Adaptive Fuzzy Control: Experiments and Comparative Analyses

Raúl Ordóñez, Jon Zumberge, Jeffrey T. Spooner, and Kevin M. Passino, *Senior Member, IEEE*

***Abstract*—Advances in nonlinear control theory have provided the mathematical foundations necessary to establish conditions for stability of several types of adaptive fuzzy controllers. However, very few, if any, of these techniques have been compared to conventional adaptive or nonadaptive nonlinear controllers or tested beyond simulation; therefore, many of them remain as purely theoretical developments whose practical value is difficult to ascertain. In this paper we will develop three case studies where we perform a comparative analysis between the adaptive fuzzy techniques in [1]–[3] and some conventional adaptive and nonadaptive nonlinear control techniques. In each case, the analysis will be performed both in simulation *and in implementation*, in order to show practical examples of how the performance of these controllers compares to conventional controllers in real systems.**

***Index Terms*—Adaptive fuzzy control, fuzzy control, intelligent control.**

I. Introduction

While nonadaptive fuzzy control[1] has proven its value in some applications, it is sometimes difficult to specify the rule base for some plants, or the need could arise to tune the rule-base parameters if the plant changes. This provides the motivation for adaptive fuzzy control, where the focus is on the automatic on-line synthesis and tuning of fuzzy controller parameters (i.e., the use of on-line data to continually "learn" the fuzzy controller, which will ensure that the performance objectives are met). The first adaptive fuzzy controller called the linguistic self-organizing controller (SOC) was introduced in [7]; several applications of this method have been studied (see the references in [8]). More recently, the "fuzzy model reference learning controller" (FMRLC) was introduced in [8]–[10], its extensions in [11], and both simulation [10]–[16] and implementation studies [17], [18][2] have shown this method to be quite successful. Many other adaptive fuzzy control techniques exist and the reader is referred to [9] and [10] for a more complete overview.

The problem with the SOC and FMRLC is that while they appear to be practical heuristic approaches to adaptive fuzzy control there is no proof that these methods will result in a stable closed-loop system (verification of stability is important, especially for safety-critical systems). Recently, however, several researchers have explored ideas from conventional adaptive and neural control to establish stability conditions for a variety of adaptive fuzzy control techniques [1]–[4], [19]–[23] and neural control methods [3], [24]–[29]. Generally, these techniques can be split into two categories: direct and indirect adaptive fuzzy control. In indirect adaptive fuzzy control, there is an identifier mechanism that produces a model of the plant which is then used to specify the controller (i.e., we update the controller parameters indirectly by first estimating the model parameters). In direct adaptive control, a model of the plant is not estimated; instead, we directly tune the controller parameters using plant data. Regardless of the method chosen or whose approach one takes, the practical value of these adaptive controllers is questionable since

1) there have been very few comparative analyses with conventional adaptive or nonadaptive nonlinear control methods;
2) there seem to be no experimental studies to determine how well these techniques perform in implementation, especially relative to conventional adaptive or nonadaptive nonlinear control techniques.

A complete assessment that would clarify how the above adaptive controllers would perform relative to all conventional methods and a wide variety of experimental settings is clearly beyond the scope of this or any single paper. Here, we use three case studies to compare the adaptive fuzzy controllers (both direct and indirect) in [1]–[3], to some of the more popular conventional linear and nonlinear methods.[3] The case studies we focus on are a rotational inverted pendulum, a process control experiment, and a ball-beam experiment. In the case of the pendulum, we provide a model of the dynamics of the plant, explain the experimental setup, develop conventional and adaptive fuzzy controllers, and provide both simulation and implementation results. For the other case studies, we simply give a brief description of the experiment and provide our results in the interest of brevity. The paper is organized as follows.

In Section II, we present the rotational inverted pendulum case study. After explaining the experimental setup and model, we develop a linear quadratic regulator (LQR), a (nonadaptive) feedback linearizing controller, and an adaptive feedback

Manuscript received November 29, 1995; revised August 2, 1996. This work was supported in part by the National Science Foundation, Grants EEC 9315257 and IRI 921332.

The authors are with the Department of Electrical Engineering, The Ohio State University, Columbus, OH 43210 USA.

Publisher Item Identifier S 1063-6706(97)00593-6.

[1] The authors assume that the reader has a good understanding of nonadaptive fuzzy control. For an introduction, see [4]–[6].

[2] The FMRLC has also been successfully implemented on a rotational inverted pendulum, a single-link flexible robot, and an induction machine. It has also been applied to the multiple-input multiple-output (MIMO) problem of a two-link rigid robot.

[3] The authors must assume that the reader has some familiarity with the techniques in [1]–[3]; these works provide all the necessary background details.

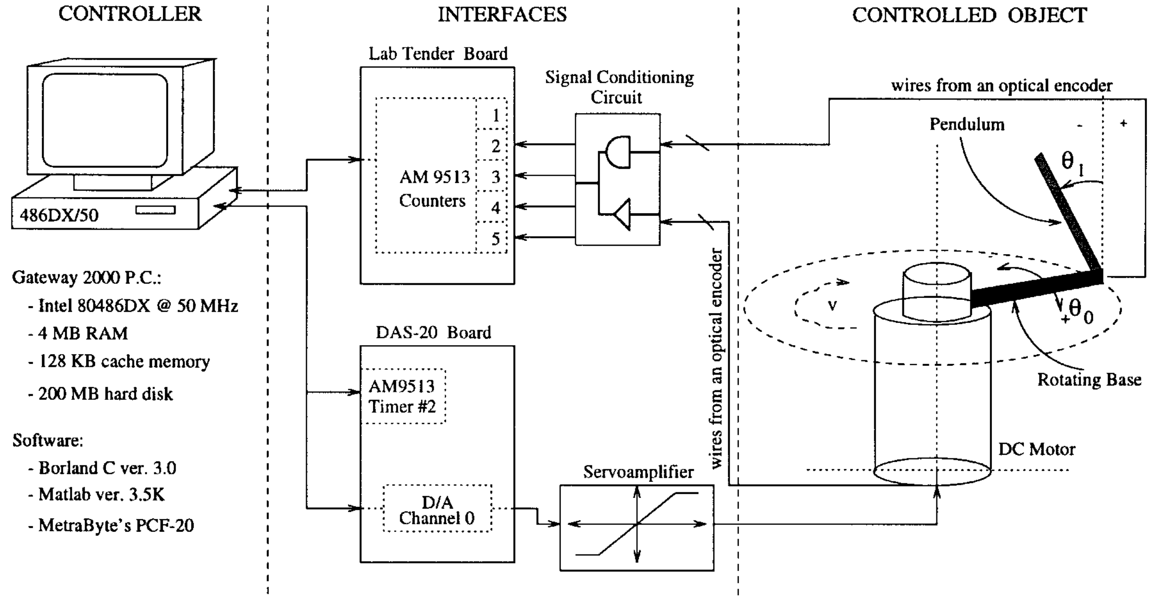


Fig. 1. Hardware setup of the inverted pendulum system.

linearizing controller (AFL). Next, we design two indirect adaptive fuzzy controllers, one that does not use *a priori* information about the plant, and one that does. Following this, we specify two direct adaptive fuzzy controllers, one that uses a feedback linearizing controller as the "known part" of the controller and another that uses an LQR for initialization. Simulation and implementation results are shown in all cases. Finally, we summarize and discuss the results and the performance of each controller.

In Section III, we present the process control case study. After providing the experimental setup and model we summarize our results using a feedback linearizing controller and an indirect adaptive fuzzy controller. In Section IV, we first describe the ball-beam experiment and its mathematical model. Then we develop a (nonadaptive) fuzzy controller and a direct adaptive fuzzy controller and compare their performance in simulation and implementation. In Section V, the concluding remarks, we summarize the overall results, provide a broad assessment of the apparent advantages and disadvantages of the adaptive fuzzy control techniques, and provide some future research directions which help to identify limitations of the scope and content of this paper. This paper is a significantly expanded version of the one in [30].

II. Rotational Inverted Pendulum

In our first case study for indirect and direct adaptive fuzzy control, we will focus on a rotational inverted pendulum test bed. Since adaptive control is being studied, special emphasis will be put on robustness by investigating the ability of the controllers to compensate for significant plant parameter variations.

A. Experiment Setup

The rotational inverted pendulum is an underactuated (i.e., it has fewer inputs than degrees of freedom), unstable system that presents considerable control-design challenges, and is, therefore, appropriate for testing the performance of different control techniques. The experimental setup used in this paper was developed in [31] and [32], where a nonlinear mathematical model of the system was obtained via identification techniques, and four different control methods were applied: proportional derivative control, linear quadratic regulation, direct fuzzy control, and autotuned fuzzy control.

The hardware setup of the system is shown in Fig. 1 (taken from [31]). It consists of three principal parts: the pendulum itself (controlled object), interface circuits, and the controller, implemented by means of a C program in a digital computer. The controller can actuate the pendulum by turning the dc motor to which it is attached. The motor has an optical encoder on its shaft that allows the measurement of its angle (with respect to the starting position) which we will refer to as θ_0. The shaft of the motor has a rotating base fixed to it. The pendulum can rotate freely around the base, and its angle, θ_1, can also be measured with an optical encoder with respect to the pendulum's stable equilibrium point, where it is assumed to have a value of π radians. The input voltage to the dc motor amplifier is constrained to a range of ±5 V.

As detailed in [31], the rotational pendulum system presents two somewhat separate problems: first, a controller needs to be designed that is able to balance the pendulum, and second, an adequate algorithm has to be used to swing up the pendulum so that when it reaches an upright position (i.e., where $\theta_1 \approx 0$) its angular velocity $(\dot{\theta}_1)$ is close to zero. This facilitates the job of the controller which "catches" the pendulum and tries to balance it. In this paper, we will not be concerned with swing-up details, and will concentrate only on the balancing control of the pendulum. The so-called "simple energy pumping" swing-up algorithm developed in [31][4] will be used without changes in all the experiments and

[4] Please consult this reference for details on the swing-up algorithm.

simulations, only with minor tunings depending on the nature of the test. This algorithm is just a proportional controller which takes as input the error between a maximum swing angle (the tuning parameter) and the base angle θ_0.

In implementation, a sampling time of 0.01 s was used. All the simulation and experimental plots include the swing-up phase and show the first 6 s only, since this time was considered enough to show the representative aspects of the results.

B. Modeling and Simulation

The rotational inverted pendulum can be represented with a four-state nonlinear model. The states are θ_0, $\dot{\theta}_0$, θ_1, and $\dot{\theta}_1$. Of them, only θ_0 and θ_1 are directly available for measurement; the other two states have to be estimated. To do this, we use a first-order backward difference approximation of the derivative. As tested in [31], this estimation method turns out to be very reliable and accurate. Thus, for the rest of the discussion, it will be assumed that all states are directly available for the controllers without need of further estimation.

The differential equations that describe the dynamics of the pendulum system (note that $\theta_1 = 0$ is the unstable equilibrium point) are given by

$$\ddot{\theta}_0 = -\, a_p\dot{\theta}_0 + K_p u \qquad (1)$$

$$\ddot{\theta}_1 = -\,\frac{C_1}{J_1}\,\dot{\theta}_1 + \frac{m_1 g l_1}{J_1}\,\sin\theta_1 + \frac{K_1}{J_1}\,\ddot{\theta}_0 \qquad (2)$$

where $m_1 = 8.6184 \times 10^{-2}$ Kg is the mass of the pendulum, $l_1 = 0.113$ m is the distance from the center of mass of the pendulum, $g = 9.8066\, m/s^2$ is the acceleration due to gravity, $J_1 = 1.301 \times 10^{-3}\, N - m - s^2$ is the inertia of the pendulum, $C_1 = 2.979 \times 10^{-3}\,(N - m - s)$/rad is the frictional constant between the pendulum and the rotating base, $K_1 = \pm 1.9 \times 10^{-3}$ is a proportionality constant, and u is the control input (voltage applied to the motor). A first-order model of the dc motor is given by $\Theta_0(s)/U(s) = K_p/s{+}a_p$ with $a_p = 33.04$ and $K_p = 74.89$. The numerical values of the constants were determined experimentally in [31]. Note that the sign of K_1 depends on whether the pendulum is in the inverted or the noninverted position, i.e., for $\pi/2 < \theta_1 < 3\pi/2$ we have $K_1 = 1.9 \times 10^{-3}$, and $K_1 = -1.9 \times 10^{-3}$ otherwise (recall that $\theta_1 = \pi$ is the stable equilibrium point). When simulating the system, a conditional statement is used to determine the sign of K_1 according to the relation above.

Let $x_1 = \theta_0$, $x_2 = \dot{\theta}_0$, $x_3 = \theta_1$, and $x_4 = \dot{\theta}_1$. Then a state variable representation of the plant is given by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= a_1 x_2 + b_1 u \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= a_2 x_2 + a_3 \sin x_3 + a_4 x_4 + b_2 u \end{aligned} \qquad (3)$$

where $a_1 = -a_p$, $a_2 = -(K_1 a_p/J_1)$, $a_3 = m_1 g l_1/J_1$, $a_4 = -(C_1/J_1)$, $b_1 = K_p$, and $b_2 = K_1 K_p/J_1$. Since we are only interested in balancing the pendulum, we take the output of the system as $y = x_3$.

For simulation of the system, a fourth-order Runge–Kutta numerical method was used in all cases, with an integration step size of 0.001 s. The controllers are assumed to be continuous; therefore, the sampling time of the controller was set equal to the integration step size. Also, the initial conditions were kept identical in all simulations; these are $x_1(0) = 0$ rad, $x_2(0) = 0$ rad$/s$, $x_3(0) = \pi$ rad, and $x_4(0) = 0\,$rad$/s$. Under these conditions, the pendulum is in the downward position. When the simulations start, the pendulum is first swung up with the same swing-up algorithm used for implementation, and then "caught" by the balancing controller currently being tested to resemble experimental conditions as accurately as possible. The balancing controller begins to act when $|x_3| \leq 0.3\,$rad; at the same time, the swing-up controller is shut down.

