Comparison of two approaches to automated PI controller tuning for an industrial weigh belt feeder

Yanan Zhao, * Emmanuel G. Collins, Jr. †

Department of Mechanical Engineering, Florida A&M University–Florida State University, Tallahassee, Florida 32310, USA

(Received 17 September 2003; accepted 27 March 2004)

Abstract

In this paper, two advanced PI controller tuning methods, unfalsified control and fuzzy control, are applied to an industrial weigh belt feeder that has significant nonlinearities. Both methods do not require an explicit plant model. The advantage of the unfalsified PI control design method is that it is able to directly incorporate multiple performance criteria, while the advantage of fuzzy logic is that it is able to directly incorporate human reasoning in the design process. Experimental results exhibit the effectiveness of both control methods. A detailed comparison of the two approaches is given in the areas of allowed design specifications, process knowledge requirements, computational requirements, controller development effort, transient performance, and the ability to handle motor saturation. © 2004 ISA—The Instrumentation, Systems, and Automation Society.

Keywords: Fuzzy logic control; Unfalsified control; PI control; Weigh belt feeder

1. Introduction

Automated controller tuning is of significant interest to control engineers since it can lead to increased performance while saving time and energy [1,2]. The original work on tuning focused on the development of procedures and recommendations to be used manually by operators. However, more sophisticated methods of autotuning have since been proposed. For example, automatic tuning is now used extensively in PID control [1,3,4].

PID controllers have been the most commonly used controllers in industrial control practice for the past 60 years, even though great progress in control theory has been made over this period. This is because of their simple structure, ease of design and implementation, and transparent interpretation. A poorly tuned control system may waste energy and cause excessive and unnecessary wear of actuators [2]. High performance is always the design target in industrial control applications and recently many modern control techniques that improve PID tuning have been reported [1,5,6]. In this research, automated PI controller tuning techniques are studied for an industrial weigh belt feeder.

The weigh belt feeder used in this research (see Fig. 1) was designed and manufactured by Merrick Industries, Inc. of Lynn Haven, Florida. It is a process feeder that is typically used in a food, chemical, or plastics manufacturing process. To ensure a constant feed rate in industrial operation, a PI control law is designed and implemented in the Merrick controller. In current practice the PI tuning process is performed manually by an engineering technician. However, for better and more consistent quality, it is desired to use automated PI tuning [7].
The dynamics of the weigh belt feeder are dominated by the motor. To protect the motor, the control signal is restricted to lie in the interval $[0, 10]$ V. The motor also has significant friction. In addition, the sensors exhibit significant quantization noise. Hence the weigh belt feeder exhibits nonlinear behavior. The system nonlinearities make standard tuning methods difficult to apply. For example, in attempting to apply the Ziegler-Nichols tuning method to the weigh belt feeder, the system saturated before the ultimate gain was obtained. Relay feedback autotuning [1,8], another widely used method, does not perform well for highly nonlinear systems or systems with large disturbances. When relay feedback tuning was attempted, because of the sensor quantization noise and the motor friction, the desired square wave with symmetric positive and negative half cycles could not be achieved, even after considering the relay hysteresis and compensating for part of the load disturbance. There are many other methods of PID controller tuning in the literature, usually based on knowledge of the process parameters. Due to the nonlinearities of the weigh belt feeder, its process parameters change with time and set point. For example, friction is highly nonlinear and depends on multiple parameters that vary during the process [9]. In an effort to design PI controllers for this type of process, two non-model-based PI tuning methods were studied in this research: unfalsified PI control and fuzzy PI control. Both methods do not require an explicit plant model, hence they are suitable for control design for nonlinear plants that are difficult to model.

The unfalsified design concept [10,11] points out that the control law can be obtained directly from a set of candidate controllers by using stored sensor output signals and actuator input signals. Consequently, it is not necessary for a controller to actually be inserted in the feedback loop in order to be falsified. Thus the adaptive unfalsified control processes may be significantly less susceptible to poor transient response than other processes that require inserting controllers in the loop. Another particularly attractive feature of unfalsified control is the ability to find control laws that can meet multiple objectives. For the weigh belt feeder, the objectives are to design PI controllers to minimize the transient and steady-state errors of the feed rate and to avoid actuator saturation at the same time. To reduce the computational time used for the unfalsification, a genetic algorithm was adopted to perform the unfalsification.

