Hybrid Fuzzy-State Feedback Control for Fast Mechatronic Systems: Modeling and Experimental Validation on a Rotary Inverted Pendulum

Ali Gamal, Mohamed Emad, Essam Shaban, and Ayman Nada

Abstract— **This paper introduces a general control framework for fast mechatronic systems that integrates linearized state-space control with nonlinear fuzzy logic strategies. The approach is applied to the Rotary Inverted Pendulum (RIP), a benchmark system characterized by rapid dynamics and strong nonlinearity. The system's equations of motion are derived using the Euler-Lagrange method and linearized around the upright equilibrium point to design a state-feedback controller via the Linear Quadratic Regulator (LQR). To manage the system's nonlinear behavior during the swing-up phase and near-region transitions, a fuzzy logic controller is incorporated, dynamically tuning the control effort based on the pendulum's angular position and velocity. This hybrid scheme allows the linear controller to dominate where approximation is valid, while the fuzzy logic system handles broader, nonlinear dynamics. The full control architecture is implemented in LabVIEW and experimentally validated on the NI ELVIS II platform. Both simulation and hardware results show notable improvements in settling time, energy efficiency, and robustness. The proposed framework demonstrates high applicability to the control of fast-response mechatronic systems operating under mixed linear and nonlinear systems.**

I. INTRODUCTION

The Rotary Inverted Pendulum (RIP) is a classic benchmark system in control theory due to its inherent nonlinearity, underactuation, and open-loop instability [1]. Its dynamic complexity makes it an ideal testbed for validating advanced control strategies applicable to real-world systems.

Traditional control strategies like PID or LQR have been widely applied to the RIP. While PID controllers are simple to implement, their effectiveness is limited by the nonlinear nature of the system and the difficulty of parameter tuning [2]. LQR, on the other hand, offers optimal control within a linearized framework, but its robustness is often inadequate under uncertainties or external disturbances. These limitations have driven a significant research shift toward hybrid and intelligent control frameworks [3].

In recent years, fuzzy logic control (FLC) has gained popularity as a model-free, heuristic-based control technique capable of handling nonlinearity and imprecision in dynamic systems [4]. Its ability to encode expert knowledge in the form of intuitive rules makes it especially suitable for systems like RIP, where modeling uncertainties are non-negligible. Fuzzy logic has been successfully combined with classical controllers like PID, LQR, and sliding mode control (SMC) to form hybrid controllers that benefit from both analytical optimality and adaptive nonlinear behavior [5].

This paper proposes a hybrid control framework that leverages state-feedback control using LQR based on a linearized model around the upright equilibrium, and fuzzy logic control for real-time gain adaptation focused on the pendulum angle correction term.

Unlike fully fuzzy or fully linear schemes, our approach uses fuzzy logic only to tune the proportional gain of the pendulum angle, allowing for computational efficiency while still addressing nonlinear behavior during the stabilization phase. This allows the controller to switch adaptively between regions where the linear model is valid and where nonlinear compensation is essential.

A distinguishing feature of this work is the adoption of a Denavit-Hartenberg (DH) parameterization for the mathematical modeling of the rotary inverted pendulum system. This approach, commonly employed in industrial robotic systems [6], facilitates the construction of homogeneous transformation matrices and enables accurate symbolic evaluation of spatial velocities and kinetic energy.

The control scheme is implemented in NI LabVIEW using the Control Design and Simulation Toolkit [7], and deployed on an NI ELVIS II test rig. Experimental validation is presented alongside simulation results to demonstrate improved performance in settling time, disturbance rejection, and control efficiency. The proposed architecture demonstrates high applicability to fast-response mechatronic systems operating in mixed linear/nonlinear regimes, extending beyond the RIP testbed.

II. System description & Mathematical Modeling

The rotary inverted pendulum consists of L-shape arm which is connected to motor shaft and pivots between $\pm 180^\circ$ degrees. At the end of the arm, a pendulum body is suspended on a horizontal axis at the end of the L-shape arm. The pendulum and arm angle are measured by encoders attached. The output variables are the angle of the pendulum and the angle of the arm. The motor is mounted vertically in the ground, see Fig.(1). The pendulum link and weight combined has the mass M_p and a total length of L_p and can rotate freely in the vertical plane. It sets up the angle α in the vertical direction. The length from arm pivot to pendulum pivot is r_a and sets up the angle θ in the horizontal direction. The length from pendulum centre of mass to its pivot is l_p. The parameters used in the simulation are found in [8].

The equations of motion can be determined from Lagrange's equation,

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} + \frac{\partial U}{\partial q} = Q - \frac{\partial D}{\partial \dot{q}} \tag{1}$$

where T is the kinetic energy, U is the potential energy, Q is a generalized force vector, and $\frac{\partial D}{\partial \dot{q}}$ represents dissipative

Ali Gamal, Mohamed Emad, and Ayman Nada are with the Department of Mechatronics and Robotics (MTR), School of Innovative Design Engineering (IDE), Egypt-Japan University of Science and Technology (E-JUST), Alexandria 21934, Egypt. Emails: ali.gamal@ejust.edu.eg, mohamed.mohi@ejust.edu.eg, ayman.nada@ejust.edu.eg

Essam Shaban is with the College of Applied Industrial Technology, Jazan University, Saudi Arabia, and on leave from the Faculty of Engineering (Mataria), Helwan University, Cairo, Egypt. Email: e.e.shaban@gmail.com