For the swing-up algorithm, we have one design parameter, the angle Γ, with values typically between 1.1 and 1.4 radians; this angle determines the maximum amplitude of each swing. The gain K_{sw} is fixed at 0.75 for all tests. Then, taking V as the swing-up control input, the algorithm works as follows:

```
If $\theta_0 - \pi < 0$ then $\theta_{0\mathrm{ref}} = -\Gamma$
   else $\theta_{0\mathrm{ref}} = \Gamma$
      $e_{sw} = \theta_{0\mathrm{ref}} - \theta_0, \quad V = K_{sw} e_{sw}.$
```

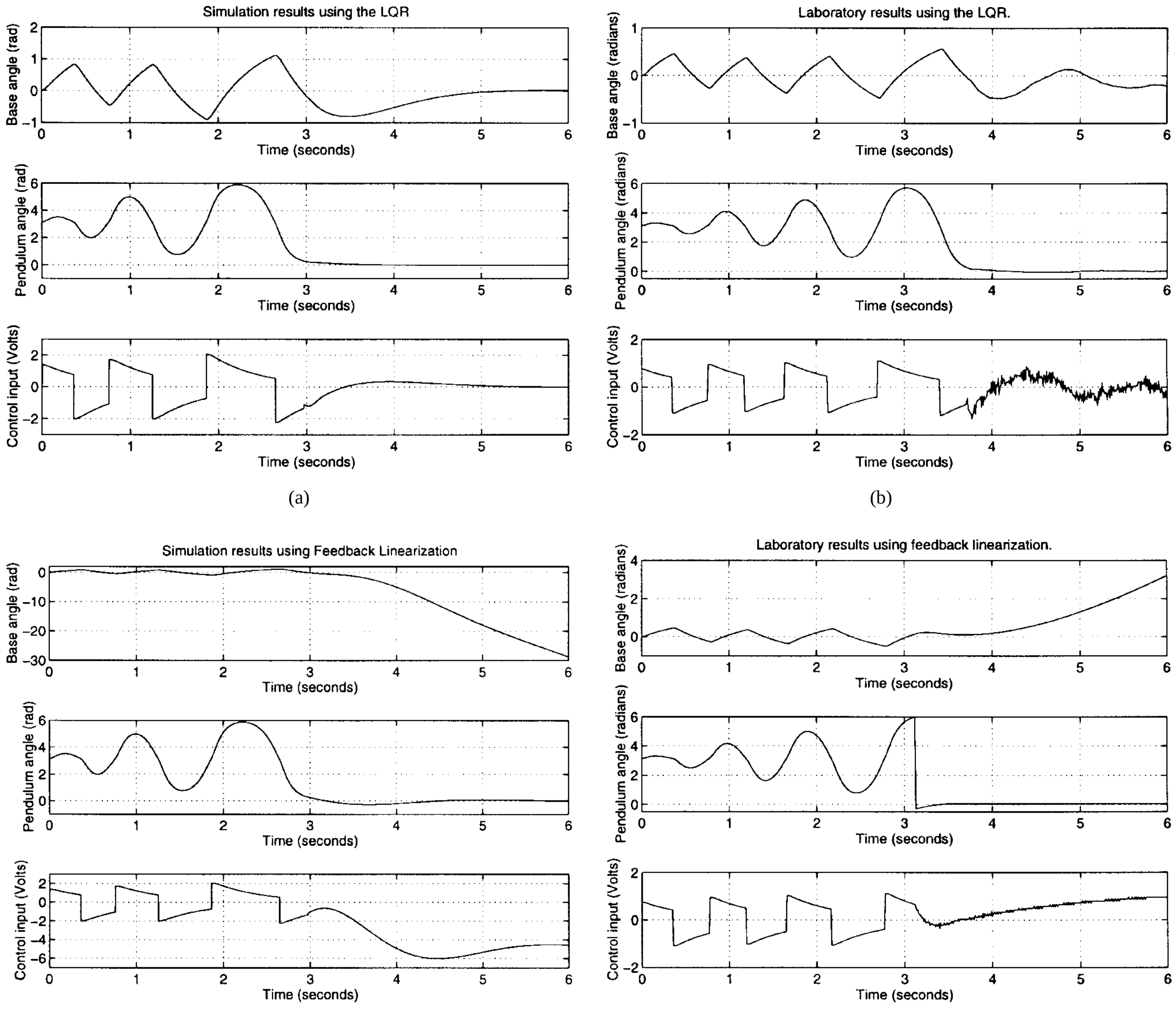
C. Two Nonadaptive Controllers

In this section, two nonadaptive controllers for the inverted pendulum will be introduced, and these will serve as a baseline for comparison to the results to follow. First, a linear quadratic regulator from [31] (which provided the best experimental results for nominal conditions) will be used. Second, a feedback linearizing control law will be used; as shown later, there is no guarantee of boundedness using this technique, and the results obtained here corroborate this theoretical prediction.

1) Linear Quadratic Regulator: To design an LQR for the pendulum, the approach taken in [31] was to linearize the system model by using the approximation $\sin x_3 \approx x_3$, which is valid for small angles, in (3). The resulting system can be shown to be controllable; thus, an LQR can be constructed. For the design, the greatest penalty was assigned to the error in states x_3 and x_4, since the primary objective of the controller is to balance the pendulum, and not to keep the base from moving. The state-feedback gain obtained was tested experimentally and, after some fine tuning, the gain vector used was $K = [-0.7,\, 1,\, 10.8,\, 0.7]^\top$, where the state error $\mathbf{e} = \mathbf{x_r} - \mathbf{x}$ ($\mathbf{x_r}$ is a reference-state trajectory, typically set identically equal to zero) is used, and $\mathbf{x} = [x_1,\, x_2,\, x_3,\, x_4]^\top$.

As shown in Fig. 2(a) and (b), the LQR performs very well in both simulation and implementation; it successfully balances the pendulum and drives all the system states to zero. The control input ideally goes to zero when equilibrium is reached [Fig. 2(a)], although in practice it does not [Fig. 2(b)], since the unmodeled aspects of the system (e.g., sensor noise, sampling time, nonlinear characteristics, etc.) prevent the controller from behaving perfectly. The results obtained here closely match those in [31].

2) Feedback Linearizing Controller: We find that the inverted pendulum has a *strong relative degree* [33], [34] of two because after differentiating the output twice, we obtain $\ddot{y} = a_2\, x_2 {+} a_3\, \sin x_3 {+} a_4\, x_4 {+} b_2 u$. Then, as described in [35]



(a)

(b)

(c)

(d)

Fig. 2. (a) LQR simulation. (b) LQR implementation. (c) Feedback linearization simulation. (d) Feedback linearization implementation.

and [36], we can define a feedback linearizing control law as

$$u_f = \frac{1}{b_2}\left[-(a_2 x_2 + a_3 \sin x_3 + a_4 x_4) + \nu\right] \qquad (4)$$

where ν is a signal that we will define later. By applying this control law, the plant state space is mapped to another, linear space. Since the plant has a strong relative degree, it is possible to find this mapping $\mathbf{T}$ such that $\mathbf{T}(\mathbf{x})$ is a diffeomorphism. Then, $\mathbf{z} = \mathbf{T}(\mathbf{x})$ is a new linear state variable representation of the system if we use the control law (4). The system has two unobservable states that form the so-called *internal dynamics* [33] of the system. Such a mapping $\mathbf{T}(\mathbf{x})$ can be found to be $z_{11} = T_1(\mathbf{x}) = x_3$, $z_{12} = T_2(\mathbf{x}) = x_4$, $z_{21} = T_3(\mathbf{x}) = x_1 + x_3$, and $z_{22} = T_4(\mathbf{x}) = x_2 - (b_1/b_2)x_4$. Given this new set of states, the output of the plant is given by $y = z_{11}$.

If we restrict the output y to be identically zero for all time, and given that $\mathbf{x} = 0$ is an (unstable) equilibrium point of the undriven system (see [35] and [36]), then the *zero dynamics* of the pendulum are given by $\dot{z}_{21} = z_{22}$ and $\dot{z}_{22} = (a_1 - a_2 b_1/b_2)z_{22}$. For the system to be minimum phase, its zero dynamics have to be *asymptotically stable* [33], [34], i.e., $a_1 - a_2 b_1/b_2 > 0$ for this choice of $\mathbf{T}(\mathbf{x})$. However, a simple computation shows that this is not the case; indeed, the zero dynamics of the pendulum are only *marginally stable* because $a_1 - a_2 b_1/b_2 = 0$. This causes two states of the system (x_1 and x_2, since x_3 and x_4 are bounded when the output is bounded because we assume the state $\mathbf{x}$ to be uniformly continuous) to be potentially unbounded under feedback linearization. As will be seen later, this prediction is corroborated, both in simulation and experimentation.

The marginal stability of the zero dynamics implies that although the control law (4) will yield a stable input–output behavior under the right choice of ν. If the initial conditions of the system are distinct from zero, then a subset of the states will be unbounded; in particular, by solving the zero dynamics differential equations for some nonzero initial conditions, we find that the state x_1 (which represents the base angle θ_0 of the rotational pendulum) will be given by $x_1(t) = ct$ in steady state, where t represents time and c is an integration constant.

It is worth noting that the unboundedness of x_1 can be tolerated in this experiment, since all it means is that the pendulum base keeps rotating while the pendulum is being balanced. Of course, in practice, a limit is imposed on the rate and amount of this rotation to protect the machinery, but in principle, the marginal stability of the zero dynamics can effectively be dealt with.

To simulate and implement the feedback linearizing closed loop, it is necessary to specify the signal ν. Let $y_m(t)$ be a prespecified reference trajectory (set equal to zero for all cases in this work), at least twice differentiable, and take $e_o = y - y_m$ (following the notation in [35] and [36]). Then, let $\nu = \ddot{y}_m - 2\dot{e}_o - 8e_o$ with which we obtain a stable (in the input–output sense) closed-loop system with poles at $s = -1 \pm 2.65j$. This choice of the closed-loop poles was made because of practical considerations: experience with feedback linearization on the pendulum experiment indicates that attempting to bring the output error to zero too fast can sometimes result in failure or a degraded performance; therefore, these somewhat slow poles were chosen.

The simulation results in Fig. 2(c) are as expected—the pendulum is balanced, and the base keeps rotating at an almost constant velocity. Note that this controller settles the control input at a nonzero value, which in turn introduces energy into the system and causes its base to rotate; this is easily explained by the theoretical analysis of the zero dynamics. A comparison with the experimental results in Fig. 2(d) shows significant similarities: after the swing-up phase, the states x_1 and x_3, as well as the control input, behave as in the simulation. Observe that in simulation the feedback linearizing controller reaches a peak value of -6 V to balance the pendulum; although in implementation the control input is limited to ± 5 V as explained above, no such bound was used for simulation since we desired to preserve ideal circumstances regarding control action.

D. Adaptive Feedback Linearization

Both the feedback linearizing controller and the LQR, as well as almost any other nonadaptive technique used on the pendulum fail (or in the case of direct fuzzy control have degraded performance [31]) when the nominal system (i.e., the pendulum without any mass changes or added disturbances) is altered in an *unknown way*. It is in such a situation that adaptive control plays a central role since it is, at least in principle, able to deal with significant plant changes. In this investigation, two types of plant alterations were used—a container half filled with metal bolts fixed at the tip of the pendulum, and a container half filled with water fixed at the tip of the pendulum.

The added weight (not accounted for in the design of the controllers) not only shifts the pendulum's center of mass away from the pivot point (which, in turn, decreases the natural frequency of the pendulum) and makes the effects of friction less dominant, but also introduces random disturbances that vary in nature with the bolts and the water. In the case of the water container, a "sloshing liquid" effect is created, which strongly affects the dynamics of the system. As will be seen from the results below, the bolts have a different type of effect that is caused by their "rattling" during balancing.

We note that the adaptive controllers in this subsection and the remainder of this section are based on the assumption that the plant is minimum-phase (see [3], [36]). Since this assumption does not hold for the pendulum, some consequences of the techniques are no longer guaranteed; specifically, we should not expect the state x_1 (and possibly the control input) to be bounded, because its behavior will depend on the initial conditions of the system. However, the study of the adaptive techniques on such a system is of theoretical and practical relevance because, in the first place, it provides a good example of practical results very well predicted by theoretical analysis of nonlinear systems, and in the second place, it can give insight into how to overcome the limitations of the adaptive controllers inherited by their underlying assumptions. As will be shown, it is indeed possible to obtain not only stability and boundedness, but also good robustness to unmodeled plant changes.

The adaptive fuzzy techniques illustrated in this paper are based on feedback linearization; therefore, adaptive feedback linearization (AFL) seemed the most natural choice for reference and comparison to conventional adaptive control. For the design of this controller, the technique described in [35] and [36] was used as we discuss next.

We first rewrite (3) as a linear combination of known, fixed, nonlinear functions

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}) + g(\mathbf{x})\,u \\ y &= h(\mathbf{x}) \end{aligned} \tag{5}$$

where

$$\begin{aligned} f(\mathbf{x}) &= \theta_1^{(1)^*}\begin{bmatrix} x_2 \\ 0 \\ x_4 \\ 0 \end{bmatrix} + \theta_2^{(1)^*}\begin{bmatrix} 0 \\ x_2 \\ 0 \\ 0 \end{bmatrix} + \theta_3^{(1)^*}\begin{bmatrix} 0 \\ 0 \\ 0 \\ x_2 \end{bmatrix} \\ &\quad + \theta_4^{(1)^*}\begin{bmatrix} 0 \\ 0 \\ 0 \\ \sin x_3 \end{bmatrix} + \theta_5^{(1)^*}\begin{bmatrix} 0 \\ 0 \\ 0 \\ x_4 \end{bmatrix} \\ g(\mathbf{x}) &= \theta_1^{(2)^*}\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \theta_2^{(2)^*}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \\ h(\mathbf{x}) &= x_3. \end{aligned} \tag{6}$$

We shall *estimate* $f(\mathbf{x})$ and $g(\mathbf{x})$ by searching for the optimum vectors $\theta^{(1)^*} = [\theta_1^{(1)^*}, \cdots, \theta_5^{(1)^*}]^\top$ and $\theta^{(2)^*} = [\theta_1^{(2)^*}, \theta_2^{(2)^*}]$. We use $\theta^{(1)}(t)$ and $\theta^{(2)}(t)$ to denote the estimates of the optimum parameter vectors at time t. Then, the adaptive control law is given by $u_{af} = 1/(L_g L_f h)_e[-(L_{f^2}\, h)_e + \nu]$ where $(L_f h)_e$ stands for the estimated Lie derivative of h with respect to f, as defined in [34] and [36], and the variable $\mathbf{x}$ has been dropped for convenience. To allow for tracking, we take $\nu = \ddot{y}_m + \alpha_2(\dot{y}_m - \dot{y}) + \alpha_1(y_m - y)$, with $\alpha_1 = 8$ and $\alpha_2 = 2$ to have the same poles as in the nonadaptive feedback linearization. Notice that $\dot{y}$ does not need to be estimated, since $\dot{y} = \dot{x}_3 = x_4$ has already been approximated by a backward difference, as explained above.

Following [35], define $\Theta \in \Re^{32}$ as a vector containing all the combinations $\theta_i^{(1)}(t)$, $\theta_i^{(2)}(t)$, $\theta_i^{(1)}(t)\theta_j^{(1)}(t)$, and $\theta_i^{(1)}(t)\theta_j^{(2)}(t)$. For adaptation, define an error signal of the form $e_1 = \beta_2 \dot{e}_o + \beta_1 e_o$ with $e_o = y - y_m$, where the transfer function $(\beta_2 s + \beta_1)/(s^2 + \alpha_2 s + \alpha_1)$ is strictly positive real [33]. The adaptation law given by a normalized gradient approach is

$$\dot{\Theta} = -\frac{e_1\, w}{1 + w^\top w} \tag{7}$$

where w is the *regressor* vector obtained by computing the output error equation $\ddot{e}_o + \alpha_2\, \dot{e}_o + \alpha_1\, e_o$ (see [36]). To start the search for the optimum vectors $\theta^{(1)^*}$ and $\theta^{(2)^*}$ at the best known point in the search space, their estimates $\theta^{(1)}(t)$ and $\theta^{(2)}(t)$ were initialized using the parameters obtained from the system model, $\theta^{(1)}(0) = [1, a_1, a_2, a_3, a_4]^\top$, and $\theta^{(2)}(0) = [b_1, b_2]^\top$. For simulation and implementation we used $\beta_1 = 0.08$ and $\beta_2 = 0.1$. These values for β_1 and β_2 were determined via manual tuning; after several simulation and experimentation trials, they were the choices that, as far as we could determine, made the controller work at its best and still maintain the strictly positive real condition.

Fig. 3 contains the results obtained with AFL. Fig. 3(a) has characteristics similar to those of the nonadaptive feedback linearization in Fig. 2(a), although the required control input stays within implementation bounds. For the nominal plant, the controller exhibits very good behavior, as seen in Fig. 3(b): the pendulum is perfectly balanced and the control input settles at a value close to zero so the base rotates slowly. In Fig. 3(c) we observe that the controller manages to balance the pendulum with bolts, but only for a short time; the base is turning rapidly and, at about the fifth second, the control input reaches the its limit of 5 V. The controller had the greatest problems with water and was not able to maintain equilibrium, as seen in Fig. 3(d). In the next section, we will show how the adaptive fuzzy control techniques can significantly outperform this conventional adaptive controller.

E. Indirect Adaptive Fuzzy Control

Here, an indirect adaptive fuzzy controller (IAFC) will be developed for the inverted pendulum; two possible configurations will be presented and used for experiments. First, a controller that does not make an explicit use of the known plant dynamics to estimate the “certainty equivalence control term” [1], [3] will be used on the pendulum. Second, it will be illustrated how to incorporate the knowledge of the model (3) in the design. It will be shown experimentally that such an enhanced controller has, in the case of the pendulum, a noticeable advantage over the previous techniques, and provides an increased robustness against the induced disturbances.

For what follows in this section and the next, the notation from [1]–[3] will be used. Moreover, in the interest of being brief, we do not repeat the theoretical development of the controllers in [1]–[3], and simply provide a complete description of each controller. To fully understand why the controllers in this and the next section result in stable operation, the reader should consult [1]–[3].