Fuzzy logic control provides a formal methodology for representing, manipulating, and implementing a human’s heuristic knowledge about how to control a system [12]. It has been found particularly useful for controller design when the plant model is unknown or difficult to develop. It does not need an exact process model and has been shown to be robust with respect to disturbances, large uncertainty, and variations in the process behavior [13]. Two types of fuzzy logic controllers (FLC’s), PI-like FLC’s (including gain scheduled and self-tuning PI-like FLC’s) and PI FLC’s, were designed for the weigh belt feeder [14]. PI-like FLC’s do not have explicit proportional and integral gains; instead the control signal is directly deduced from the knowledge base and the fuzzy inference. In contrast, PI FLC’s are composed of the conventional PI control system in conjunction with a set of fuzzy rules (knowledge base) and a fuzzy reasoning mechanism to tune the PI gains online. Since the PI FLC’s outperform the PI-like FLC’s for the weigh belt feeder [14] and they have explicit P and I gains, only PI FLC’s are compared with unfalsified PI controllers.

Both unfalsified PI control and fuzzy logic PI control are effective solutions to systems with time-varying parameters and other uncertainties. In this paper, a detailed comparison of the two methods from the point of view of allowed design specifications, process knowledge requirements, computational requirements, controller development effort, transient performance and the ability to handle motor saturation is given. This comparison can help a control engineer choose the most suitable type of controller for a given application.

The paper is organized as follows. Section 2 describes unfalsified PI control design with genetic

---

**Fig. 1.** The Merrick weigh belt feeder.
algorithm implementation for the weigh belt feeder. Section 3 describes fuzzy logic PI control for the weigh belt feeder. Section 4 presents a detailed comparison of the two control design methods. Finally, Section 5 presents some conclusions.

2. Unfalsified PI control

The unfalsified control concept is used here as a means of using either open- or closed-loop test data to identify a subset of controllers (from an initial set) that is not proved to violate the multiple objectives specified by the control engineer [11]. Consider the feedback control system of Fig. 2. Given $K = \{K_1, K_2, \ldots, K_N\}$, a finite, initial set of causally left-invertible control laws, the goal of unfalsified control is to determine a $K_i \in K$ ($i = 1, 2, \ldots, N$) such that for a set of plants $P$, representing the variations of the real system, the closed-loop system response satisfies a set of performance specifications involving the command signal $r(t)$, the command input $u(t)$, and the measured output $y(t)$.

Let $(\vec{y}, \vec{u})$ denote a set of open- or closed-loop test data, taken with sample period $h$ in the time interval $[t_0, t_0+ph]$ where $p$ is some positive integer, and $\vec{r}_{K_i}$ is the set of command signals that would have yielded the signals $\vec{u}$ and $\vec{y}$ if the controller $K_i$ were in the loop. Assume that there exist scalar cost functions $J_j: \mathbb{R}^{(p+1)} \times \mathbb{R}^{(p+1)} \times \mathbb{R}^{(p+1)} \rightarrow \mathbb{R}, j = 1, \ldots, q$, and associated performance specifications [7]

$$J_j(\vec{r}_{K_i}, \vec{y}, \vec{u}) \leq \bar{\sigma}_j, \quad j = 1, \ldots, q.$$  

Then, for each control law $K_i$ ($i = 1, 2, \ldots, N$) one can compute the set of costs $\{J_j(\vec{r}_{K_i}, \vec{y}, \vec{u})\}_{j=1}^{q}$. A controller is falsified if there exists $j \in \{1, 2, \ldots, q\}$ such that

$$J_j(\vec{r}_{K_i}, \vec{y}, \vec{u}) > \bar{\sigma}_j,$$  

2.1. Performance specifications for the weigh belt feeder

The cost functions used in the unfalsification process are based on the actual engineering goals and the observed closed-loop performance of the feeder. As previously mentioned, the control signal $u(t)$ is restricted to lie in the interval $[0, 10]$ V. Also, in practice, $r(t) \leq 5$ V (corresponding to a belt speed of $2.54 \times 10^{-2}$ m/sec). It was experimentally observed that saturation occurs if the following constraint is violated:

$$J_1(\vec{r}_{K_i}, \vec{u}) = \rho \max_{t \in T} \frac{|u(t)|}{\min_{t \in T} |r(t)|} \leq 2, \quad (3)$$  

where $T = (t_0, t_0+h, \ldots, t_0+ph)$ and $\rho$ takes on different values for different set points. In particular, $\rho$ increases as the set point increases. The number 2 in Eq. (3) is the ratio of the absolute value of the saturation control signal (10 V) and the maximum reference signal (5 V). Since actuator saturation is a hard constraint (i.e., it must be satisfied for proper operation of the controlled system), the constraint (3) is also treated as a hard constraint.

To achieve a good step response (i.e., low overshoot and fast settling time) another cost function is needed for unfalsification. The cost function constructed using experimental observations is

$$J_2(\vec{r}_{K_i}, \vec{y}) = \frac{\|\vec{r}_{K_i} - \vec{y}\|_2}{\|\vec{r}_{K_i}\|_2} + \rho_1 K_p + \rho_2 K_i, \quad (4)$$  

where $K_p$ and $K_i$ are, respectively, the proportional and integral gains of the PI controller. The terms $\rho_1 K_p$ and $\rho_2 K_i$ are used to take into account the transient overshoot of the system. The parameters $\rho_1$ and $\rho_2$ are selected to make the latter two terms in Eq. (4) comparable in magnitude to the first term.

In the unfalsified tuning process for the weigh belt feeder a controller is falsified if Eq. (3) is violated. Among $K_u$, the current set of unfalsified controllers, the “best” controller is considered to be the controller that satisfies

$$\min_{K \in K_u} J_2(\vec{r}_{K}, \vec{y}). \quad (5)$$
2.2. Experimental results with a genetic algorithm implementation

A genetic algorithm (GA) [15,16], which may be viewed as a random search process, can be used to avoid computing the costs for each of the unfalsified controllers and hence can increase the computational efficiency of the unfalsified control process. The unfalsified control problem described above can be viewed as a constrained optimization problem, i.e., solve the optimization problem (5) subject to the constraint (3). A GA was adapted to such a constrained optimization problem for unfalsified PI control implementation as described in Ref. [7].

Due to the nonlinearities in the dynamics of the feeder, specifically the friction and actuator saturation, one fixed controller does not perform well for each set point. Hence different PI controllers need to be designed for different set points. As mentioned above, 5 V is the maximum possible value of the reference command and PI controllers were designed for set points equal to 1, 2, 3, 4, and 5 V, respectively. The initial set of candidate proportional and integral gains were both chosen within the range of $[0.1, 3.2]$ with the grid corresponding to 0.1.

A closed-loop data set, generated from the actual hardware, was used for the unfalsified control design process for each of the five set points. The detailed experimental setup is described in Ref. [17]. Fig. 3 shows the corresponding step responses of the plant output and the control signal at set points of 1 and 5 V (the curves are similar under the other set points). Clearly the control signals are within the range of saturation and the system outputs have desirable performance (i.e., no overshoot and fast rise times). The results in this research clearly demonstrated the ability of the unfalsification procedure to yield high performing PI controllers. Also, the results demonstrated the ability of the GA to reduce the computational requirements of an exhaustive search, especially as the size of the initial candidate set increased.

3. Fuzzy PI control

Fuzzy PI control can be classified into two major categories according to their constructions: fuzzy PI-like controllers and fuzzy PI controllers [14]. In the first category, a fuzzy PI-like controller is constructed as a set of heuristic control rules. That is, the control signal or the incremental change of the control signal is built as a nonlinear function of the error, change of error and acceleration error, where the nonlinear function includes fuzzy reasoning. Thus there are no explicit proportional and integral gains; instead the control signal is directly deduced from the knowledge base and the fuzzy inference. They are referred to as fuzzy PI-like controllers because their structure is analogous to that of the conventional PI controller. In the second category, a fuzzy PI controller is composed of the conventional PI control system in conjunction with a set of fuzzy rules (knowledge...
base) and a fuzzy reasoning mechanism to tune the proportional and integral gains online. In this paper, only the second category of controllers (i.e., conventional PI controllers with fuzzy tuned P and I gains) is introduced as they outperformed PI-like fuzzy logic controllers for the weigh belt feeder [14] and have explicit proportional and integral gains.