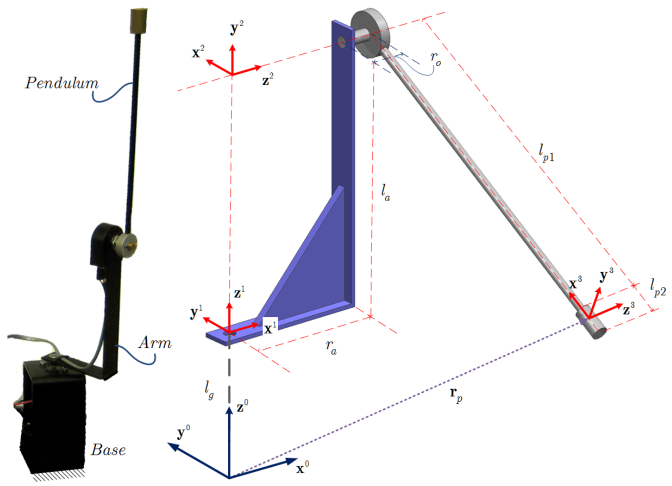


Fig. 1. Inverted Pendulum: Physical system & DH frames

energy from damping. The kinetic energy of the L-shape rotary arm is defined by the rotational kinetic energy

$$T_{arm} = \frac{1}{2} J_{arm}\ \dot{\theta}^2 \tag{2}$$

and the kinetic energy of the pendulum is the sum of rotational and translational kinetic energy

$$T_{pend} = \frac{1}{2} J_p\ \dot{\alpha}^2 + \frac{1}{2} m_p\ \dot{\mathbf{r}}_p^T \dot{\mathbf{r}}_p \tag{3}$$

where $\mathbf{r}_p$ is the position vector of the pendulum center of mass in spatial domain.In order to evaluate the kinetic energy of the rotating pendulum, Eq.(3), the term $\dot{\mathbf{r}}_p^T \dot{\mathbf{r}}_p$ should be expanded and symbolically manipulated. This vector can be evaluated by the aid of homogeneous transformation based on Denavit-Hartenberg (DH) notation [6], see Fig.(1). The DH parameters of the inverted pendulum system can be expressed as listed in Tab.(I), accordingly, the generalized HTM can be obtained by matrix multiplication as follows:

$$\mathbf{T}_{0,3} = \mathbf{T}_{0,1}\mathbf{T}_{1,2}\mathbf{T}_{2,3} = \left[\begin{array}{ccc|c} s_1 s_2 & s_1 c_2 & c_1 & r_2 c_1 - l_2 s_1 s_2 \\ -c_1 s_2 & -c_1 c_2 & s_1 & r_2 s_1 + l_2 c_1 s_2 \\ c_2 & -s_2 & 1 & l_g + l_a - l_2 c_2 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{4}$$

where $s_1 = \sin(\theta)$, $c_1 = \cos(\ \theta)$, $s_2 = \sin(\alpha)$, $c_2 = \cos(\alpha)$. Thus, the position vector of the pendulum center of mass can be expressed as

$$\mathbf{r}_p = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} r_2 c_1 - l_2 s_1 s_2 \\ r_2 s_1 + l_2 c_1 s_2 \\ l_g + l_a - l_2 c_2 \end{bmatrix} \tag{5}$$

The term $\dot{\mathbf{r}}_p^T \dot{\mathbf{r}}_p$ can be expanded as follow

$$\begin{aligned} \dot{\mathbf{r}}_p^T \dot{\mathbf{r}}_p &= \left(l_2^2 - l_2^2 c_2^2 + r_2^2\right) \dot{\theta}^2 + 2 r_2 l_2 c_2\ \dot{\theta}\ \dot{\alpha} + l_2^2\ \dot{\alpha}^2 \\ &= r_2^2 \dot{\theta}^2 + l_2^2 s_2^2\ \dot{\theta}^2 + 2 r_2 l_2 c_2\ \dot{\theta}\ \dot{\alpha} + l_2^2\ \dot{\alpha}^2 \end{aligned} \tag{6}$$

The total kinetic energy, T due to the motion of arm and pendulum can be expressed as

$$\begin{aligned} T &= \frac{1}{2} J_a \dot{\theta}^2 + \frac{1}{2} J_p \dot{\alpha}^2 + \frac{1}{2} m_p (r_2^2 \dot{\theta}^2 + l_2^2 s_2^2 \dot{\theta}^2 \\ &\quad + 2 r_2 l_2 c_2 \dot{\theta} \dot{\alpha} + l_2^2 \dot{\alpha}^2) \end{aligned} \tag{7}$$

The potential energy, U due to gravity can be expressed as

$$U = m_p g\ \Delta z_p \tag{8}$$

The Δz_p can be obtained as

$$\Delta z_p = z_p\big|_{\alpha=0} - z_p\big|_{\alpha\neq 0} = l_2 - l_2 c_2 \tag{9}$$

TABLE I

DH PARAMETERS OF ROTARY INVERTED PENDULUM

i	α_i	l_i	θ_i	d_i
0	0	0	0	l_g
1	$\frac{\pi}{2}$	0	$\theta + \frac{\pi}{2}$	l_a
2	0	l_2	$\alpha + \frac{\pi}{2}$	r_2