1) Design without Use of Plant Dynamics Knowledge: As previously shown, the pendulum model has a relative degree of two. The input–output differential equation of the pendulum model can thus be rewritten as

$$\ddot{y} = [\alpha_k(t) + \alpha(\mathbf{x})] + [\beta_k(t) + \beta(\mathbf{x})]u \tag{8}$$

where, for now, we take $\alpha_k(t) \equiv 0$ and $\beta_k(t) \equiv 0$ (α_k and β_k are known, measurable parts of the dynamics [3]), and substituting the numerical values of the parameters we obtain

$$\alpha(\mathbf{x}) \approx 48.2521\, x_2 + 73.4085\, \sin x_3 - 2.2898\, x_4 \tag{9}$$

$$\beta(\mathbf{x}) \approx -109.3705. \tag{10}$$

In these equations, we use “approximately equal” signs because the numerical parameters of the equations are not expected to represent the pendulum’s input–output dynamics *exactly*; rather, the right-hand side of (9) and (10) are simply our *best known approximations* to $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$, respectively. Note that $\beta(\mathbf{x}) < 0$, so there exists a $\beta_0 < 0$ (take, for instance, $\beta_0 = -100$, which gives us a safe margin of error) such that $\beta(\mathbf{x}) \leq \beta_0$ for all $t \geq 0$; thus, $\beta(\mathbf{x})$ is bounded away from zero, a condition we will need to ensure stability.

It is possible to represent (9) and (10) using a special form of Takagi–Sugeno fuzzy systems [37]. To briefly present the notation, take a fuzzy system denoted by $\tilde{f}(\mathbf{x})$. Then, $\tilde{f}(\mathbf{x}) = \sum_{i=1}^{p} c_i\, \mu_i / \sum_{i=1}^{p} \mu_i$. Here, singleton fuzzification of the input $\mathbf{x} = [x_1, \cdots, x_n]^\top$ is assumed; the fuzzy system has p rules, and μ_i is the value of the membership function for the antecedent of the ith rule given the input $\mathbf{x}$. It is assumed that the fuzzy system is constructed in such a way that $\sum_{i=1}^{p} \mu_i \neq 0$ for all $\mathbf{x} \in \Re^n$. The parameter c_i is the consequent of the ith rule which, in this paper, will be taken as a linear combination of Lipschitz continuous functions $z_k(\mathbf{x}) \in \Re$, $k = 1, \cdots, m-1$, so that $c_i = a_{i,0} + a_{i,1}\, z_1(\mathbf{x}) + \cdots + a_{i,m-1}\, z_{m-1}(\mathbf{x})$, $i = 1, \cdots, p$. Define

$$z = \begin{bmatrix} 1 \\ z_1(\mathbf{x}) \\ \vdots \\ z_{m-1}(\mathbf{x}) \end{bmatrix} \in \Re^m$$

$$\zeta^\top = \frac{[\mu_1 \cdots \mu_p]}{\sum_{i=1}^{p} \mu_i}$$

$$A^\top = \begin{bmatrix} a_{1,0} & a_{1,1} & \cdots & a_{1,m-1} \\ a_{2,0} & a_{2,1} & \cdots & a_{2,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p,0} & a_{p,1} & \cdots & a_{p,m-1} \end{bmatrix}.$$

Then, the nonlinear equation that describes the fuzzy system can be written as $\tilde{f}(\mathbf{x}) = z^\top A \zeta$ (notice that standard fuzzy systems may be treated as special cases of this more general representation [3]).

Given this notation, we can write

$$\alpha(\mathbf{x}) = z_\alpha^\top A_\alpha^* \zeta_\alpha + d_\alpha(\mathbf{x}) \tag{11}$$

$$\beta(\mathbf{x}) = z_\beta^\top A_\beta^* \zeta_\beta + d_\beta(\mathbf{x}) \tag{12}$$

where $d_\alpha(\mathbf{x})$, $d_\beta(\mathbf{x})$ are the approximation errors that arise

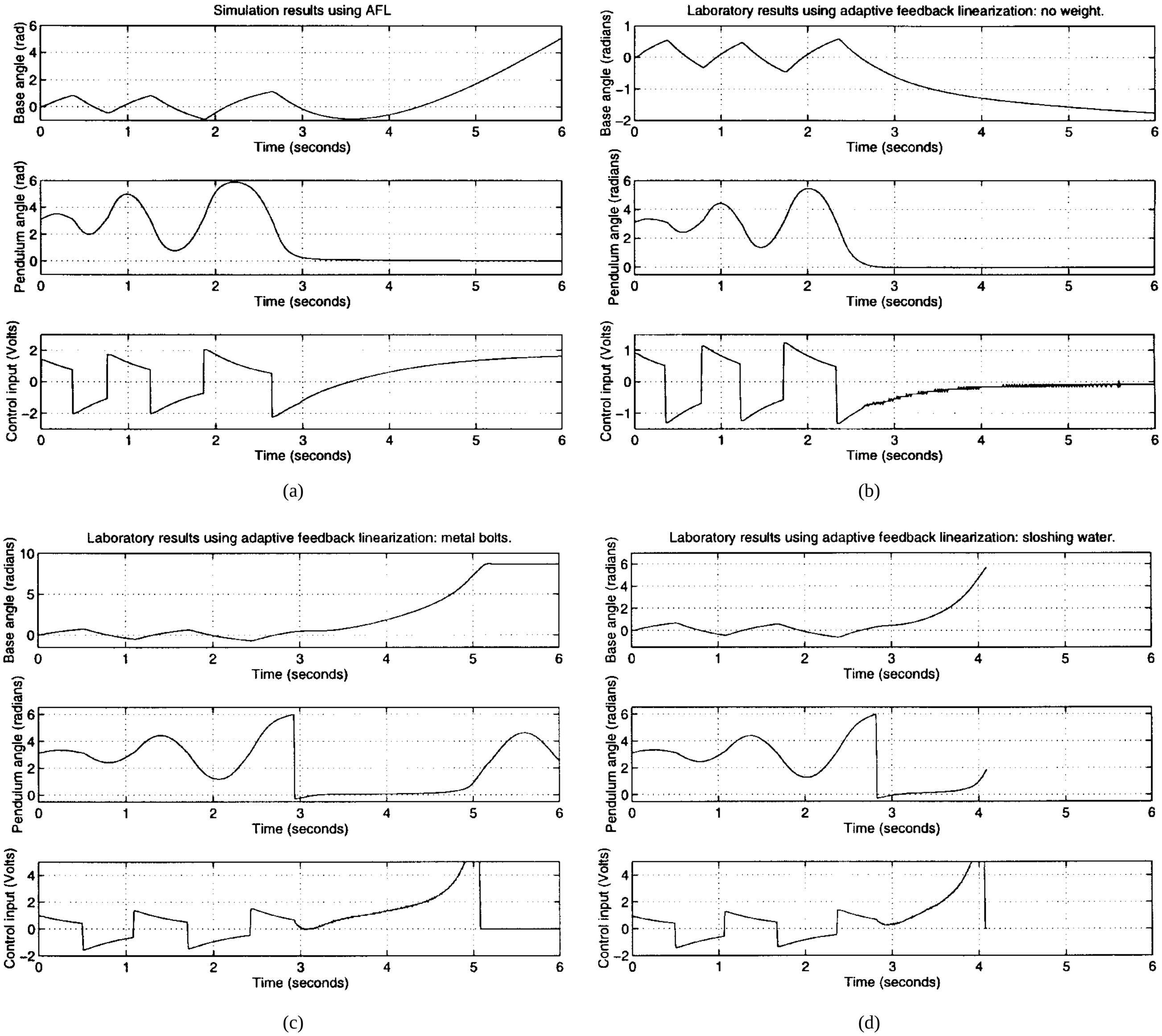


Fig. 3. (a) AFL simulation. (b) Experimental results of AFL with nominal plant. (c) Experimental results of AFL with disturbance: metal bolts. (d) Experimental results of AFL with disturbance: sloshing water.

when $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ are represented with finite fuzzy systems, and A_α^*, A_β^* represent the optimum and unknown fuzzy system parameters that minimize the approximation errors. We assume that $D_\alpha(\mathbf{x}) \geq |d_\alpha(\mathbf{x})|$ and $D_\beta(\mathbf{x}) \geq |d_\beta(\mathbf{x})|$, where $D_\alpha(\mathbf{x})$ and $D_\beta(\mathbf{x})$ are known bounds of the approximation errors between the actual system and its fuzzy system representation. In simulation, these functions were taken as $D_\alpha(\mathbf{x}) = 0.5$ and $D_\beta(\mathbf{x}) = 1.1$, since the optimum representation error was expected to be small. In the laboratory, however, it was necessary to increase these bounds to $D_\alpha(\mathbf{x}) = 5$ and $D_\beta(\mathbf{x}) = 8$, because apparently the complexities of the real plant were much harder to represent than the model. Note that these values were chosen as the result of a tuning process, where rough intuitive estimates of the values were used to start with and then tuned to improve the performance of the controller. The functional effect of increasing the error bounds $D_\alpha(\mathbf{x})$ and $D_\beta(\mathbf{x})$ is to increase the magnitude of the sliding mode control term (see below).

We approximate the unknown fuzzy system representation using

$$\hat{\alpha}(\mathbf{x}) = z^\top A_\alpha(t)\zeta_\alpha \tag{13}$$

$$\hat{\beta}(\mathbf{x}) = z^\top A_\beta(t)\zeta_\beta \tag{14}$$

where the matrices $A_\alpha(t)$ and $A_\beta(t)$ will be adaptively updated, as shown below in (18). Based on the general form of the system model (3), we take the following set of equations for both $\hat{\alpha}(\mathbf{x})$ and $\hat{\beta}(\mathbf{x})$ (also used for the direct adaptive fuzzy controller in the next section): $z = [1, x_1, x_2, \sin x_3, x_4]^\top$. The fuzzy systems use five rules each of the form

$$\textbf{If } x_3 \text{ is } F_i \textbf{ then } c_i = f_i(z), \qquad i = 1, \cdots, 5 \tag{15}$$

where each $f_i(z)$ is, respectively, a row of the matrices

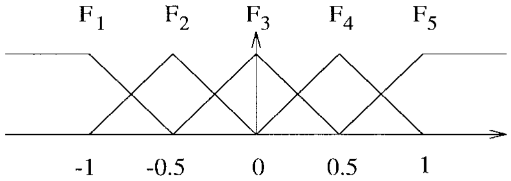


Fig. 4. Input membership functions.

$z^\top A_\alpha(t)$ and $z^\top A_\beta(t)$, and we initialize the system with

$$A_\alpha(0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ a_2 & a_2 & a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 & a_3 & a_3 \\ a_4 & a_4 & a_4 & a_4 & a_4 \end{bmatrix}$$
$$A_\beta(0) = \begin{bmatrix} b_2 & b_2 & b_2 & b_2 & b_2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{16}$$

Note that this fuzzy system design is overspecified, mainly in the case of $\hat{\beta}(\mathbf{x})$ because, from the system model, this function is not expected to depend on the state vector $\mathbf{x}$. However, this choice was made to allow for a greater adaptation flexibility. The initialization (16) gives the system the best-known starting point in the search space. The input fuzzy sets F_i are as described in Fig. 4 (the normalizing gain for the input to the fuzzy system x_3 was set to one for simplicity).

Define the signals $e_s = \kappa_0 \, e_o + \dot{e}_o$ and $\nu_a(t) = \ddot{y}_m + \eta e_s + \kappa_0 \, \dot{e}_o$, where now we take $e_o = y_m - y$ and we let $\eta = 1$, $\kappa_0 = 8$ (with these choices, the poles of the error transfer function are at $s = -1$ and $s = -9$, which produce a small error settling time). Then, the indirect adaptive control law [1], [3] is given by

$$u_i = u_{ce} + u_{si} + u_{bi} \tag{17}$$

where the *certainty equivalence control term* is taken as $u_{ce} = [1/\hat{\beta}(\mathbf{x})][-\hat{\alpha}(\mathbf{x}) + \nu_a(t)]$. The *sliding-mode control term* is given by $u_{si} = (1/\beta_0)[D_\alpha(\mathbf{x}) + D_\beta(\mathbf{x})|u_{ce}|]\mathrm{sgn}\,(e_s)$. To define the *bounding control term* u_{bi}, we first need to determine bounding functions $\alpha_1(\mathbf{x})$ and $\beta_1(\mathbf{x})$ for $|\alpha(\mathbf{x})|$ and $|\beta(\mathbf{x})|$, respectively. Based on the numerical values of (9) and (10), the bounding functions were empirically determined to be $\alpha_1(\mathbf{x}) = 70\,x_2 + 75\,x_3 + 10\,x_4$, $\beta_1(\mathbf{x}) = 140$. Then, let $u_{bi} = \{(1/\beta_0)(|\hat{\alpha}(\mathbf{x})| + \alpha_1(\mathbf{x}) + [|\hat{\beta}(\mathbf{x})| + \beta_1(\mathbf{x})]\,|u_{ce}|) + |u_{si}|\}\,\mathrm{sgn}\,(e_s)$ whenever $|e_s| > M_e$, and $u_{bi} = 0$ otherwise. The parameter M_e defines a bounded, closed subset of the e_s error-state space within which the error is guaranteed to stay. For simulation, we took $M_e = 0.4$; again, a larger margin had to be used in implementation, and the smallest acceptable value was $M_e = 3$. Note that although it is possible, in principle, to take an arbitrarily small M_e, in practice it is often the case that the bounding control acts "too much" with a small M_e, and the unavoidable limits in the control-input signal cause the system to become unstable. Also note that from the stability analysis in [1] and [3], the bounding control term is not required for stability and may, thus, beset to zero if it provides no advantage in a particular design. However, it gives the designer the flexibility of choosing a hard bound for the tracking error, which in the case of the pendulum is a useful feature.

To define the adaptation equations, let I_5 be a 5×5 identity matrix, and let $Q = 0.05\,I_5$ for simulation, and $Q = 0.1\,I_5$ for implementation, and take

$$\begin{aligned} \dot{A}_\alpha(t) &= -Q^{-1}\,z\,\zeta_\alpha^\top\,e_s \\ \dot{A}_\beta(t) &= -Q^{-1}\,z\,\zeta_\beta^\top\,e_s\,u_{ce}. \end{aligned} \tag{18}$$

A projection algorithm is used to ensure that $A_\alpha(t)$ and $A_\beta(t)$ remain within reasonable limits; specifically, it is sufficient to ensure that $\hat{\beta}(\mathbf{x})$ is bounded away from zero with $\hat{\beta}(\mathbf{x}) \leq \beta_0 < 0$.

Note that the IAFC adaptation algorithm guarantees that the parameter error matrices $\Phi_\alpha(t) = A_\alpha(t) - A_\alpha^*$ and $\Phi_\beta(t) = A_\beta(t) - A_\beta^*$ will at least stay bounded. Note also that (9) and (10) are themselves only approximations, based on our best knowledge of the plant. Thus, it is possible that the fuzzy system representation of the input–output equation $\hat{\alpha}(\mathbf{x})$ and $\hat{\beta}(\mathbf{x})$ does not converge to (9) and (10), but perhaps to a better (or worse) model of the system.

In this way, the IAFC from [1] and [3] is completely specified. One of its assumptions is not satisfied, namely, that the zero-dynamics of the plant are exponentially attractive; thus, as happened with the adaptive and nonadaptive feedback linearizing controllers, boundedness of the state x_1, and possibly of the control input u_i, is not expected. However, in principle, the controller should be able to achieve output convergence (i.e., keep the pendulum balanced).

We see in Fig. 5(a) that this is indeed the case. The pendulum is successfully balanced, and the settling time of the controller is smaller than in the case of any of the previous controllers since here $\nu_a(t) = \ddot{e}_o + 9\dot{e}_o + 8e_o$, which has poles at $s = -1$ and $s = -9$; the reason why this signal was not used for the (adaptive and nonadaptive) feedback linearizing controllers is that with it they performed worse, both in simulation and experimentation, in terms of error convergence and robustness.

For implementation, we see in Fig. 5(b) that the pendulum is balanced using the nominal plant, although the output error is not exactly zero. When the bolts disturbance is used [Fig. 5(c)], the controller has trouble similar to AFL [see Fig. 3(c)] because the control input reaches its lower limit of -5 V. We see in Fig. 5(d) that with sloshing water, the controller performs better, although it is apparent that the control input limit is about to be reached. Thus, the performance of this IAFC design is roughly similar to that of adaptive feedback linearization.