### 3.1. System diagram

Fig. 4 shows the system diagram of the controller. For fuzzy PI control, the control signal is generated according to the online tuning of the P and I gains based on the transfer function,

$$H(z) = K_p + K_i T_s \frac{z}{z-1} = K_p \left(1 + \frac{T_i}{T_s} \frac{z}{z-1}\right),$$

where $K_p$ is the proportional gain, $K_i$ is the integral gain, $T_i = K_p / K_i$ is the integral time constant, and $T_s$ is the sampling period. It is assumed that $K_p$ is in the prescribed range $[K_{p,\text{min}}, K_{p,\text{max}}]$; the appropriate range is determined experimentally.

There are two fuzzy logic reasoning systems included in the diagram. The first one has two inputs, the error $e(k)$ and change of error $\Delta e(k)$, which are defined by $e(k) = r(k) - y(k)$ and $\Delta e(k) = e(k) - e(k-1)$, where $r$ and $y$ denote the applied set point input and plant output, respectively. Indices $k$ and $k-1$ indicate the present state and the previous state of the system, respectively. The output is the proportional gain $K_p$. The second fuzzy system has the same inputs, but the output is the integral time constant $T_i$. The integral gain $K_i$ was then obtained by $K_p / T_i$.

### 3.2. Membership functions and rule bases

To make implementation of a fuzzy logic controller possible with limited processor throughput, this research focused on reducing the number of fuzzy sets and thereafter the number of fuzzy rules. Here, the membership functions (MF’s) for $e$ and $\Delta e$ are defined on the common normalized domain $[-1,1]$, where each has three fuzzy sets N (negative), ZE (zero), and P (positive) as shown in Fig. 5. The two MF’s of $K_p$, corresponding to the fuzzy sets S (small) and B (big), are shown in Fig. 6. The MF’s of $T_i$, corresponding to the singleton fuzzy sets S (small), M (medium), and B (big), are also shown in Fig. 6. Below, it is explained how the singleton MF’s can be adjusted for different set points.

The fuzzy rules in Tables 1 and 2 are based on the desired characteristics of the step responses. For example, at the beginning of the control action, a big control signal is needed in order to achieve a fast rise time. Thus the PI controller should have a large proportional gain and a large integral gain. When the step response reaches the set point, a small control signal is needed to avoid a large overshoot. Thus the PI controller should have a small proportional gain and a small integral gain.

---

Fig. 4. Closed-loop system with fuzzy PI controller.

Fig. 5. Membership functions for $e$ and $\Delta e$. 

---
3.3. Tuning algorithm for the fuzzy PI controller

The scaling factors which describe the particular input normalization and output denormalization play a role similar to that of the gains of a conventional controller. Hence they are of utmost importance with respect to controller stability and performance [18]. The relationship between the scaling factors $G_e$ and $G_{\Delta e}$ and the input variables of the FLC are $e_N = G_e e$ and $\Delta e_N = G_{\Delta e} \Delta e$. Selection of suitable values of these scaling factors are made based on expert knowledge about the process to be controlled and through trial and error. Adjustment rules have been developed for the scaling factors by evaluating control results [19,20]. Here, the scaling factors $G_e = 0.1$ and $G_{\Delta e} = 1$ were chosen.

There are two other tuning algorithms used for the fuzzy PI controller design. Due to the nonlinearity of the feeder, to avoid high overshoot at higher set points, it is necessary to suitably reduce both the proportional gain and the integral gain. Hence we chose a gain scheduling coefficient $r_3 = \frac{1}{(1 + 0.2 \times sp)}$, where $sp$ stands for set point. This coefficient was used for the online tuning of both the range of $K_p$ and the MF’s of $T_i$. For different set points the range of the proportional gain was chosen as $[0, K_{p,\text{max}}]$, where $K_{p,\text{max}} = p_3 \times K_{p,\text{max}_0}$, and $K_{p,\text{max}_0} = 3.2$ was chosen according to experimental experience.