$r_2 = r_a + r_o$, $l_2 = l_{p1} + (l_{p2}/2)$

Thus,

$$U = m_p g l_2 \left(1 - c_2\right) \tag{10}$$

The dissipative energy, D due to damping can be written as

$$D = \frac{1}{2} C_a \dot{\theta}^2 + \frac{1}{2} C_p \dot{\alpha}^2 \tag{11}$$

Then, the equations of motion, can be concluded as

$$\begin{aligned} \left(m_p l_2^2 s_2^2 + m_p r_2^2 + J_a\right) \ddot{\theta} + \left(m_p l_2 r_2 c_2\right) \ddot{\alpha} + 2 m_p l_2^2 s_2 c_2\, \dot{\theta} \dot{\alpha} \\ - m_p l_2 r_2 s_2\, \dot{\alpha}^2 + C_a\, \dot{\theta} = \tau_1 \end{aligned}$$

$$\begin{aligned} \left(m_p l_2 r_2 c_2\right) \ddot{\theta} + \left(m_p l_2^2 + J_p\right) \ddot{\alpha} - m_p l_2^2 s_2 c_2\, \dot{\theta}^2 \\ + m_p g\, l_2 s_2 + C_p\, \dot{\alpha} = 0 \end{aligned}$$

which can be presented as in a compact size, one can write the equations of motion aa follow

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{Q}_g(\mathbf{q}) = \mathbf{Q} \tag{12}$$

where $\mathbf{q} = [\theta \quad \alpha]^T$, and $\mathbf{Q}$ is the external forces, other matrices are defined as

$$\begin{aligned} \mathbf{M}(\mathbf{q}) &= \begin{bmatrix} \left(m_p l_2^2 s_2^2 + m_p r_2^2 + J_a\right) & \left(m_p l_2 r_2 c_2\right) \\ \left(m_p\ l_2 r_2 c_2\right) & \left(m_p l_2^2 + J_p\right) \end{bmatrix} \\ \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) &= \begin{bmatrix} m_p l_2^2 s_2 c_2\ \dot{\alpha} + C_a & m_p l_2^2 s_2 c_2\ \dot{\theta} - m_p l_2 r_2 s_2\, \dot{\alpha} \\ -m_p l_2^2 s_2 c_2 \dot{\theta} & C_p \end{bmatrix} \\ \mathbf{Q}_g(\mathbf{q}) &= \begin{bmatrix} 0 \\ m_p\, g\ l_2\, s_2 \end{bmatrix} \end{aligned}$$

It should be mentioned here, that the use of DH parameterization provides a structured and geometrically consistent method for modeling rigid-body kinematics in mechatronic systems. However, it assumes ideal link rigidity and precise joint configurations. It may not fully account for joint wear, or flexibility. The equations of motion, Eq.(12), can be solved for $\ddot{\mathbf{q}}$, and the vector of state variables can be constructed such that

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \tag{13}$$

thus, the time derivative of the state vector can be written as

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{X}}_1 \\ \dot{\mathbf{X}}_2 \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_2 \\ \mathbf{M}^{-1}\left(\mathbf{Q} - \mathbf{C}\dot{\mathbf{q}} - \mathbf{Q}_g\right) \end{bmatrix} \tag{14}$$

Equation (14) represents the state space equations of the system. Therefore, in the state space formulation, the second-order differential equations, i.e., Eq.(12) can be replaced by a system of first-order differential equations that has a number of equations equal to twice the number of generalized coordinates of the system. In the Direct Integration Method of equations of motion, at $t = t_0$, the initial conditions on $\mathbf{q}_0 = \mathbf{q}(t_0)$ and $\dot{\mathbf{q}}_0 = \dot{\mathbf{q}}(t_0)$ can be utilizied to start the integration process. The simulation of free falling pendulum, i.e., $\mathbf{Q} = \mathbf{0}$, is carried out and compared with the experimental measurements. The initial configuration are as follow: $\mathbf{q}_0 = [\theta_0 \quad \alpha_0]^T = [0 \quad 170°]^T$, the simulation results are shown in Figs.(2, 3).

For the inverted pendulum system, in which the external force vector $\mathbf{Q} = [\tau_1 \quad 0]^T$ is expressed interms of the control input, as

$$\mathbf{Q}=\begin{bmatrix}1\\0\end{bmatrix}\tau_1=\mathbf{b}u$$

where $u=\tau_1$, and $\mathbf{b}=[1 \quad 0]^T$. Thus, the acceleration vector $\ddot{\mathbf{q}}$, can be expressed as

$$\ddot{\mathbf{q}}=\mathbf{M}^{-1}\left(\mathbf{b}u-\mathbf{C}\dot{\mathbf{q}}-\mathbf{Q}_g\right) \tag{15}$$

Consequently, Eq.(14) can be written as

$$\dot{\mathbf{X}}=\mathbf{F}(\mathbf{X},\mathbf{u})=\begin{bmatrix}\mathbf{X}_2\\ \mathbf{M}^{-1}\left(\mathbf{b}u-\mathbf{C}\dot{\mathbf{q}}-\mathbf{Q}_g\right)\end{bmatrix}=\begin{bmatrix}\mathbf{F}_1\\ \mathbf{F}_2\end{bmatrix}\triangleq\begin{bmatrix}\dot{\mathbf{q}}\\ \ddot{\mathbf{q}}\end{bmatrix} \tag{16}$$

in which the state vector is defined as by Eq.(13).