2) Incorporation of Plant Dynamics Knowledge in Design: To improve the robustness characteristics of the IAFC, we will now take a slightly different design approach and will make explicit and direct use of the knowledge we have of the plant, i.e., the nonlinear model (3). By comparing the simulation and experimental results so far, we see that although very useful for theoretical analysis and design, the model is nevertheless a relatively poor approximation of the rotational

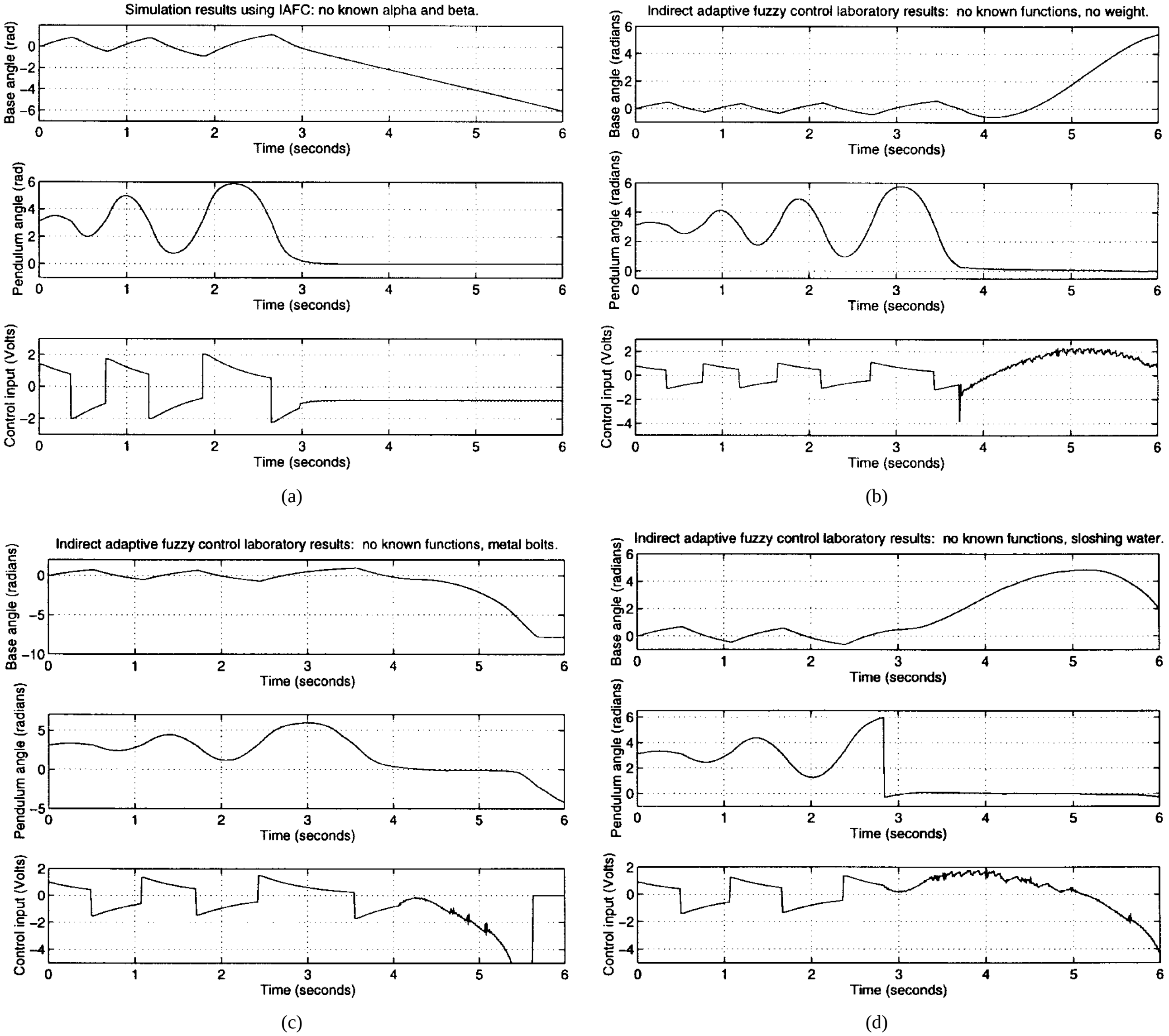


Fig. 5. No plant dynamics knowledge used: (a) IAFC simulation. (b) Experimental results of IAFC with nominal plant. (c) Experimental results of IAFC with disturbance—metal bolts. (d) Experimental results of IAFC with disturbance—sloshing water.

inverted pendulum. In spite of this fact, it can effectively be incorporated into the indirect adaptive scheme, and thus provide it with an improved disturbance rejection ability.

We are now assuming that the pendulum input–output equation can be represented by some known nonzero fixed functions $\alpha_k(t)$ and $\beta_k(t)$, and unknown functions $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$, which are to be identified on-line by the IAFC adaptation mechanism. Since the model (3) represents all our knowledge about the pendulum, we use it to specify the known functions as

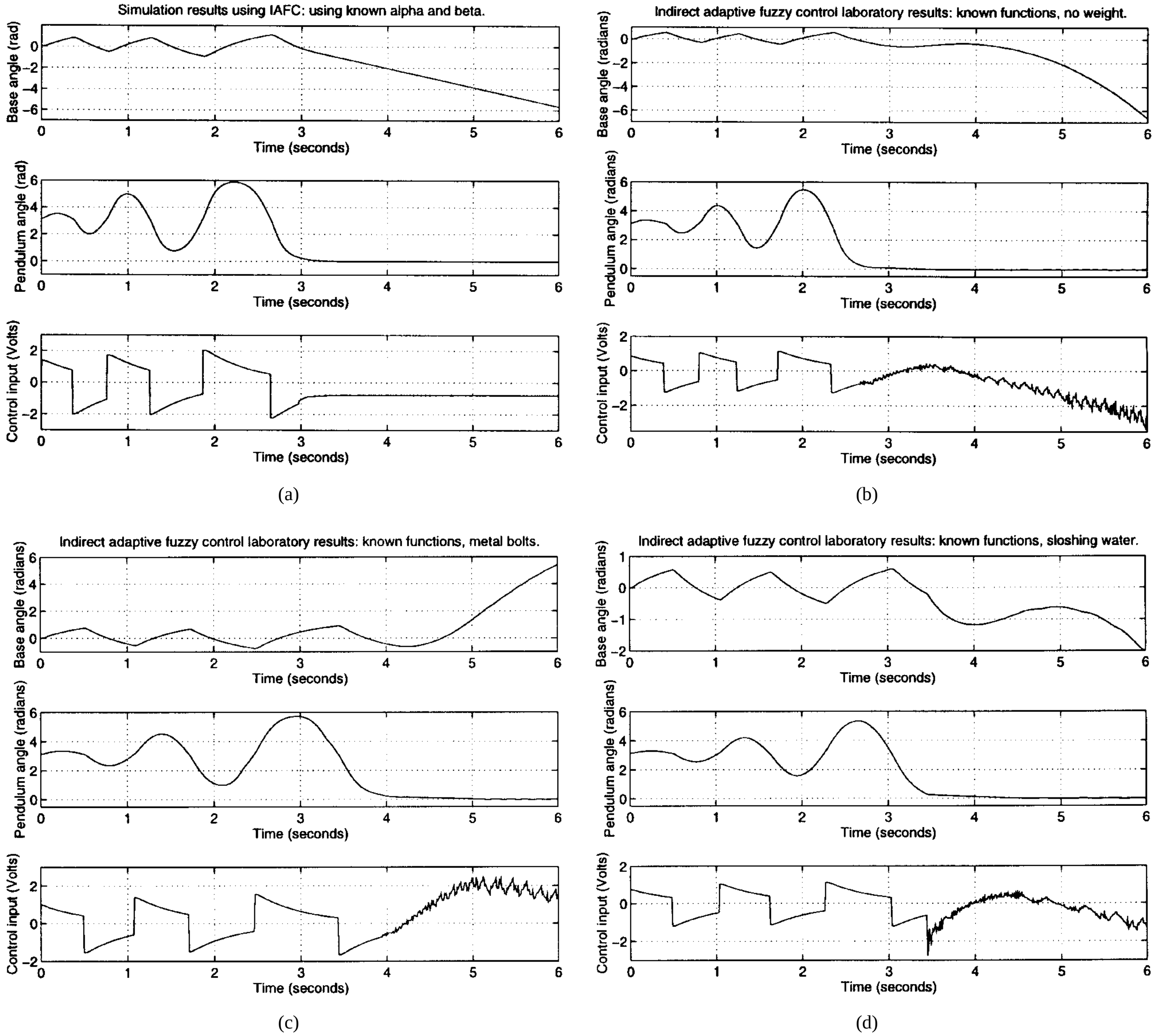
$$\alpha_k(t) = a_2 x_2 + a_3 \sin x_3 + a_4 x_4 \tag{19}$$

$$\beta_k(t) = b_2. \tag{20}$$

Note that α_k and β_k are known functions of time since we can measure the entire state of the plant.

The unknown functions $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ can be represented using fuzzy systems as in (11) and (12), and this optimum representation can then be approximated using (13) and (14) where we use the same vector of functions z we used above. Since we know nothing about $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$, the most logical way to initialize their fuzzy system approximation is by letting $A_\alpha(0) = \mathbf{0}$ and $A_\beta(0) = \mathbf{0}$. In this way, the adaptation mechanism will attempt to identify the plant by introducing variations to the functions defined by (19) and (20). Notice the fundamental difference that this design has with respect to the previous one: above, the input–output dynamics of the system were represented entirely by $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$, which were estimated by the adaptation mechanism. Here, we let the IAFC estimate *perturbations* off $\alpha_k(t)$ and $\beta_k(t)$ [recall that $\alpha_k(t)$ and $\beta_k(t)$ contain all our knowledge about the plant] using $\hat{\alpha}(\mathbf{x})$ and $\hat{\beta}(\mathbf{x})$. As we will see, the characteristics of the adaptive process change based on how we define and initialize the system.

The approximation bounds $D_\alpha(\mathbf{x})$ and $D_\beta(\mathbf{x})$ need not be reset with this configuration, because the representation errors $d_\alpha(\mathbf{x})$ and $d_\beta(\mathbf{x})$, using (19) and (20), are expected to be



(a)

(b)

(c)

(d)

Fig. 6. Plant dynamics knowledge used: (a) IAFC simulation. (b) Experimental results of IAFC with nominal plant. (c) Experimental results of IAFC with disturbance—metal bolts. (d) Experimental results of IAFC with disturbance—sloshing water.

of the same order of magnitude as in the previous case, and possibly less, or at most, equal; therefore, they are taken as defined before, with their respective values for simulation and implementation. Following a similar reasoning, the functions $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ are expected to be less than or equal in magnitude to (9) and (10), respectively; thus, although we could redefine the bounds $\alpha_1(\mathbf{x})$ and $\beta_1(\mathbf{x})$, we chose to keep them as in the previous subsection for simplicity.

To completely determine the controller it is necessary to redefine the certainty equivalence control term, as $u_{ce} = 1/[\beta_k(t)+\hat{\beta}(\mathbf{x})]\{-[\alpha_k(t)+\hat{\alpha}(\mathbf{x})]+\nu_a(t)\}$. The sliding mode and bounding control terms as well as the adaptation law (18) are taken without change, and the indirect adaptive control law is given by (17).

Observation of the simulation results in Fig. 6(a) using this modified IAFC shows little difference with Fig. 5(a); apparently, the basic characteristics of the controller remain the same, i.e., the state x_1 is still unbounded, and the pendulum is balanced with a very similar control input. Implementation of the controller on the nominal plant, as seen in Fig. 6(b), shows a slightly faster error convergence to zero. The most notable differences arise when the plant is disturbed; in Fig. 6(c) we see that the controller is able to handle the bolts disturbance effectively and without saturation of the control input. The same good behavior is observed in Fig. 6(d), where the plant is under the effects of sloshing water dynamics; initially, the pendulum is not perfectly balanced, but eventually the error converges to zero.

F. Direct Adaptive Fuzzy Control

Now we turn our attention to the direct adaptive fuzzy control (DAFC) method of [2] and [3] for the inverted pendulum. Using IAFC, a controller is constructed that seeks to identify the plant dynamics and use its best estimate to produce an approximation to a feedback linearizing law. Here, the approach is rather to search for an unknown control law that provides (at least) asymptotically stable tracking and is able to compensate for disturbances and maintain stability. As was the case with IAFC, the DAFC methodology allows the designer to use previous knowledge or experience with the plant in various ways. Here, we will illustrate two representative possibilities, and we will see that is possible to obtain significantly different control results depending on the approach taken.

In IAFC, it was possible to use a known part of the plant dynamics represented by α_k and β_k in the control design. We saw that for the pendulum application it was beneficial to include the known dynamics because it increased the robustness of the design. DAFC provides the designer with a method to incorporate a best guess of what the controller should be (below we will call this the "known controller," denoted by u_k). The algorithm then adaptively tunes a fuzzy controller to compensate for inaccuracies in our choice of this known controller.

1) Design Using Feedback Linearization as a Known Controller: As described in [2] and [3], DAFC is a somewhat more restrictive technique than its indirect counterpart since, in addition to the assumption that the plant is minimum-phase, it is also required that the system input–output (8) is such that $\beta_k(t) = 0$ for $t \geq 0$; further, it is assumed that $\beta(\mathbf{x})$ is bounded by two finite constants, β_0 and β_1. For the pendulum, this assumption holds since $-\infty < \beta_1 \leq \beta(\mathbf{x}) \leq \beta_0 < 0$, where, for instance, we take, as before, $\beta_0 = -100$ and $\beta_1 = -140$. The last plant assumption needed in [2] and [3] is that for some $B(\mathbf{x}) \geq 0$, $|\dot{\beta}(\mathbf{x})| \leq B(\mathbf{x})$. Since $\beta(\mathbf{x})$ is expected to be a constant, we can safely set $B(\mathbf{x}) = 0$, and the assumption holds.

Note that the control equations derived in [2] and [3] are based on the premise that $\beta(\mathbf{x})$ is positive, but it is stated there that the laws can be modified to allow for the negative case. Thus, the equations used here will be slightly modified versions of those in [2] and [3], as required by the characteristics of the pendulum; specifically, the adaptation differential equation and the sliding-mode control term will each have a small but crucial sign change.

Let u^* be an unknown ideal controller that we will try to approximate. In [2] and [3], this ideal controller is assumed to be a feedback linearizing law of the form $u^* = 1/\beta(\mathbf{x})[-\alpha(\mathbf{x}) + \nu_a(t)]$ In general, it is possible to express u^* in terms of a Takagi–Sugeno fuzzy system, as $u^* = z_u^\top A_u^* \zeta_u + u_k + d_u(\mathbf{x})$ where u_k is some known controller term, which we will use in this section and set equal to zero in the next, and $d_u(\mathbf{x})$ is the error between the fuzzy representation and u^*. It is assumed that $D_u(\mathbf{x}) \geq |d_u(\mathbf{x})|$, where $D_u(\mathbf{x})$ is a known bound for the error. In practice, it is often hard to have a concrete idea about the magnitude of $D_u(\mathbf{x})$, because the relation between u^* and its fuzzy representation might be difficult to characterize; however, it is much easier to begin with a rough, intuitive idea about this bound, and then iterate the design process and adjust it, until the performance of the controller indicates that one is close to the right value. For simulation, we found that $D_u(\mathbf{x}) = 0.01$ gave us good results, and in the laboratory, we increased it to $D_u(\mathbf{x}) = 0.1$. These bounds are both relatively small, which indicates that the fuzzy system we used, although a simple one, could represent the ideal controller with sufficient accuracy.

We are going to search for u^* using

$$\hat{u} = z^\top A_u(t)\zeta_u + u_k \tag{21}$$

where $\zeta_u \in \Re^5$ is as defined for the IAFC, with the fuzzy sets of Fig. 4. The matrix $A_u(t) \in \Re^{5\times 5}$ is adaptively updated online, and the function vector z is taken as in the previous Section. The fuzzy system again uses only five rules, as given by (15), and now each $f_i(\mathbf{x})$ is a row of the matrix $z^\top A_u(t)$. *To approximate a feedback linearizing controller* we will define u_k as in law (4), and we will take ν as in the feedback linearization design. Further, following the same line of thought as in Section II-E.2, we initialize the fuzzy system with $A_u(0) = \mathbf{0}$.