The singleton membership function of $T_i$ was adjusted online along with the set point, $mf = mf_0 / r_3$. Thus these MF’s shift right as the set point increases, while they shift left as the set point decreases as shown in Fig. 6, where the solid lines represent the MF’s at the 5 V set point and the dotted lines represent the MF’s at the 1 V set point. Here, the singleton values chosen for $sp = 0$ is $S = 0.75$, $M = 1$, and $B = 1.5$. Sugeno-type inference was used for the fuzzy reasoning of $T_i$.

3.4. Experimental results

The maximum possible sampling rate of the data acquisition hardware is 0.001 sec; however, the processor overload point may be much higher than 0.001 sec depending on the size of the model, the type of simulation integration algorithms and so on. Thus the sampling time is chosen as $T_s = 0.01$ sec in the experimentation. It is observed that this sampling rate can avoid processor over-
load while allowing acceptable performance to be achieved.

Fig. 7 shows the experimental results for the fuzzy PI controllers implemented for set points of 1 and 5 V. It is clearly that the desired performance (fast rise time, no overshoot) is achieved. The performance is similar at the other set points.

4. Comparison of unfalsified PI control and fuzzy PI control

The unfalsified control and fuzzy control provide two different approaches for PI controller tuning for the weigh belt feeder. In the following, the two approaches are compared in detail and possible future research for each method is described.

4.1. Proportional and integral gains

For unfalsified PI control, proportional and integral gains are obtained by selecting the best PI controller among the PI candidate set that satisfies the performance specifications. Since the unfalsified controllers were implemented off-line, both the proportional and integral gains are constant at a fixed set point.

In contrast, fuzzy PI controllers obtain their proportional and integral gains online based on the
error and change of error signal at each sampling period, thus both gains are time varying. Fig. 8 shows the changes of the proportional and integral gains of the fuzzy PI controllers as a function of time at set points of 1 and 5 V (the curves are similar at the other set points). It is seen that both gains converge very fast in the first few seconds, and are subsequently only finely tuned around the mean steady-state values.

Table 3 lists the P and I gains of the unfalsified controllers and the mean value of the P and I gains of the fuzzy controllers from 8 to 40 sec at the five set points. It is seen that for fuzzy controllers both P and I gains decrease along with the increase of the set points. This trend matches well with our experimental observation that the higher the set point, the lower the control effort. The values of P and I gains of unfalsified controller do not show such an obvious trend. This is partially due to the fact that both the candidate P and I gains are within the set of [0.1, 0.2, 0.3,..., 3.2] and hence the unfalsified control process cannot choose arbitrary values for P and I. Also, the construction of the performance specifications and the termination criterion of the implementation influence the final values of the gains.

4.2. Classification as adaptive control

Both approaches can be categorized as an adaptive control method. Fuzzy logic control can be classified as adaptive control, because its proportional and integral gains are tuned online at each sample period to improve the performance of the system. Although, unfalsified control design was implemented off-line in this research, it can be implemented online as an adaptive method. For this online implementation, a switch to the “best” unfalsified controller can occur at each sample period or at an integer multiple of the sample period.

4.3. Allowed design specifications

Multiple criteria can be proposed for unfalsified PI controller design. Then the optimal PI controller is selected to meet the multiple objectives specified by the designer. In contrast, fuzzy PI control cannot explicitly incorporate multiple criterion. The rule bases constructed for fuzzy PI control must implicitly represent the multiple control objectives.

4.4. Process knowledge

Both methods do not require an explicit plant model. However, experimental experience with the plant is required in the design process. A key to the unfalsified control design was the construction of the cost functions reflecting the performance specifications, which required experimental experience with the plant. In fuzzy logic control expert experience, i.e., the “rules of thumb” of how to achieve a good control, is the basis for constructing the rule bases. Experimental experience is also needed to select the range of proportional gain.