The linearized form of the system equations, at ($\mathbf{X}=\mathbf{X}^*$, $\mathbf{u}=\mathbf{u}^*$), can be written as [9]

$$\dot{\mathbf{X}}=\mathbf{A}\ \Delta\mathbf{X}+\mathbf{B}\ \Delta\mathbf{u} \tag{17}$$

The state matrix $\mathbf{A}$, and the input matrix $\mathbf{B}$ can be obtained as

$$\mathbf{A}=\left(\frac{\partial\mathbf{F}}{\partial\mathbf{X}}\right)\Bigg|_{(\mathbf{X}^*,\mathbf{u}^*)}=\begin{bmatrix}\frac{\partial\mathbf{F}_1}{\partial\mathbf{X}_1} & \frac{\partial\mathbf{F}_1}{\partial\mathbf{X}_2}\\ \frac{\partial\mathbf{F}_2}{\partial\mathbf{X}_1} & \frac{\partial\mathbf{F}_2}{\partial\mathbf{X}_2}\end{bmatrix}=\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ \frac{\partial\ddot{\mathbf{q}}}{\partial\mathbf{q}} & \frac{\partial\ddot{\mathbf{q}}}{\partial\dot{\mathbf{q}}}\end{bmatrix} \tag{18}$$

$$\mathbf{B}=\left(\frac{\partial\mathbf{F}}{\partial\mathbf{u}}\right)\Bigg|_{(\mathbf{X}^*,\mathbf{u}^*)}=\begin{bmatrix}\frac{\partial\mathbf{F}_1}{\partial\mathbf{u}}\\ \frac{\partial\mathbf{F}_2}{\partial\mathbf{u}}\end{bmatrix}=\begin{bmatrix}\mathbf{0}\\ \mathbf{M}^{-1}\mathbf{b}\end{bmatrix} \tag{19}$$

The partial derivative of $\ddot{\mathbf{q}}$ with respect to $\mathbf{q}$, takes the form of

$$\frac{\partial\ddot{\mathbf{q}}}{\partial\mathbf{q}}=\mathbf{M}^{-1}\left(-\frac{\partial\mathbf{C}}{\partial\mathbf{q}}\dot{\mathbf{q}}-\frac{\partial\mathbf{Q_g}}{\partial\mathbf{q}}\right)+\frac{\partial\mathbf{M}^{-1}}{\partial\mathbf{q}}\left(\mathbf{b}u-\mathbf{C}\dot{\mathbf{q}}-\mathbf{Q}_g\right) \tag{20}$$

Similarly, the partial derivative of $\ddot{\mathbf{q}}$ with respect to $\dot{\mathbf{q}}$, estimated as

$$\frac{\partial\ddot{\mathbf{q}}}{\partial\dot{\mathbf{q}}}=\mathbf{M}^{-1}\left(-\frac{\partial\mathbf{C}\dot{\mathbf{q}}}{\partial\dot{\mathbf{q}}}-\frac{\partial\mathbf{Q_g}}{\partial\dot{\mathbf{q}}}\right)+\frac{\partial\mathbf{M}^{-1}}{\partial\ \dot{\mathbf{q}}}\left(\mathbf{b}u-\mathbf{C}\dot{\mathbf{q}}-\mathbf{Q}_g\right) \tag{21}$$

At some operating point, $\mathbf{q}=\mathbf{q}^*$, the velocity vector $\dot{\mathbf{q}}^*=\mathbf{0}$, consequently $\ddot{\mathbf{q}}^*\triangleq\mathbf{0}$. Thus, at this operating point, since $\mathbf{M}^{-1}\neq\mathbf{0}$ the term $\left(\mathbf{b}u-\mathbf{C}\dot{\mathbf{q}}-\mathbf{Q}_g\right)\triangleq\mathbf{0}$, see Eq. (15). Thus, Eqs. (20, 21) can be reduced to the following form

$$\frac{\partial\ddot{\mathbf{q}}}{\partial\mathbf{q}}\Bigg|_{(\dot{\mathbf{q}}^*=\mathbf{0},\ \ddot{\mathbf{q}}^*=\mathbf{0})}=-\mathbf{M}^{-1}\frac{\partial\mathbf{Q_g}}{\partial\mathbf{q}} \tag{22}$$

$$\frac{\partial\ddot{\mathbf{q}}}{\partial\dot{\mathbf{q}}}\Bigg|_{(\dot{\mathbf{q}}^*=\mathbf{0},\ \ \ddot{\mathbf{q}}^*=\mathbf{0})}=-\mathbf{M}^{-1}\mathbf{C} \tag{23}$$

Substituting Eqs. (22, 23) into Eqs. (18, 19), yields:

$$\mathbf{A}=\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{M}^{-1}\frac{\partial\mathbf{Q_g}}{\partial\mathbf{q}} & -\mathbf{M}^{-1}\mathbf{C}\end{bmatrix},\quad \mathbf{B}=\begin{bmatrix}\mathbf{0}\\ \mathbf{M}^{-1}\mathbf{b}\end{bmatrix}$$

1) Validation of the Linearized Model: Let us examine the linearized model at operating point, $\mathbf{q}^*=[\theta \quad \alpha]^{\mathbf{T}}=\mathbf{0}$, $\dot{\mathbf{q}}^*=\mathbf{0}$, and $\ddot{\mathbf{q}}^*=\mathbf{0}$. The numerical simulation is carried out using LabVIEW and presented in Figs.(4, 5). These figures show the simulation of a free-falling pendulum with an initial configuration of $\mathbf{q}_0=[\theta_0 \quad \alpha_0]^T=[0 \quad 10^\circ]^T$ and demonstrate acceptable agreement between the linear and nonlinear models. In contrast, a separate simulation using an initial angular velocity of $\alpha_0=20^\circ$ exhibits a clear divergence between the two models. This divergence increases as the initial configuration deviates further from the operating point used for linearization.