The DAFC control law is given by $u_d = \hat{u} + u_{sd} + u_{bd}$. It is formed by three terms: the fuzzy approximation to the optimum controller (21), and sliding and bounding control terms. Take the signals e_s and $\nu_a(t)$ as defined for the IAFC case. Since (as noted above) $B(\mathbf{x}) = 0$, the sliding-mode term is given by $u_{sd} = -D_u(\mathbf{x})\,\mathrm{sgn}\,(e_s)$. Note the minus sign which is a result of the fact that $\beta(\mathbf{x}) < 0$.

The bounding-control term needs the assumption that $\alpha(\mathbf{x})$ is bounded, with $|\alpha(\mathbf{x})| \leq \alpha_1(\mathbf{x})$. We take $\alpha_1(\mathbf{x})$ as defined before; then, if $e_s > M_e$, $u_{bd} = \{|\hat{u}| + |u_{sd}| + [\alpha_1(\mathbf{x}) + |\nu_a(t)|]/\beta_0\}\,\mathrm{sgn}\,(e_s)$ and $u_{bd} = 0$ otherwise.

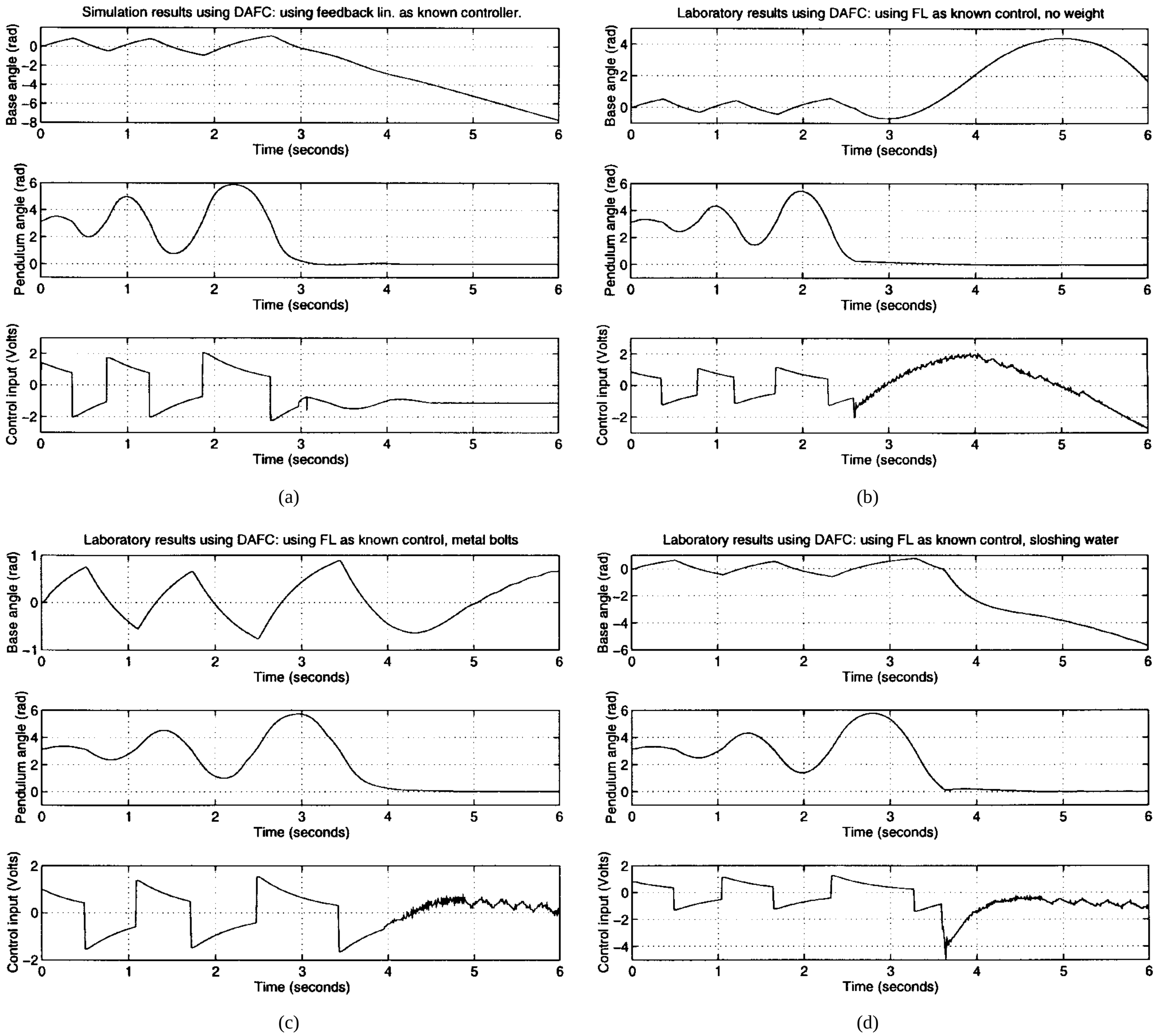
For simulation, we used $M_e = 0.6$ and increased it to $M_e = 2.5$ in implementation. Please refer to the discussion on IAFC for an explanation on how we determined these values.

The last part of the DAFC mechanism is the adaptation law, which is chosen in such a way that the output error converges asymptotically to zero, and the parameter error remains at least bounded. This law is given, in general, by

$$\dot{A}_u(t) = Q_u^{-1}\, z\zeta_u^\top[-e_s - q(t)]. \tag{22}$$

Again, note the minus sign for e_s. The parameter $q(t)$ can be chosen nonzero to potentially improve adaptation [2], [3], but here we took $q(t) = 0$ for $t \geq 0$. For simulation, we used $Q_u = 0.9\,I_5$, and in experimentation we decreased the gain slightly to $Q_u = 0.5\,I_5$. With these choices the algorithm was able to adapt and estimate the control law $\hat{u}$ fast enough to perform well and compensate for disturbances, but without inducing oscillations typical of a too high adaptation rate.

Fig. 7(a) shows the simulation results with this controller. It has a behavior typical of feedback linearizing controllers on this plant: the control input settles and oscillates around a nonzero value, thus keeping the pendulum base rotating. Observe in Fig. 7(b) the performance of the DAFC design on the nominal plant: the error is effectively decreased to zero, and the behavior of the base is similar to the previous cases. Again, the advantages given by the adaptive capability of this algorithm appear most distinctively in the presence of strong disturbances: the controller is quite successful with both the metal bolts [Fig. 7(c)] and the sloshing water [Fig. 7(d)]. The pendulum is kept balanced, and the control input remains within small bounds around zero. Thus, this design proved to be robust and reliable although it still has the weakness that all the other adaptive controllers presented in this work (until now) share—it is not able to deal with the marginal stability



(a) (b) (c) (d)

Fig. 7. DAFC using feedback linearizing u_k. (a) DAFC simulation. (b) Experimental results of DAFC with nominal plant. (c) Experimental results of DAFC with disturbance—metal bolts. (d) Experimental results of DAFC with disturbance—sloshing water.

condition of the system's zero dynamics. Therefore, as a last and, in our opinion, best fuzzy adaptive design example, we will now describe a DAFC that cannot only compensate for the induced disturbances (and, in fact, it does it with greater ease than all the previous controllers), but is also able to keep state boundedness, even though the theoretical analysis of [2] and [3] does not predict it (recall that such analysis does not preclude it).

2) Using the LQR to Obtain Boundedness: Although the theoretical analysis in [2] and [3] uses the assumption that the unknown control law u^*, which the DAFC tries to identify as a feedback linearizing law, it was found experimentally that it is not necessarily the case. If the right known controller is used and/or the adaptation mechanism is initialized appropriately, then the adaptation algorithm will converge to a controller that might behave in a very different manner because this mechanism seems to try to find the (local) optimum controller closest to its starting point in the search space, and this optimum does not necessarily have to be a feedback linearizing controller.

This finding is of special importance when the control design task involves dealing with a nonminimum phase plant like the pendulum, for which feedback linearization-based adaptive techniques have the limitation of being unable to maintain complete state boundedness. As stated before, the unboundedness of the state x_1 is admissible for the pendulum, but it might not be for other systems.

Consider, for instance, that a nonadaptive controller is available that can control the nonminimum phase plant with state boundedness. Then, it is possible that the desirable boundedness characteristics of this controller can be incorporated into the DAFC design, and enhanced by the robustness that the adaptive method provides. It is not yet known how to characterize, in general, the controllers that can be used in such a way; however, for our present study, a most natural and intuitive choice for this purpose is the LQR. This controller implements a linear function of the plant states, and is, therefore, able to drive the state error to zero for the nominal plant while maintaining state boundedness. Observe in Fig. 2(a) and (b) that all the plant states are indeed kept bounded. The LQR was shown to have a very good performance on the nominal, undisturbed system. Nevertheless, it fails immediately when significant disturbances are introduced.

A DAFC will be designed based on the LQR, so that its good behavior in terms of state boundedness can be kept, and its weakness regarding plant disturbances eliminated. Two different, and functionally equivalent ways were found to accomplish this. The first makes use of the term u_k, as illustrated above. The second uses an appropriate initialization of the matrix A_u. Since the use of u_k has already been shown, only the second approach will be described here.

Again, take the control law u_d defined above. The bounding and sliding-mode control terms are taken without changes. Also, the adaptation law (22) is used, now with a smaller gain (i.e., we slow adaptation down), $Q_u = 0.005\, I_5$, for simulation and implementation purposes. This adaptation gain was chosen via tuning of the controller. We found that higher gains tended to produce a more oscillatory behavior.

The fundamental difference between this and the previous design lies in the ideal u^* that we aim to identify. Before, the adaptive search was configured in such a way that the mechanism converged to a feedback linearizing law; now, we want it to identify a control input that behaves basically like an LQR, i.e., we want to implement an *adaptive LQR*. To do this, it is necessary to start the adaptation algorithm at a point in the search space in the proximity of the ideal LQR controller u^*. The closest approximation we have to this ideal u^* is the state feedback gain vector K of the LQR controller. Therefore, we will use it to initialize the fuzzy approximation of the desired control $\hat{u}$. Take the fuzzy system described by (21), with the same functions vector z as before, and let $u_k \equiv 0$. Then we initialize the matrix $A_u(t)$ as

$$A_u(0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \\ 1 & 1 & 1 & 1 & 1 \\ 10.8 & 10.8 & 10.8 & 10.8 & 10.8 \\ 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \end{bmatrix}. \tag{23}$$

Notice that the sign of the gains has been reversed since in this case we do not use the state error $\mathbf{x_r} - \mathbf{x}$, but rather the vector z, which consists of functions of the states themselves. It is worth mentioning that an alternative similar way of implementing this design consists of using the control term $u_k = K^\top \mathbf{x}$ (i.e., we set u_k equal to the LQR state feedback law) and letting $A_u(0) = \mathbf{0}$. We have tested this approach and it also works very well.

In this way, the design is complete, and the obtained results corroborate our expectations about it. We see in Fig. 8(a) the behavior of the controller in simulation. Observe that it closely resembles the performance of the LQR in Fig. 2(a), both in terms of the states and the control input it produces.

Fig. 8(b) shows the experimental results of the modified DAFC on the nominal plant. The pendulum is balanced with a control input that approaches zero in average (which means that the state x_1 is not going to grow without bound), and the performance is similar to that of the LQR in Fig. 2(b), although the output error is not exactly zero. The most interesting results are found in Fig. 8(c) and (d). We note that under both the metal bolts and the sloshing water disturbances the controller is able to maintain convergence and, in addition, it has a behavior much like that of an LQR retuned for the disturbed system: the pendulum base does not keep rotating, but lightly oscillates around a constant position, and the pendulum is balanced with a control input that has an average value close to zero. We observe in both cases, and most distinctly in the case of the water, how the controller adapts to the system with random disturbances; the control-input oscillations are relatively large at first, and after a couple of seconds decrease in amplitude as the DAFC approximates the ideal controller more and more. At the same time, the error converges to zero, and the base movement decreases.

G. Summary of Pendulum Results

We have studied several control approaches for the rotational inverted pendulum. The first two are nonadaptive, conventional controllers—an LQR and a feedback linearizing controller. We saw that these methods present an adequate behavior on the nominal plant in terms of our basic control objective, which is to balance the pendulum. We saw that feedback linearization has the disadvantage of making the state x_1 unbounded, due to the nonminimum phase nature of the pendulum. From a practical point of view, this is an undesirable feature also shared by several of the adaptive schemes we analyzed.

We then applied adaptive feedback linearization and indirect and direct adaptive fuzzy control to the pendulum. We found that AFL and IAFC without plant knowledge had similar troubles when working under disturbances. We were able to increase robustness using our plant knowledge in IAFC and by specifying the known controller in DAFC as a feedback linearizing controller. Finally, the DAFC design with LQR initialization presented a very interesting feature—it retained the "good" characteristics of the LQR (state boundedness, error convergence) and at the same time added the benefits of adaptation (apparent robustness to disturbances).

One must be careful in trying to evaluate these results. It is probably not fair to say that AFL and IAFC "failed" and DAFC "succeeded;" recall that the pendulum does not satisfy the zero-dynamics assumption of all these methods. However, our experience indicates that at least in some cases, the adaptive fuzzy methods we investigated have an advantage with respect to the conventional methods—they allow for more design flexibility. This is clearly illustrated by our two IAFC designs. The IAFC without plant knowledge performed poorly under disturbances, similar to adaptive feedback linearization. However, the IAFC method allowed us to improve performance by using our plant knowledge more effectively in the control design. In a similar way, DAFC using a feedback linearizing law as the known part of the controller displayed an improved behavior in comparison with the first two adaptive techniques. Apparently, the explicit use of our knowledge of the plant in one case,

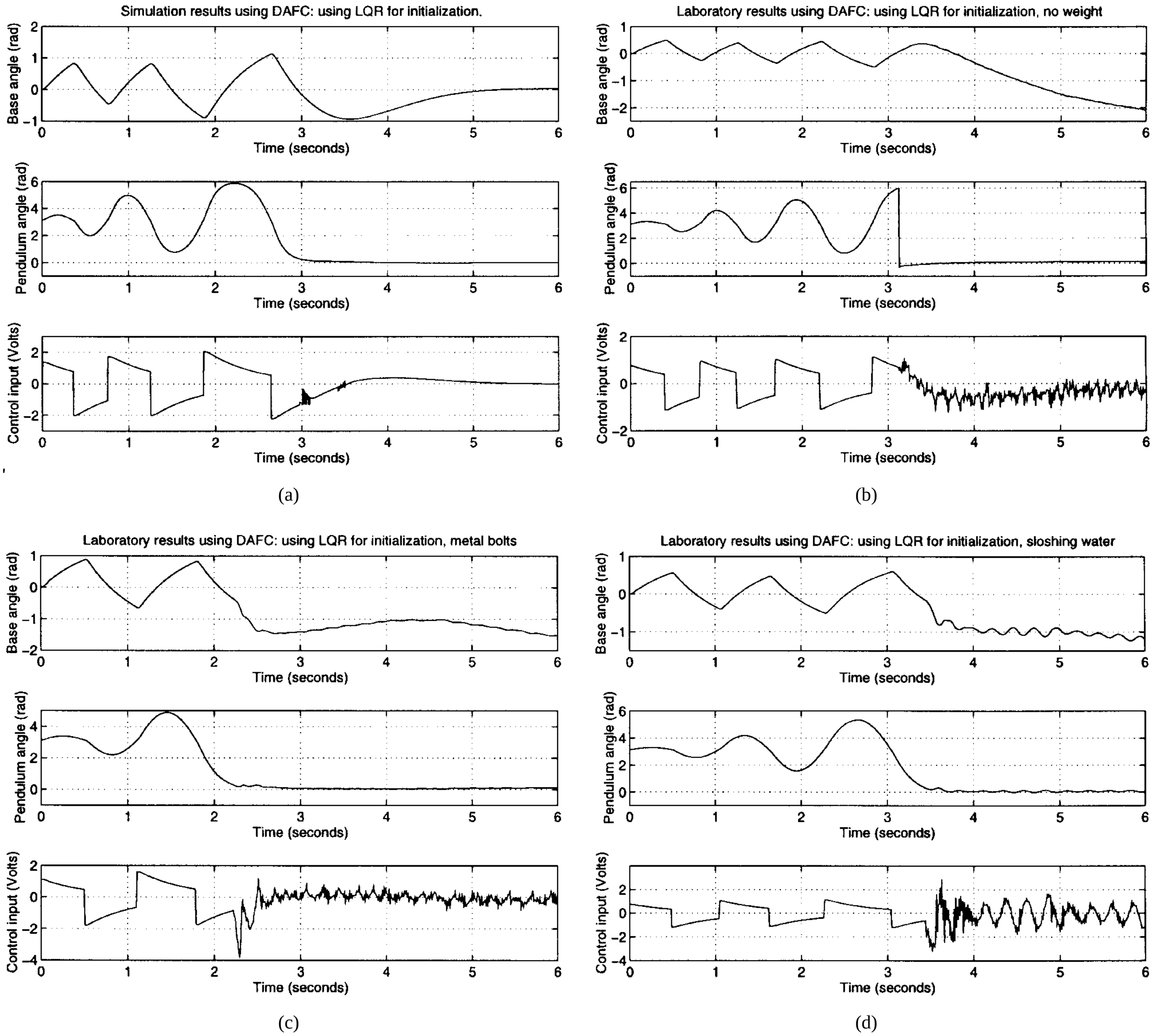


Fig. 8. DAFC initialized as LQR. (a) DAFC simulation. (b) Experimental results of DAFC with nominal plant. (c) Experimental results of DAFC with disturbance—metal bolts. (d) Experimental results of DAFC with disturbance—sloshing water.

and of what the control law should be in the other, increased the robustness of the algorithms. Furthermore, we managed to obtain an even greater improvement by heuristically turning the DAFC technique into an "adaptive LQR." Although we provide no theoretical justification for this result, it seems reasonable to think that direct adaptive techniques, in general, are fundamentally different from indirect adaptive methods; direct adaptive controllers might be regarded as generalized search mechanisms that are able to approximate different local optimum points in their search space (the controller search space).