4.5. Computational requirements

For unfalsified control, the computational time is dependent on the size of the candidate set, the computational cost of the cost function and the efficiency of the search method. Currently, it is more computationally intensive than the fuzzy control.

Fuzzy logic control requires less online computational effort. At each sample period the proportional and integral gains are updated according to the reasoning of the proposed fuzzy rule bases. Since much effort was devoted to reducing the size of the rule bases in this research, fuzzy logic control was more computationally efficient than unfalsified control.

4.6. Controller development effort

In unfalsified control design significant development efforts are needed in the construction of cost functions that correctly reflect the underlying per-
formance specifications. These cost functions are key to guaranteeing good performance of the unfalsified controller.

Fuzzy controller design required less development effort. First, fuzzy reasoning rules for both the proportional gain and integral time constant were constructed based on the desired characteristics of the step responses. Meanwhile, only a simple tuning mechanism for the range of the proportional gain and the membership functions of the integral time constant was built.

4.7. Transient performance and motor saturation

Unfalsified control design benefited from its ability to explicitly handle multiple objectives. Hence it was possible to avoid motor saturation by falsifying the candidate controllers that were predicted to cause saturation. Also, good transient response was achieved since the best unfalsified controller was chosen based on the optimization of a carefully chosen cost function.

The fuzzy logic controllers also performed well. The control signal was generated online based on the error and change of error at each sample period. The fuzzy rules yielded good transient performance. As the online reasoning based on the error and change of error has the ability to correct the control effort promptly, the fuzzy system can adjust the control signal around the desired control effort, which can prevent the system from motor saturation. In our experiments, motor saturation never occurred in the process of fuzzy PI controller design.

Fig. 9 shows the comparison of step responses of an unfalsified PI controller and a fuzzy PI controller under set points of 1 and 5 V. Even though the step response curves are very similar, fuzzy PI control has a faster rise time.

The best feature of unfalsified PI control design is that it can explicitly incorporate multiple control objectives including motor saturation avoidance. However, fuzzy PI control has better performance in terms of computational requirements, controller development effort and transient performance. Both methods have the potential for improved performance. In this research the unfalsified control concept was used for off-line controller design. However, as previously mentioned, the method has the potential to be implemented on-line using a computationally efficient GA. For example, a GA with a real-valued representation can be studied as a means of implementing the unfalsified control online, since it has been reported that a GA using real-valued representation is an order of magnitude more efficient in terms of CPU time than a GA based on a binary representation [16].

In this research, fuzzy PI controllers were designed with the membership functions, scaling factors, and fuzzy reasoning rules tuned manually by trial and error. The controller developed in this manner can yield satisfactory performance but may not yield the best achievable performance. To save the time and cost of the tuning process, both genetic fuzzy or neural fuzzy systems can be applied for the tuning of parameters and rule bases of fuzzy logic controllers [21–24].
5. Conclusions

Two approaches for tuning PI controllers under high nonlinearity due to motor saturation, friction, and sensor quantization were described. The first method uses the unfalsified concept in which the control objectives can be clearly specified. The second method uses fuzzy logic to continuously tune the proportional and integral gains. The power of these approaches has been demonstrated through their applications to an industrial weigh belt feeder. The two approaches were compared to show their strengths and limitations.

Acknowledgment

This research was supported in part by the National Science Foundation under Grant CMS-9802197.

References


Yanan Zhao received the Ph.D. degree in Mechanical Engineering from the Florida State University in 2001. She is currently a Research Associate in the Department of Mechanical Engineering at the Florida A&M University—Florida State University College of Engineering. Her research interests include automated controller tuning, intelligent control systems for autonomous vehicles, fluidic thrust vector control, and evolutionary methods for control design.
Emmanuel G. Collins received the Ph.D. degree in Aeronautics and Astronautics from Purdue University in 1987. He worked for seven years in the Controls Technology Group at Harris Corporation, Melbourne, FL, before joining the Department of Mechanical Engineering at the Florida A&M University—Florida State University College of Engineering, Tallahassee, FL, where he currently serves as Professor and Director of the Center for Intelligent Systems, Control, and Robotics (CISCOR). Dr. Collins is the author of over 140 technical publications in controls and robotics.