2) Inclusion of actuator dynamics: In the actual physical system the control input is not the torque, but is actually the voltage applied to the motor. The actuator dynamics must be included into the state equations since the computer does not control the motor torque directly but controls the voltage being applied to the motor. The torque generated at the arm pivot from the motor voltage, V_m, is given by

$$\tau_1=\frac{K_t}{R_m}\left(V_m-K_m\dot{\theta}\right) \tag{24}$$

The linear dynamics of the motor, Eq. (24) can be combined with the differential equation of the pendulum, and consequently, the modified state matrices take the form of:

$$\mathbf{A}=$$

$$\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & \frac{r_2\,m_p^2 g\ell_2^2}{J_a m_p\ell_2^2+J_p m_p r_2^2+J_a\,J_p} & -\frac{K_m K_t\left(m_p\ell_2^2+J_p\right)}{R_m\left(\left(m_p\ell_2^2+J_p\right)J_a+J_p\,m_p\,r_2^2\right)} & 0\\ 0 & +\frac{\left(m_p r_2^2+J_a\right)m_p\,g\ell_2}{J_a\,m_p\ell_2^2+J_p\,m_p r_2^2+J_a\,J_p} & -\frac{K_m K_t\left(m_p\ell_2^2+J_p\right)}{R_m\left(J_a m_p\ell_2^2+J_p\,m_p r_2^2+J_a\,J_p\right)} & 0\end{bmatrix},$$

$$\mathbf{B}=\begin{bmatrix}0\\ 0\\ \frac{K_t\left(m_p\ell_2^2+J_p\right)}{R_m\left(\left(m_p\ell_2^2+J_p\right)J_a+J_p\,m_p\,r_2^2\right)}\\ -\frac{K_t\ \ell_2\,m_p\,r_2}{R_m\left(J_a m_p\ell_2^2+J_p\,m_p r_2^2+J_a\,J_p\right)}\end{bmatrix} \tag{25}$$

III. STATE FEEDBACK CONTROL

To regulate the pendulum near the upright equilibrium, a state feedback controller based on the Linear Quadratic Regulator (LQR) method is designed. The controller is derived from a linearized state-space model of the nonlinear rotary inverted pendulum system. The control objective is to find an optimal gain matrix K such that the control law:

$$u(t)=-Kx(t) \tag{26}$$

minimizes the quadratic cost function:

$$J=\int_0^\infty\left(x(t)^\top Qx(t)+u(t)^\top Ru(t)\right)dt \tag{27}$$

where $Q\geq 0$ and $R>0$ are symmetric weighting matrices that penalize state deviations and control effort, respectively. The LQR gain matrix is structured to include feedback from both angular position and angular velocity of the pendulum and the arm:

$$K=[k_{p,\theta} \quad k_{p,\alpha} \quad k_{v,\theta} \quad k_{v,\alpha}]^T \tag{28}$$

The control input V_m that drives the DC motor is then computed as

$$V_m=k_{p,\theta}\cdot\theta+k_{p,\alpha}\cdot\alpha+k_{v,\theta}\cdot\dot{\theta}+k_{v,\alpha}\cdot\dot{\alpha} \tag{29}$$

Implementation Using LabVIEW

The state-space model and LQR controller are implemented using NI LabVIEW Control Design and Simulation Toolkit, see Fig.(6). The toolkit is used to:

Import the linearized model $\longrightarrow$ Specify the Q and R matrices $\longrightarrow$ Automatically compute the optimal gain matrix K $\longrightarrow$Simulate the closed-loop response.

The optimized gain values obtained via LQR tuning are

$$K=\left[-6.5\left[\frac{V}{rad}\right] \quad 80\left[\frac{V}{rad}\right] \quad -2.75\left[\frac{V}{rad/s}\right] \quad 10.5\left[\frac{V}{rad/s}\right]\right]^T \tag{30}$$

These values are used in the control law implemented in LabVIEW and tested in real-time using the NI ELVIS II platform.

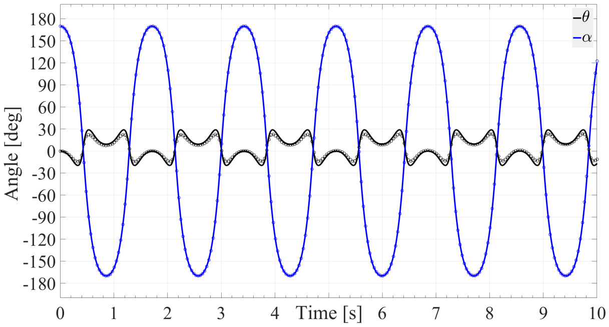


Fig. 2. Angular displacements: —— (Nonlinear model), ○ (Experimental)

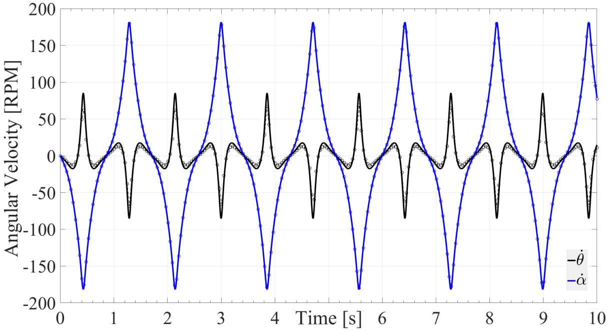


Fig. 3. Angular velocities: —— (Nonlinear model), ○ (Experimental)