III. Process Control Experiment

The process control experiment in our laboratory has been designed to emulate systems found in chemical processes by providing the ability to study liquid level control with various disturbances and plant variations. Other research on intelligent control for this system can be found in [17]. Work similar to this tank experiment, but with a slightly different setup can be found in [38]. In this section, we develop a conventional feedback linearizing controller and an indirect adaptive fuzzy controller, and we compare their performance for a variety of experimental conditions.

A. Experimental Setup

The process control experiment consists of two tanks, as shown in Fig. 9. The "fill" tank contains a liquid whose volume we wish to control (note that this volume is proportional to the liquid level). We denote the liquid volume by L_f and measure it using gallons. When full, the fill tank contains ten gallons of liquid. The reference input, which is a desired level, is denoted by L_d. The second, a "reservoir" tank, contains the liquid that will be pumped into and out of the fill tank and is the same size as the fill tank. There are two controlled pumps and another pump that is used for creating a disturbance. The first pump is a variable rate dc pump (which we denote by

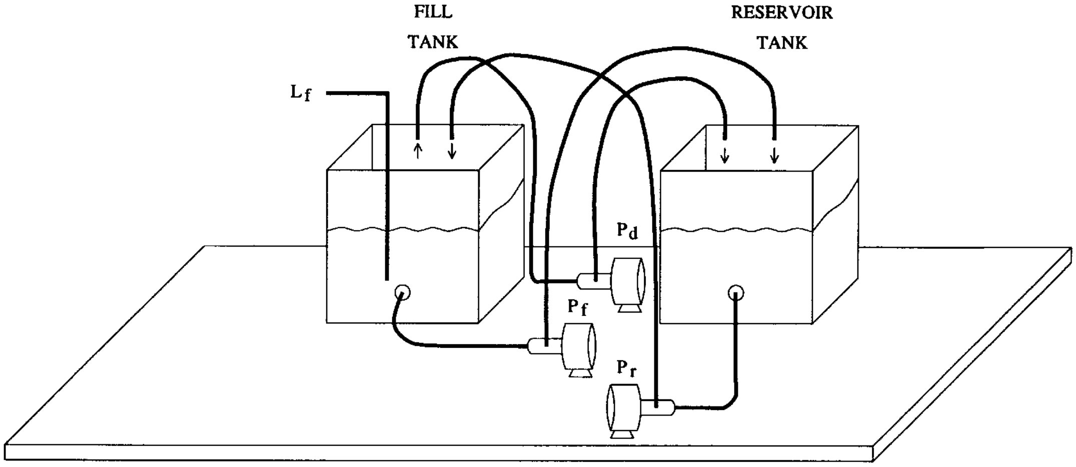


Fig. 9. Process control experiment.

P_r) which pumps liquid from the reservoir tank into the fill tank. The next pump is an ac pump (which we denote by P_f) which can only be turned off and on. This pump will be used to control the amount of liquid leaving the fill tank. The last pump, another variable rate dc pump (which we denote by P_d), is used to create a disturbance by removing liquid from the fill tank. The control input to the system is a single voltage u where a positive value of sufficient magnitude will cause the dc pump P_r to pump liquid into the fill tank and a negative u of sufficient magnitude will cause the ac pump P_f to pump liquid out of the fill tank.

The pumps have dead zones of different magnitudes and saturation nonlinearities that make the control problem of regulating the liquid level a challenging one. Also, the pumps introduce electrical noise and delays into the system. Finally, sensing problems are caused by the Styrofoam® ball used to measure the liquid level when the liquid surface oscillates. We will more carefully quantify some of these effects by providing a mathematical model of the plant.[5]

B. Model

Using some basic modeling ideas, we have found that a reasonably good model of the experiment is given by

$$\dot{L}_f = \alpha_r(u) - \alpha_f(L_f) - \rho(L_f) \tag{24}$$

where $\alpha_f(L_f)$ is a liquid dependent disturbance caused by pump P_d, u is a voltage input (with values between −8.5 to 10.0 V) which controls pumps P_f and P_r, $\alpha_r(u)$ represents the combined effects of the pumps P_f and P_r, and $\rho(L_f)$ is some unknown but relatively small disturbance which is always positive (e.g., a small leak). Experimentally we have determined that $\alpha_r(u) = R(u)$ with $\alpha_f(L_f) = 0.87R[d(L_f)]$ if $d(L_f) \geq 0$ and zero otherwise, where

$$R(x) = \begin{cases} -0.0333, & \text{if} \quad x < -7.0 \\ 0.0000, & \text{if} \quad -7.0 \leq x \leq 4.3 \\ 0.0058\,x - 0.0092, & \text{if} \quad 4.3 < x < 10.0 \\ 0.0488, & \text{if} \quad 10.0 \leq x. \end{cases} \tag{25}$$

The disturbance was chosen as $d(L_f) = 15.0/\pi \tan^{-1}(L_f)$ for two reasons. First, we wanted a disturbance which took on values between 5.0–9.0 V since this fact would help ensure the model will operate in a "continuous" region (we wanted to stay away from the dead zones of the pumps). Second, we wanted a disturbance which was dependent on the volume of liquid since we actually view the disturbance as an effect from a human operator or other subsystem that will generally remove more liquid from the tank when there is more liquid available. The chosen disturbance meets both of these requirements.

Each controller was allowed to control the process for two different experimental conditions. Each test run constituted tracking a desired liquid volume reference input L_d that was of the form $L_d = 6 - 1.0e^{-0.02t}$ for $t \leq 250$ and $L_d = 5 + 1.0e^{-0.02t}$ for $250 < t < 500$. For simulation and implementation, we used a 0.25 s sampling period since the plant is a slow one. We investigated the nominal plant [i.e., $\alpha_f(L_f) = 0$] and the plant with disturbance [i.e., $\alpha_f(L_f) = 0.87d(L_f)$] that represents a degraded pump.

C. Feedback Linearizing Control

The first controller we study makes use of feedback linearization. This controller was designed assuming that we have complete knowledge of the disturbance since if this term is ignored, the feedback linearization procedure produces a simple proportional controller. For the indirect adaptive fuzzy controller and feedback linearizing controller, the small leaking disturbance $\rho(L_f)$ has been ignored in the design stage since it is considered to be an unknown effect. For brevity we omit the control equation development and proceed to report on our results with this controller, which assumes $R(x) = 0.0058x - 0.0092$, so $u = 0.2062 + 4.1539 \tan^{-1}(L_f) + 1/0.0058(\dot{L}_d + L_d - L_f)$.

Simulation and implementation plots are shown in Fig. 10 where we see that while our model represents the gross characteristics of the plant it is certainly not highly accurate.[6]

[5] For further reference on the process control experiment please consult [17].

[6] It is important to note that it is very difficult to come up with a highly accurate model of this process, partly for the reasons given where we described the experimental process, and due to the fact that over time, the experiment changes in a variety of ways (e.g., the filters in the pumps become dirty, which has a significant impact on the plant's behavior).

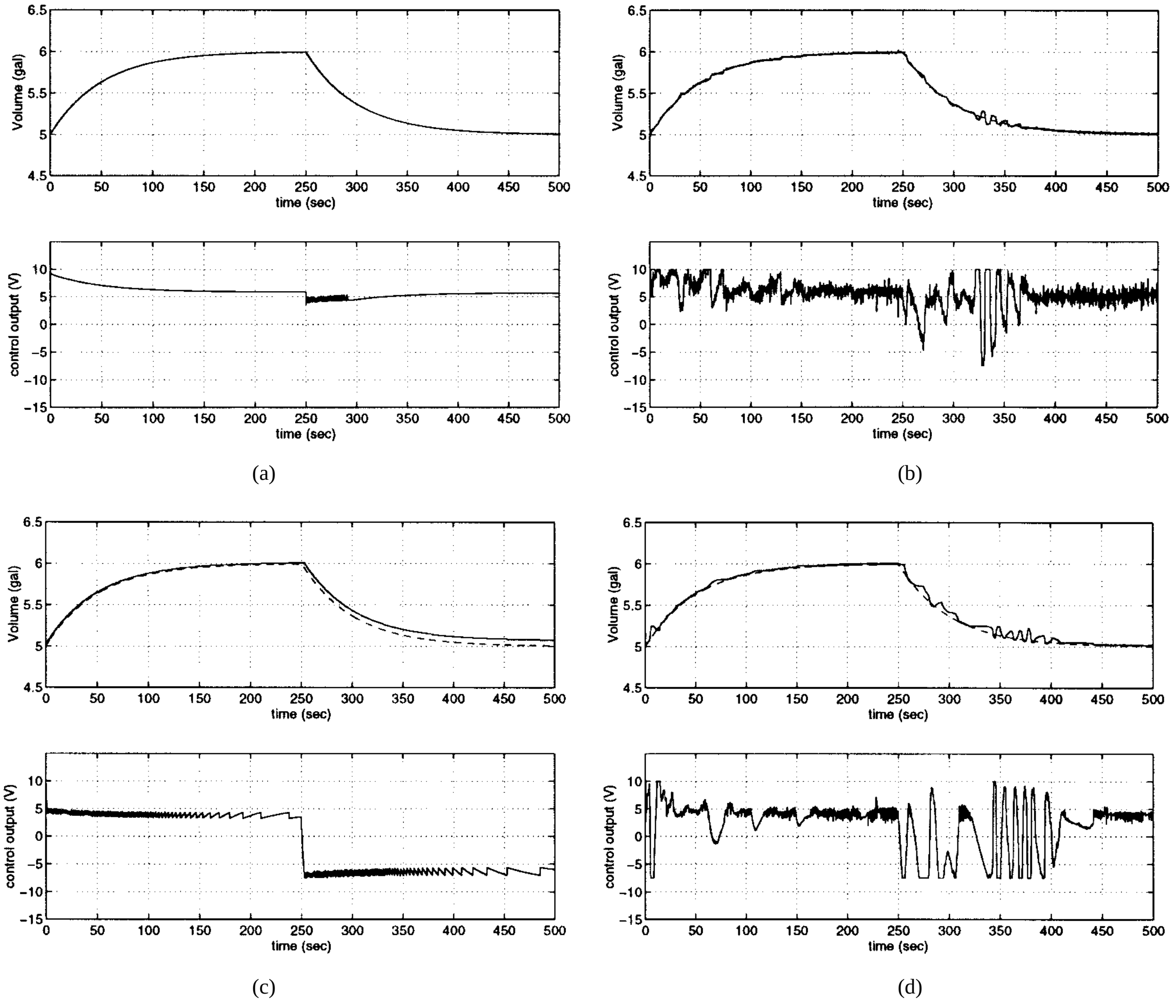


Fig. 10. (a) Feedback linearization simulation with disturbance. (b) Feedback linearization implementation with disturbance. (c) Feedback linearization simulation with nominal plant. (d) Feedback linearization implementation with nominal plant.

As seen in these figures, the feedback linearizing controller did well in the presence of the disturbance in both simulation and implementation (the measure of performance for the controllers was based on their ability to track the reference input, their ability to minimize control energy, and their ability to minimize control oscillations). The slight oscillation of the control voltage at about 250 s in the simulation was caused in part by the fact that the controller entered into one of the dead zones of the dc pump P_r. The feedback linearizing controller did not perform as well in the case where there was no disturbance in both simulation and implementation. This result was expected since the controller was specifically designed for the disturbance, and it shows that this feedback linearization design is not particularly robust to plant changes.

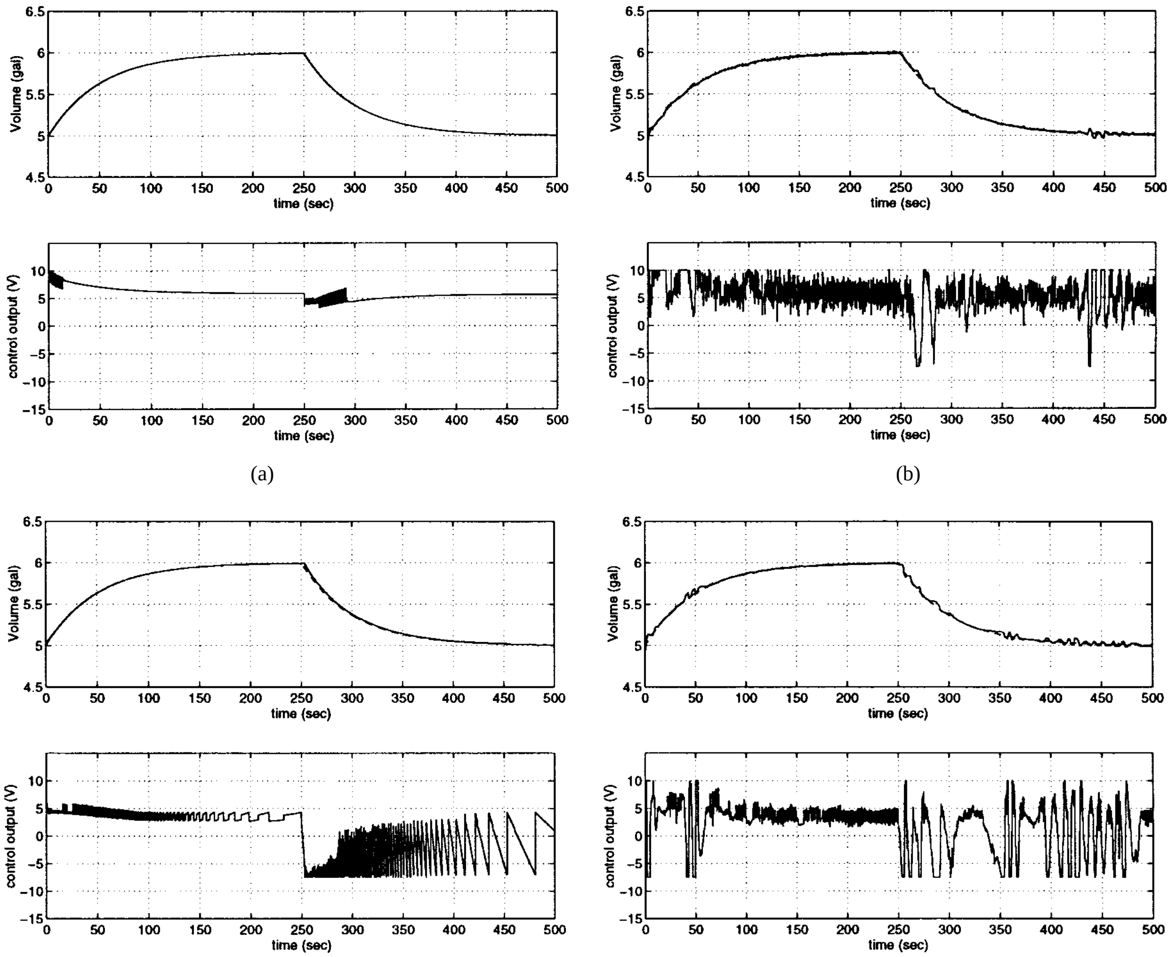
D. Indirect Adaptive Fuzzy Control

The next controller we studied was the IAFC. Since the design procedure for this method was already illustrated in Section II, we will not show the mathematical details and will concentrate on our results.

