
z-regulated output.

For a closed-loop system as in Fig.6.1
the plant transfer function $G(s)$: $\begin{pmatrix} z(s) \\ y(s) \end{pmatrix} = G(s) \begin{pmatrix} w(s) \\ u(s) \end{pmatrix}$
and the controller transfer function $K(s)$ a $u(s) = K(s)y(s)$ (1)

where$u$, w are control and exogenous inputs and y, z are measured and controlled outputs, respectively and the state-space equations of a structure are as under :

$$\{\dot{x}\}=[A]\{x\} + [B_1]\{w\} + [B_2]\{u\}, \quad (2)$$

$$\{z\} = [C_1]\{x\} + [D_{12}]\{u\}, \quad (.3)$$

$$\{y\} = [C_2]\{x\} + [D_{21}]\{w\}. \quad (.4)$$

Hence, the state-space representation in the H∞ controller description consists of the quintuple (A, B1, B2, C1, C2). For this representation (A, B2) is stabilizable and (A,C2) is detectable.

Real- Time Controller

A Compact Rio real-Time controller is mostly preferable controller in LABVIEW that is shown in below figure. This is supplied with a controlled voltage through a power supply battery. This controller carries some slots on which DAC cards are mounted. A specific DAC card is used for controlling a specific parameter. The slots on which a DAC card is mounted is known as chassis. The controller having its own memory for storing a program in this and can be directly connected to the computer system.

Fig. 5 :CompactRio real-time controller

VI. COMPARISON WORK

H∞methods are used in control theory to synthesize controllers achieving stabilization with guaranteed performance but on the other hand in PID controller it is not guaranteed.

To use H∞ methods, a control designer expresses the control problem as a mathematical optimization problem and then finds the controller that solves this optimization but in PID controller there is no requirement of mathematical formulation.

H∞ techniques have the advantage over classical control techniques in that they are readily applicable to problems involving multivariate systems with cross-coupling between channels.

Disadvantages of H∞ techniques include the level of mathematical understanding needed to apply them successfully and the need for a reasonably good model of the system to be controlled.

REFERENCES

[1] Chhabra Deepak, BhushanGian, and ChandnaPankaj, 2011 “Design and Analysis of Smart Structures for Active Vibration Control using Piezo-Crystals”. International Journal of Engineering and Technology Volume 1 No. 3, ISSN2049-3044.

[2] Chhabra Deepak, BhushanGian, and ChandnaPankaj, 2014, “Optimization of Collocated/Noncollocated Sensors and Actuators along with Feedback Gain Using Hybrid Multiobjective Genetic Algorithm-Artificial Neural Network,” Chinese Journal of Engineering, vol., Article ID 692140, 12 pages, 2014.

[3] Chhabra Deepak, BhushanGian, ChandnaPankaj, 2013, “Optimal Placement of Piezoelectric Actuators on Plate Structures for Active Vibration Control Using Modified Control Matrix and Singular Value Decomposition Approach” International Journal of Mechanical Science and Engineering Vol:7 No:3.

[4] Singh Pardeep, Chhabra Deepak, 2012, “Design and Analysis of Smart Cantilever Plate Structures for Active Vibration Control using Piezoelectric Sensor and Actuator” International Journal of Mobile &Adhoc Network, Vol2: issue 2,pp. 312-320

[5] Chhabra Deepak, NarwalKapil, Singh Pardeep, 2012 “Design and Analysis of Piezoelectric Smart Beam for Active Vibration Control” , International Journal of Advancements in Research & Technology, Volume 1.

[6] Kumar Amit, Chhabra Deepak, 2013, “Design of Neural Network Controller for Active Vibration control of Cantilever plate with piezo-patch as sensor /actuator” International Journal of Modern Engineering Research (IJMER) www.ijmer.com Vol. 3,pp-2481-2488.

[7] Kumar Varun, Chhabra Deepak, 2013, “Design of Fuzzy Logic Controller for Active Vibration Control of Cantilever Plate with Piezo -Patches As Sensor /Actuator” International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-2, Issue-8) August 2013.

[8] SehgalNeeraj, Malik Monu, Chhabra Deepak, 2014, "Meta-heuristics Approaches for the Placement of Piezoelectric actuators/Sensors on a Flexible Cantilever Plate: A Review" International Journal of Enhanced Research In Science Technology & Engineering, Vol. 3, pp: (7-16),ISSN: 2319-7463

[9] S. Narayanana, V. Balamurugan, 2003. Finite element modeling of piezo laminated smart structure for active vibration control with distributed sensor and actuators. Journal of Sound and Vibration, Vol. 262, pp 529-562.

[10] Mukherjee A. et al. 2002. Active vibration control of piezolaminated stiffened plates, Composite Structures, Vol. 55, pp 435–443.Vasques, C. M. A., & Rodrigues, J. D, 2006. Active vibration control of smart Piezoelectric beams: comparison of classical and optimal feedback control strategies. Computers & Structures, Vol. 84, pp 1402-1414.