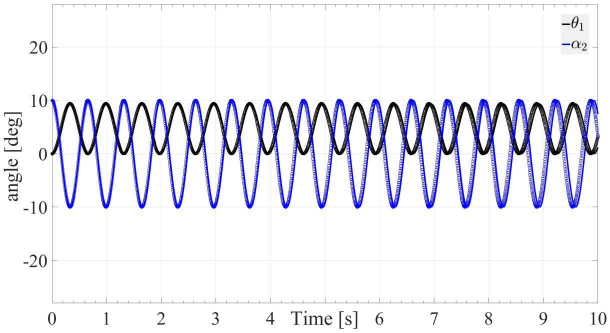


Fig. 4. Angular displacements: —— (Nonlinear model), ○ (Linear)

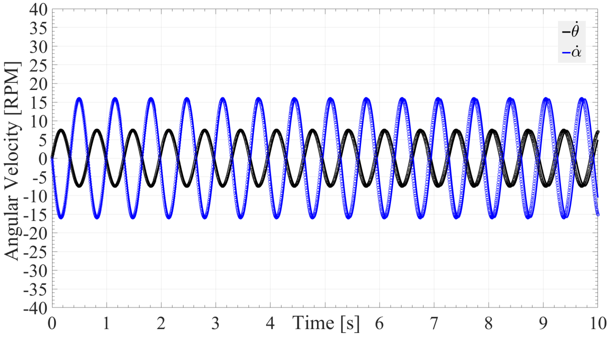


Fig. 5. Angular velocities: —— (Nonlinear model), ○ (Linear)

IV. Fuzzy Logic Controller Design

In this work, a hybrid fuzzy-state feedback controller is developed for stabilization of the rotary inverted pendulum. The hybrid nature of the controller lies in integrating a conventional state-feedback control law derived via LQR design with a fuzzy logic system that adaptively tunes the proportional gain of the pendulum angle, $k_{p,\alpha}$, during the stabilization phase.

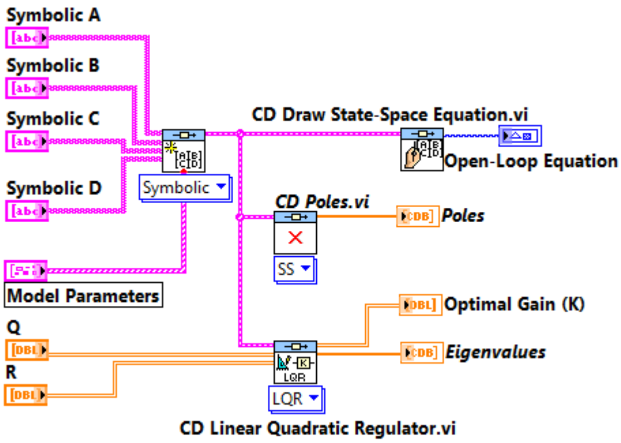


Fig. 6. Labview Block Diagram of SF-Control Design using LQR

Hybrid Control Concept

While the classical LQR controller offers optimal performance under linearized conditions, its performance can degrade under nonlinear dynamics or when the system deviates from its equilibrium point. To address this, we introduce a fuzzy logic layer that dynamically adjusts the proportional gain $k_{p,\alpha}$ based on real-time measurements of the pendulum angle α and its angular velocity $\dot{\alpha}$. This results in the following hybrid control law:

$$V_m = k_{p,\theta}\cdot\theta + k_{p,\alpha}(\alpha,\dot{\alpha}) \nearrow \cdot\alpha + k_{v,\theta}\cdot\dot{\theta} + k_{v,\alpha}\cdot\dot{\alpha} \qquad (31)$$

Here, $k_{p,\alpha}(\alpha,\dot{\alpha})$ is a gain value computed by the fuzzy inference system, while all other gains are fixed and pre-tuned through LQR.

Fuzzy Controller Inputs and Output

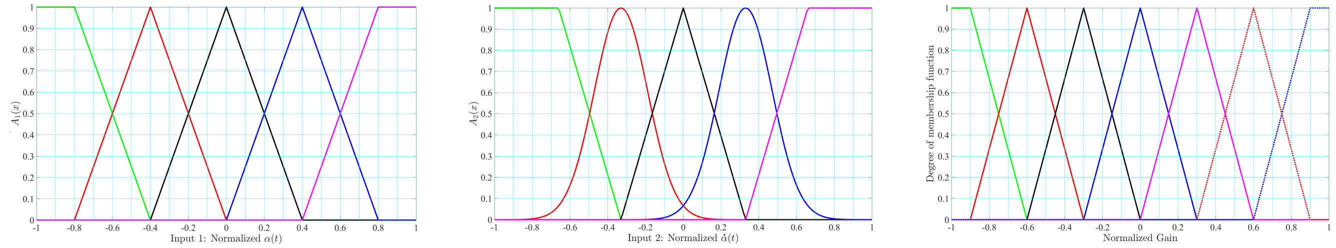
The fuzzy controller takes two inputs: the pendulum angle (range: $-15^\circ \leq \alpha \leq 15^\circ$), pendulum angular velocity ($\dot{\alpha}$: $(-5,5)$ $[rad/s]$) and the output is $k_{p,\alpha}$: the proportional gain used to amplify the pendulum angle contribution in the control law. The Membership Functions (MFs) for both input and output variables are uniformly distributed over their respective ranges, as shown in Figs. (7).