For this design we set the function $\alpha_k(t)$ to zero and only used $\beta_k(t)$, since this configuration gave us the best performance. For the estimator functions $\hat{\alpha}$ and $\hat{\beta}$, we used standard fuzzy systems (recall that a standard fuzzy system may be expressed as a special case of our Takagi–Sugeno form).

The results of the simulation and implementation are shown in Fig. 11. The IAFC performs about the same as the feedback linearizing controller for the experimental setup with a disturbance, but seems to outperform the feedback linearizing controller in tracking the reference input for the nominal plant setup. It tends to produce a more oscillating control output than the feedback linearizing controller, but both appear to use about the same amount of control energy.

Further work was done on examining whether the fuzzy systems actually estimated the plant parameters. According to the theory of the IAFC technique we are only guaranteed to have boundedness of parameter errors. There are two reasons why the estimates do no necessarily converge to their true values. First, the identifiers simply seek a model of the plant that will allow the adaptive controller to achieve its objective (i.e., stability of the closed-loop system and asymptotic convergence of the tracking error). Second, we would need to have "persistence of excitation" [36], and our simulations and implementations show this fact not to be true. Interestingly enough, parts of the fuzzy control surfaces were



(a)

(b)

(c)

(d)

Fig. 11. (a) IAFC simulation with disturbance. (b) IAFC implementation with disturbance. (c) IAFC simulation with nominal plant. (d) IAFC implementation with nominal plant.

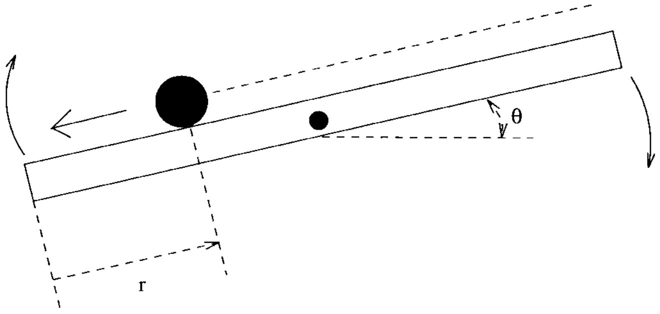


Fig. 12. Ball and beam system.

roughly tuned to represent the plant model. However, this fact did not hold for the entire surface, so there was probably no convergence of $\hat{\alpha}$ and $\hat{\beta}$ to their true values.

Overall, we suggest the IAFC technique tends to outperform the feedback linearizing control technique. Although the IAFC technique is an adaptive technique and the feedback linearizing controller is not; the feedback linearizing controller performs as well as or better than other adaptive techniques when performing comparisons in the presence of a disturbance [17]. This fact shows that the IAFC technique is a viable control technique for this experiment.

IV. Ball and Beam

Here we will use the well-known ball and beam experiment as an application example for the direct adaptive fuzzy control technique in [2] and [3]. We first introduce a mathematical model of the system which has a strong well-defined relative degree (as opposed to the ball-beam model in [39]). Then we develop a fuzzy controller for the experiment and its simulation and implementation results will be shown as a basis for comparison. Finally, we use this same fuzzy controller to design a direct adaptive fuzzy controller, as defined in [2] and [3], and the performance of the adaptive controller will be evaluated. Details about the theory behind the controllers will be omitted, and emphasis will be put on the results. For a detailed presentation of the adaptive fuzzy control theory please see Section II and refer to [2] and [3].

A. Experiment Setup, Modeling, and Simulation

Consider the ball and beam system in Fig. 12. The ball is allowed to roll (without sliding) along the beam, and its position relative to the left edge of the beam is denoted as r. The beam tilts about its center point, thus causing the ball to roll from one position to another. The control problem consists

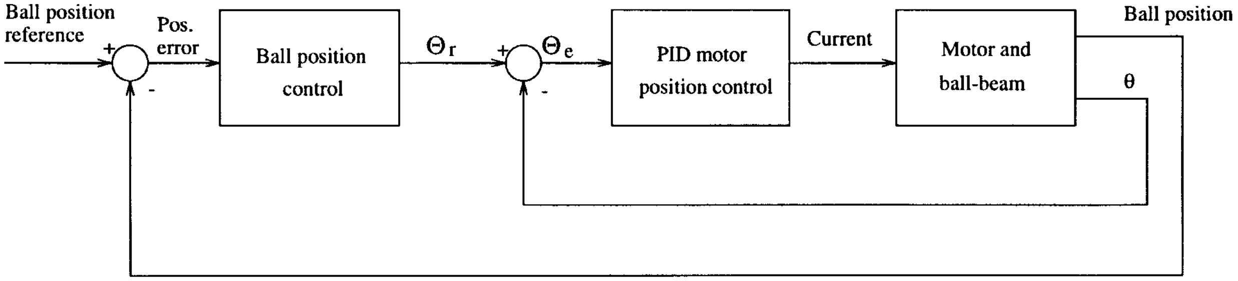


Fig. 13. *Motor-Ball-Beam Control Scheme:* Θ_r is the angle reference input, Θ_e is the angle error, and θ is the beam angle.

in designing a controller that tilts the beam in such a way that the ball is brought from its initial position to another desired position. The beam is driven by a dc motor whose shaft is attached to the center of the beam through a 50 : 1-turn ratio gear box. A ten-turn precision potentiometer is attached to the motor shaft to measure the angle of the beam. There are 32 photodiodes mounted along the bottom of the beam, spaced at 0.75-in intervals along the slot over which the ball rolls. Two lamps are positioned above the experiment so that they illuminate the whole beam area. The photodiodes detect the shadow cast by the ball to ascertain its position. Notice that this position sensing mechanism provides a discrete approximation to the actual position of the ball, and thus complicates the control task. A resistive strip could have been used to provide continuous position sensing, but this experiment was designed to be more challenging using the photodiodes (please see [40] for a detailed description of the setup).

Consider Fig. 13 for a block-diagram description of the system. Let i_a be the input armature current to the motor, θ the angle of the beam, and r the position of the ball on the beam. A simple proportional-integral-derivative (PID) controller is used to drive the motor and to position the beam at any desired angle. This controller takes as an input the error Θ_e between an angle reference Θ_r and the beam angle θ. The signal Θ_r is produced by the ball-position controller (which seeks to achieve our primary objective). By means of appropriate tuning of the PID controller, it is possible to achieve very good angle tracking, and since the inner loop has much faster dynamics than the outer loop, it can be considered virtually invisible to the ball-position controller.

Let $x_1 = \theta$ and $u = i_a$. Then, a linear state-space model of the motor is given by $\dot{x}_1 = x_2$, $\dot{x}_2 = x_3 + b_1 u$, $\dot{x}_3 = a_1 x_2 + a_2 x_3 + b_2 u$, where $a_1 = -87\,885.84$, $a_2 = -1416.4$, $b_1 = 280.12$, and $b_2 = -18\,577.14$ (the numerical values come from the motor specifications and the beam dimensions). If we now let $x_4 = r$ we can obtain two more equations which represent the ball and beam dynamics when the beam angle is taken as the input, using Newton's second law. Here, we are using the approximation $\sin x_1 \approx x_1$ (valid because the beam angle varies within a small range around zero) to have the input enter linearly. We have found that a reasonably good model of our ball-beam system is given by

$$\begin{aligned}\dot{x}_4 &= x_5\\ \dot{x}_5 &= a_3 x_1 + a_4 \tan^{-1}(100x_5)(e^{-10^4 x_5^2} - 1) \qquad (26)\end{aligned}$$

where $a_3 = -514.96$ and $a_4 = 9.84$, and the system output is $y = x_4$. The numerical values take into account the acceleration due to gravity and the friction constant between the ball and the beam (determined experimentally), and are scaled in such a way that the output y is in units of 0.75 in, which corresponds to the distance between the photodiodes. The function $\tan^{-1}(100x_5)(e^{-10^4 x_5^2} - 1)$ is an approximation to the acceleration due to friction that the ball experiences on the beam.

If the output y is repeatedly differentiated, we find that the system has a well-defined *strong relative degree* [33], [34] of four. Furthermore, it is possible to determine that the *zero dynamics* [34] of the system are exponentially stable (the details of the calculations involved are very tedious and are, therefore, omitted).

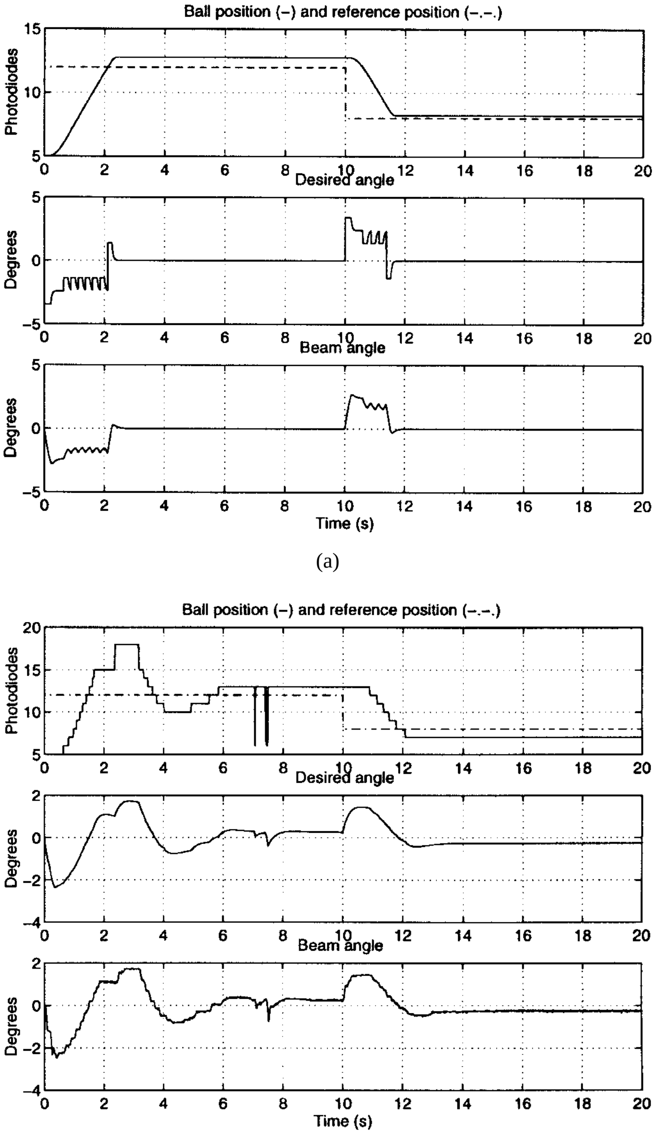
For our simulations below, we use a fourth-order Runge–Kutta numerical method with an integration step size of 0.001 s. In implementation, a sampling time of 0.01 s is used.

B. A Fuzzy Controller for the Ball and Beam

We now describe our results for the ball and beam using a standard fuzzy system for ball position control (details on the controller are omitted for brevity). The fuzzy controller has two inputs: the position error (defined as $e_r = r_{\text{ref}} - r$, where r_{ref} is the desired ball position) and the error derivative $\dot{e}_r$. We use singleton fuzzification for both inputs. For e_r we take five triangular membership functions; for $\dot{e}_r$ we use three, and we make a standard choice for the rule base. In the inference mechanism we use minimum to represent the premise and product for the implication, and centroid defuzzification is applied to obtain the output of the fuzzy system Θ_r.

For all simulations and laboratory experiments, the ball was initially on position five on the beam (that is, it was set on top of the photodiode number five, where the photodiodes are numbered from 0 to 31 beginning at the left); the desired position is twelve during the first ten seconds, and then changes to eight for another 10 s. The discrete ball position sensing mechanism of the real system is also used in simulation—the ball position controller receives not the exact value of r, but its position index (as represented by the number of photodiodes along the beam).

Fig. 14(a) shows the simulation results using the standard fuzzy controller. It has very good performance, with no overshoot, small settling time, and small steady-state error. The performance is, however, somewhat degraded in implementation, as shown in Fig. 14(b). There is overshoot and the error



(a)

(b)

Fig. 14. Direct fuzzy controller. (a) Simulation results. (b) Implementation results.

is larger. Notice the ball-position measurement noise [spikes between seventh and eighth seconds in Fig. 14(b)], which adds to the complexity of the control task.

C. Direct Adaptive Fuzzy Control

We will now use the fuzzy controller of the previous section to design a DAFC following the methodology of [2] and [3]. It is expected that the adaptive controller will achieve an improved performance and have a greater robustness against noise. Two assumptions about the plant have to be verified to apply this technique. First, the system has to be minimum phase; this condition may be easily verified for our proposed model. As we mentioned above, the ball-beam model has a relative degree of four. However, to simplify the design we ignore the motor dynamics (notice that this assumption requires the motor control to be efficient enough, i.e., it should provide good tracking with little lag; we can see that this is the case in Fig. 14: the motor shaft angle θ follows the reference θ_r closely), so we may use the approximation $x_1 \approx \Theta_r$. In this way, we only have to deal with the two-state system in (26). If we differentiate the output $y = x_4$ twice and take $x_1 = \Theta_r$, we find that

$$\ddot{y} = a_4 \tan^{-1}(100x_5)(e^{-10^4 x_5^2} - 1) + a_3\Theta_r. \qquad (27)$$

That is, the relative degree of the system that the ball position controller "sees" is two, and it has no zero dynamics; therefore, it is minimum phase (note that the much more complicated analysis that does not discard the motor dynamics leads to numerically similar results, with the system having a relative degree of four and asymptotically stable zero dynamics).

The second plant assumption of [2] and [3] can also be verified from (27). Using the notation from these references let $\beta(\mathbf{x}) = a_3 = -514.96$. Then there exist constants β_0 and β_1 such that $-\infty < \beta_1 \le \beta(\mathbf{x}) \le \beta_0 < 0$. Take, for instance, $\beta_1 = -600$ and $\beta_0 = -480$. It is also required that for some $B(\mathbf{x}) \ge 0$, $|\dot{\beta}(\mathbf{x})| \le B(\mathbf{x})$. The assumption holds if we let $B(\mathbf{x}) = 0$, since $\beta(\mathbf{x})$ is a constant.