Since the controller is focused on maintaining balance around an operating point, the Gaussian membership function is key in the central region because of its smooth transitions and high sensitivity, which ensures precise control for small angular velocity deviations [10].

The fuzzy logic system operates using a set of 25 intuitive if-then rules, summarized in Table(II). Each of the 25 rules captures a unique combination of angle and velocity linguistic labels, mapping them to one of five control gain levels. This granular approach ensures that subtle differences in the pendulum's state are reflected in the control action [9].

The principle is that the sign of α relative to the sign of $\dot{\alpha}$ reveals critical information about the pendulum's motion, specifically:

- Same Sign ($\alpha\cdot\dot{\alpha} > 0$): When the angle and angular velocity share the same sign, the pendulum is moving away from its equilibrium point, indicating continued divergence. Under these circumstances, the control gain



(a) MFs for pendulum angle normalized between $[-0.25, 0.25]$ rad

(b) MFs for pendulum angular velocity normalized between $[-5, 5]$ rad/s

(c) MFs normalized between $[0, 250]$ V/rad

Fig. 7. Membership functions (MFs) for input and output variables.

TABLE II

FUZZY RULE BASE FOR $k_{p,\alpha}$: $(-\Downarrow)$ NEGATIVE BIG, $(-\downarrow)$ NEGATIVE SMALL, $(\longleftrightarrow)$ ZERO, $(+\uparrow)$ POSITIVE SMALL, $(+\Uparrow)$ POSITIVE BIG.

$\alpha/\dot{\alpha}$	$-\Downarrow$	$-\downarrow$	$\longleftrightarrow$	$+\uparrow$	$+\Uparrow$
$-\Downarrow$	Low	Low	Low	Med	High
$-\downarrow$	Low	Low	Med	High	High
$\longleftrightarrow$	Low	Med	High	Med	High
$+\uparrow$	Med	High	Med	Med	Low
$+\Uparrow$	High	High	High	Low	Low

is reduced to prevent an overly aggressive correction that could exacerbate instability.

- Opposite Sign ($\alpha \cdot \dot{\alpha} < 0$): When the angle and angular velocity have opposite signs, the pendulum is undergoing a self-correcting motion toward the equilibrium. In this case, the control gain is increased to reinforce this stabilizing behavior and speed up convergence.

These rules result in a dynamic adaptation of the gain, improving system responsiveness and robustness under nonlinear behavior and external disturbances.

This work employs a Mamdani-type fuzzy inference system with centroid defuzzification, selected for its intuitive rule-based structure and seamless integration with LabVIEW's Fuzzy System toolkit. Gaussian membership functions are used near the equilibrium point to provide smooth transitions and high sensitivity, while triangular and trapezoidal functions are applied in the outer regions to handle larger deviations efficiently. This hybrid MF design balances precision and stability, offering fine control near the setpoint and robustness during transients. Offline tuning and sensitivity analysis confirmed that MF shapes and overlaps significantly impact system behavior, particularly in terms of overshoot, settling time, and control smoothness.

V. REAL TIME IMPLEMENTATION

In order to create the fuzzy controllers, the LabVIEW Fuzzy System Designer was used. Figure (8) shows the LabVIEW code for the fuzzy balancing controller. The system created using the Fuzzy System Designer was loaded, and then passed through to the FL Fuzzy Controller VI. Then the output from the Fuzzy Controller VI, $Kp_{alpha_{FLC}}$, is joined with the other balance state-feedback parameters, $k_{d,\alpha}$, $k_{p,\theta}$, and $k_{d,\theta}$, and then fed to the `Gain` block.

In this study, two control strategies are compared: In figures (9-13): the state feedback controller within LQR optimization (represented in black) and the hybrid state feedback and fuzzy controller (represented in blue). A disturbance is introduced to the system at different times: at $t = 12$ seconds for the state feedback controller and at $t = 16$ seconds for the hybrid controller. The state feedback controller exhibits a pronounced transient response when the disturbance is applied, with significant overshoot and oscillatory behavior as it works to restore equilibrium. In contrast, the hybrid controller, which incorporates an adaptive fuzzy logic component, responds to the disturbance with a smoother, more controlled correction. This adaptive mechanism continuously adjusts the control gains in real time, leading to reduced overshoot, diminished oscillations, and a faster settling time. These observations clearly demonstrate the superior performance of the hybrid controller over the traditional state feedback approach in mitigating the effects of external disturbances.

The hybrid SF-FLC controller demonstrated superior performance in balancing the inverted pendulum compared to the conventional SF-LQR controller. It achieved a faster settling time more than twice as fast as well as a smaller steady-state error, approximately half that of the SF controller.

These results highlight the effectiveness of replacing the fixed proportional gain with a fuzzy logic-based adaptive gain. The design enhances both precision and response speed during stabilization, validating the hybrid controller advantage for balancing fast mechatronic systems.