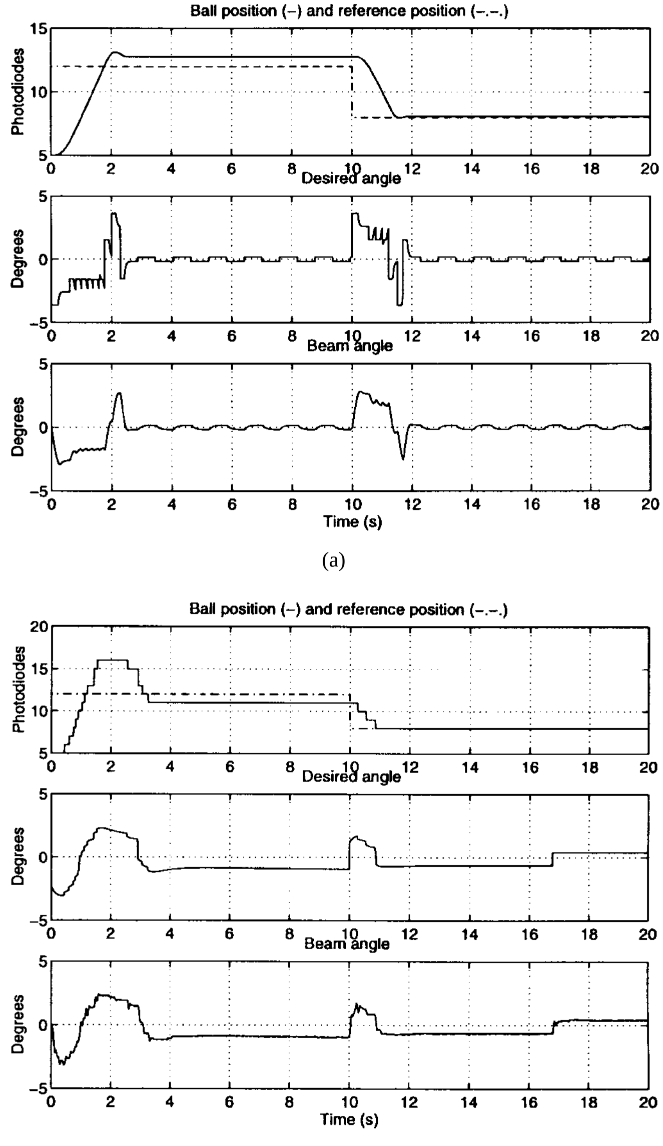
We will not show the development of the DAFC since it was already illustrated in Section II. However, there is one issue to notice: in our DAFC design we used the fuzzy system described above *without any modifications* for the term $\hat{u}$ (recall that a standard fuzzy system is a special case of the more general Takagi–Sugeno form we consider), and the adaptation laws adopted a simplified form.

Fig. 15(a) shows the performance of DAFC insimulation. We observe a behavior similar to that of the standard fuzzy controller in Fig. 14, with a slight overshoot in the output and roughly the same error and settling time characteristics. The real advantage of the adaptive method becomes plain in implementation, as shown in Fig. 15(b), where the settling time is significantly reduced, as well as the overshoot and the steady-state error. We have studied plots of how the output centers change over time while the DAFC tunes them and they are modified significantly more in implementation than in simulation due to the unmodeled nonlinear characteristics of the plant and to sensing noise. We omit the plots of these centers in the interest of brevity and since they are not particularly instructive.

In [40], a direct fuzzy controller was implemented for the ball and beam system, with experimental results apparently better than the ones in the present study; it is, however, hard to establish a comparison, because the experiment has broken several times since its construction, and a few of its components have been replaced.

V. Concluding Remarks

In this work, we have presented three case studies on the use of direct and indirect adaptive fuzzy control techniques developed in [1]–[3]. We have illustrated how the theory behind these controllers can be brought to practice by means of a design methodology, and have shown that the adaptive fuzzy controllers are able to work with complex plants under significant disturbances. Furthermore, a close correspondence between the theoretical predictions and the



(a)

(b)

Fig. 15. Direct adaptive fuzzy controller. (a) Simulation results. (b) Implementation results.

experimental results has been found. The performance of the adaptive fuzzy controllers has been compared with that of several other techniques which, according to our experience, present a good behavior for the plants we used. In the case of the rotational inverted pendulum, a comparison was made with a linear quadratic regulator and adaptive and nonadaptive feedback linearizing controllers. For the process control problem, the comparison was made with nonadaptive feedback linearization, and for the ball and beam we compared with a standard (nonadaptive) fuzzy controller.

Although the results we obtained seem to indicate that the DAFC and IAFC have comparable performance or are able to outperform the techniques they were compared with, it is still necessary to evaluate the performance of the controllers under a greater variety of conditions. It remains to be investigated how robust the controllers are against many different types of disturbances; for instance, we did not study how the adaptive fuzzy controllers react to "impact disturbances" on the pendulum. Generally speaking, disturbances of this type present a great challenge for adaptive schemes, especially if there is a sloshing liquid at the endpoint. The inverted pendulum is an example of a system with marginally stable zero dynamics that because of its nature, provides insight into the way the adaptive controllers work. And, as we saw, it was possible to design a DAFC that gave us bounded states in spite of the marginal stability of the zero dynamics. However, we provided no theoretical justification of the fact that this design worked as it did. In the cases of the process control tank and the ball-beam system, the adaptive fuzzy controllers were able to compensate for some disturbances and sensing noise, but it still seems possible that their performance could be improved (perhaps by further tuning of the techniques).

The results of our case studies suggest that investigating an extension of the adaptive schemes in [1]–[3] to certain types of nonminimum phase systems might be fruitful; if accomplished, such an extension would broaden the application spectrum of adaptive techniques in general. In addition, the IAFC and DAFC are single-input single-output schemes, and an extension to multi-input multi-output systems is currently under way; the indirect case has already been introduced in [41]. It is also important to notice that the adaptive fuzzy controllers in [1]–[3] are *continuous time* techniques; to implement them we used a digital computer, and thus were forced to implicitly use a discrete time approximation of the controllers. It is reasonable to think that a proof of stability is still applicable when a continuous time technique is discretized, but such a study is outside the scope of the present work. Recently, (in [42]) the authors have introduced a stable discrete-time adaptive control scheme for a class of nonlinear systems.

Acknowledgment

The authors would like to thank M. Widjaja, under the direction of Prof. S. Yurkovich, for developing the rotational inverted pendulum experiment. The process control experiment was initially set up by R. Garcar under the direction of Prof. Ü. Özgüner and the authors would like to thank them for this. They would also like to thank P. Jaklitsch, who later updated the experiment and made several improvements to it under the direction of Prof. S. Yurkovich, and J. Zumberge, working under the direction of Prof. K. Passino, who expanded on the improvements of P. Jacklitsch by coding all the software in C, improving the performance of the level measuring sensors, and adding an ac pump to establish the experimental testbed used for this study. They would also like to thank E. G. Laukonen, under the direction of Prof. S. Yurkovich, who developed the ball and beam experiment.

References

[1] J. T. Spooner and K. M. Passino, "Stable indirect adaptive control using fuzzy systems and neural networks," in *34th IEEE Conf. Decision Contr. Proc.,* New Orleans, LA, Dec. 1995, pp. 243–248.

[2] ——, "Stable direct adaptive control using fuzzy systems and neural networks," in *34th IEEE Conf. Decision Contr. Proc.,* New Orleans, LA, Dec. 1995, pp. 249–254.

[3] ——, "Stable adaptive control using fuzzy systems and neural networks," *IEEE Trans. Fuzzy Syst.,* vol. 4, pp. 339–359, Aug. 1996.

[4] L.-X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stabilty Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1994.

[5] K. M. Passino and S. Yurkovich, "Fuzzy control," in *Handbook on Control*, W. Levine, Ed. Boca Raton, FL: CRC, 1996, pp. 1001–1017.

[6] D. Driankov, H. Hellendoorn, and M. M. Reinfrank, *An Introduction to Fuzzy Control.* Heidelberg, Germany: Springer-Verlag, 1993.

[7] T. Procyk and E. Mamdani, "A linguistic self-organizing process controller," *Automatica*, vol. 15, no. 1, pp. 15–30, 1979.

[8] J. R. Layne and K. M. Passino, "Fuzzy model reference learning control," *J. Intell. Fuzzy Syst.*, vol. 4, no. 1, pp. 33–47, 1996.

[9] ——, "Fuzzy model reference learning control," in *Proc. 1st IEEE Conf. Contr. Applicat.*, Dayton, OH, Sept. 1992, pp. 686–691.

[10] ——, "Fuzzy model reference learning control for cargo ship steering," *IEEE Contr. Syst. Mag.*, vol. 13, pp. 23–34, Dec. 1993.

[11] W. A. Kwong and K. M. Passino, "Dynamically focused fuzzy learning control," *IEEE Trans. Syst., Man, Cybern.*, vol. 26, pp. 53–74, Feb. 1996.

[12] J. Layne, K. Passino, and S. Yurkovich, "Fuzzy learning control for antiskid braking systems," in *Proc. IEEE Conf. Decision Contr.*, Tucson, AZ, Dec. 1992, pp. 2523–2528.

[13] W. Kwong and K. Passino, "Fuzzy learning systems for aircraft control law reconfiguration," in *Proc. IEEE Int. Symp. Intell. Contr.*, Columbus, OH, Aug. 16–18, 1994, pp. 333–338.

[14] W. A. Kwong, K. M. Passino, E. G. Lauknonen, and S. Yurkovich, "Expert supervision of fuzzy learning systems for fault tolerant aircraft control," *Proc. IEEE, Spec. Issue Fuzzy Logic Eng. Applicat.*, vol. 83, pp. 466–483, Mar. 1995.

[15] ——, "Expert supervision of fuzzy learning systems with applications to reconfigurable control for aircraft," in *Proc. IEEE Conf. Decision Contr.*, Orlando, FL, Dec. 1994, pp. 4116–4121.

[16] W. Lennon and K. Passino, "Intelligent control for brake systems," in *Proc. IEEE Int. Symp. Intell. Contr.*, Monterey, CA, Aug. 1995, pp. 499–504.

[17] J. Zumberge and K. M. Passino, "A case study in intelligent control for a process control experiment," in *Proc. IEEE Int. Symp. Intell. Contr.*, Dearborn, MI, Sept. 1996, to be published.

[18] V. G. Moudgal, W. A. Kwong, K. M. Passino, and S. Yurkovich, "Fuzzy learning control for a flexible-link robot," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 199–210, May 1995.

[19] T. A. Johansen, "Fuzzy model based control: Stability, robustness, and performance issues," *IEEE Trans. Fuzzy Syst.*, vol. 2, pp. 221–234, Aug. 1994.

[20] C.-Y. Su and Y. Stepanenko, "Adaptive control of a class of nonlinear systems with fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol. 2, pp. 285–294, Nov. 1994.

[21] L.-X. Wang, "Stable adaptive fuzzy control of nonlinear systems," in *Proc. 31st Conf. Decision Cont.*, Tucson, AZ, 1992, pp. 2511–2516.

[22] ——, "A supervisory controller for fuzzy control systems that guarantees stability," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 1845–1847, Sept. 1994.

[23] J. T. Spooner and K. M. Passino, "Stable adaptive fuzzy control for an automated highway system," in *Proc. IEEE Int. Symp. Intell. Contr.*, Monterey, CA, pp. 531–536, Aug. 1995.

[24] F.-C. Chen and H. K. Khalil, "Adaptive control of nonlinear systems using neural networks—A dead-zone approach," in *Proc. Amer. Cont. Conf.*, Boston, MA, 1991, pp. 667–672.

[25] ——, "Adaptive control of nonlinear systems using neural networks," *Int. J. Contr.*, vol. 55, pp. 1299–1317, 1992.

[26] F.-C. Chen and W.-C. Tsao, "Adaptive control of linearizable discrete-time systems," in *Proc. Amer. Contr. Conf.*, Baltimore, MD, June 1994, pp. 880–881.

[27] F.-C. Chen and H. K. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 791–801, May 1995.

[28] M. M. Polycarpou and P. A. Ioannou, "Identification and control of nonlinear systems using neural network models: Design and stability analysis," Univ. Southern California, Dept. Electrical Engineering, Syst. Rep. 91-09-01, Sept. 1991.

[29] G. A. Rovithakis and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Trans. Syst., Man., Cybern.*, vol. 24, pp. 400–412, Mar. 1994.

[30] R. Ordonez, J. T. Spooner, and K. M. Passino, "Experiments and comparative analyses in adaptive fuzzy control," in *Proc. IEEE Int. Symp. Intell. Contr.*, Dearborn, MI, Sept. 1996, to be published.

[31] M. Widjaja and S. Yurkovich, "Intelligent control for swing up and balancing of an inverted pendulum system," in *Proc. 4th IEEE Int. Conf. Contr. Applicat.*, Albany, NY, Sept. 1995, pp. 534–542.

[32] S. Yurkovich and M. Widjaja, "Fuzzy controller synthesis for an inverted pendulum system," *IFAC Contr. Eng. Practice*, vol. 4, pp. 445–469, Apr. 1996.

[33] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control.* Englewood Cliffs, NJ: Prentice-Hall, 1991.

[34] M. Vidyasagar, *Nonlinear System Analysis*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[35] S. S. Sastry and A. Isidori, "Adaptive control of linearizable systems," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 1123–1131, Nov. 1989.

[36] S. S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[37] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, pp. 116–132, 1985.

[38] H. Thomas and E. Sebastian, "Approximately time optimal fuzzy control of a two-tank system," *IEEE Contr. Syst. Mag.*, vol. 14, pp. 24–30, June 1994.

[39] S. S. John Hauser and P. Kokotović, "Nonlinear control via approximate input-output linearization: The ball and beam example," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 392–398, Mar. 1992.

[40] E. G. Laukonen and S. Yurkovich, "A ball and beam testbed for fuzzy identification and control design," in *Amer. Contr. Conf.*, San Francisco, CA, June 1993, vol. 1, pp. 665–669.

[41] R. E. Ordonez, J. T. Spooner, and K. M. Passino, "Stable multi-input multi-output adaptive fuzzy control," in *IEEE Conf. Decision Contr.*, Dearborn, MI, Sept. 1996, pp. 31–36.

[42] J. T. Spooner, R. Ordonez, and K. M. Passino, "Stable direct adaptive control of a class of discrete time nonlinear systems," in *IFAC World Congress*, San Francisco, CA, July 1996, vol. K, pp. 343–348, invited paper.

Raúl Ordóñez was born in Guayaquil, Ecuador, on February 18, 1971. He received the B.S. degree (*summa cum laude*) from the Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey, México, in 1994, and the M.S. degree from The Ohio State University, Columbus, OH, in 1996, both in electrical engineering. He is currently working toward the Ph.D. at The Ohio State University.

In 1994, he received a Litton Fellowship and has since worked as a Teaching and a Research Assistant. His research interests include fuzzy and neural control, adaptive control, and stability analysis for nonlinear systems.

Jon Zumberge was born in Dayton, OH, in 1971. He received the B.S.E.E. degree, in 1994, and the M.S.E.E. degree, in 1996, both from The Ohio State University, Columbus.

He is currently working for Delphi Chassis Systems, Dayton as a Project Engineer. His research interests include adaptive fuzzy control, supervisory fuzzy control, and nonlinear identification via fuzzy systems.

Jeffrey T. Spooner received the B.A. degree in physics from Wittenberg University, Springfield, OH, in 1991, the B.S. degree in electrical engineering from The Ohio State University, Columbus, in 1993, and the M.S. degree in electrical engineering (with a specialization in control systems), in 1995, from The Ohio State University, Columbus.

Since 1995, he has been with the Control Subsystems Department at Sandia National Laboratories, Albuquerque, NM. His areas of interest include adaptive control, fuzzy systems, neural networks, and modeling.

Kevin M. Passino (S'79–M'90–SM'96) received the B.S.E.E. degree from Tri-State University, Angola, IN, in 1983, and the M.S. and Ph.D. degrees in electrical engineering from the University of Notre Dame, IN, in 1989.

He has worked in the control systems group at Magnavox Electronic Systems Co., Ft. Wayne, IN, on research in missile control, and at McDonnell Aircraft Co., St. Louis, MI, on research in flight control. He spent a year at Notre Dame as a Visiting Assistant Professor, and is currently an Associate Professor in the Department of Electrical Engineering at The Ohio State University, Columbus. He is an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, served as the Guest Editor for the 1993 *IEEE Control Systems Magazine Special Issue on Intelligent Control* and Guest Editor for a special track of papers on intelligent control for *IEEE Expert Magazine* in 1996, and is on the Editorial Board of the *International Journal for Engineering Applications of Artificial Intelligence*. He is co-editor (with P. J. Antsaklis) of the book *An Introduction to Intelligent and Autonomous Control* (Norwell, MA: Kluwer, 1993). His research interests include intelligent and autonomous control techniques, nonlinear analysis of intelligent control systems, failure detection and identification systems, and genetic algorithms for control.

Dr. Passino is a member of the IEEE Control Systems Society Board of Governors. He was the Publicity co-chair for the IEEE Conference on Decision and Control in Japan in 1996 and is the Workshops Chair for the 1997 IEEE Conference on Decision and Control. He was a Program Chairman for the 8th IEEE International Symposium on Intelligent Control in 1993, served as the Finance Chair for the 9th IEEE International Symposium on Intelligent Control, and is serving as the General Chair for the 11th IEEE International Symposium on Intelligent Control.