VI. CONCLUSION AND FUTURE WORK

This paper presented a hybrid fuzzy-state feedback controller for fast mechatronic systems, demonstrated on a rotary inverted pendulum. The controller integrates a linear quadratic regulator with a fuzzy logic system that adaptively tunes the proportional gain for the pendulum angle. Both simulation and experimental results on an NI ELVIS II platform show that the hybrid controller achieves a settling time that is more than twice as fast and a steady-state error that is approximately half of that obtained with the traditional state feedback controller. The experimental data confirm that the adaptive fuzzy tuning reduces overshoot and dampens oscillations in response to external disturbances, thereby enhancing the system's overall robustness and stability. Future work will explore the extension of the adaptive strategy to additional feedback gains, including those associated with angular velocity, to further improve performance. The proposed hybrid control framework is expected to be applicable to a wide range of fast-response mechatronic systems, and further studies will investigate its performance in different operating conditions and application scenarios.

The proposed hybrid control framework, validated on a rotary inverted pendulum, exhibits a twofold improvement in settling time. Its modular design makes it adaptable to a broader class of fast-response mechatronic systems. Nevertheless, the scalability of performance gains depends

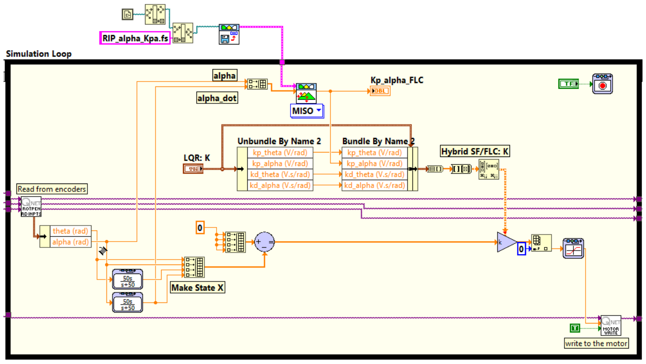


Fig. 8. LabVIEW block diagram of SF-FLC real time implementation

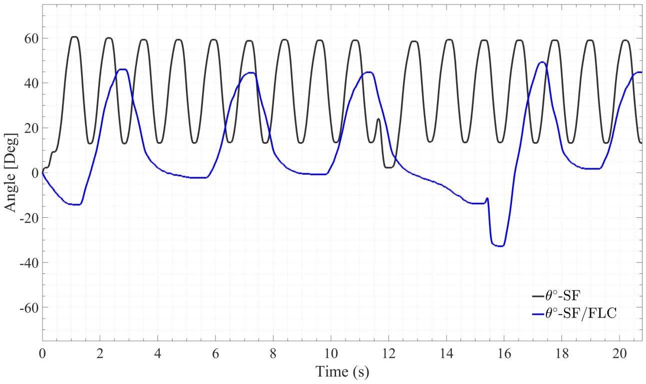


Fig. 9. Arm Angle: ━ (SF-LQR), ━ (SF-FLC)

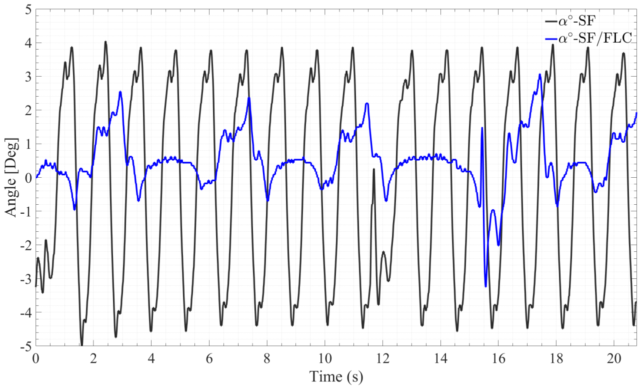


Fig. 10. Pendulum Angle: ━ (SF-LQR), ━ (SF-FLC)

on factors such as system dimensionality, actuator dynamics, and feedback loop latency. In high-dimensional systems, the complexity of tuning and rule base management may increase substantially. This could necessitate the use of hierarchical fuzzy structures or modular control schemes to maintain computational feasibility and design tractability.

References

[1] A. Nada and M. Bayoumi, "Development of a constraint stabilization method of multibody systems based on fuzzy logic control," *Multibody System Dynamics*, 2023.

[2] S. Pati, M. Patnaik, and A. Panda, "Comparative performance analysis of fuzzy pi, pd and pid controllers used in a scalar controlled induction motor drive," *2014 International Conference on Circuits, Power and Computing Technologies, ICCPCT 2014*, pp. 910–915, 3 2014.

[3] D. Somwanshi, M. Bundele, G. Kumar, and G. Parashar, "Comparison of fuzzy-pid and pid controller for speed control of dc motor using labview," *Procedia Computer Science*, vol. 152, pp. 252–260, 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050919306702

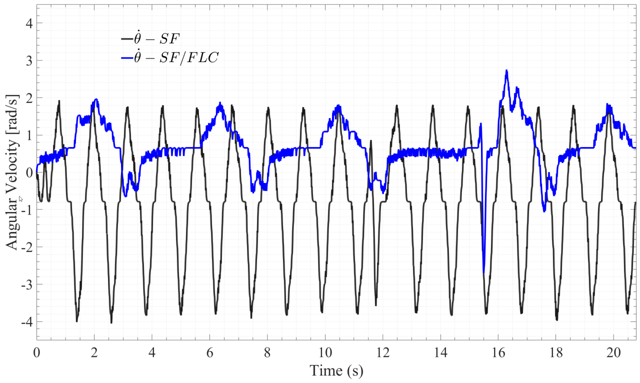


Fig. 11. Arm Angular velocity: ━ (SF-LQR), ━ (SF-FLC)

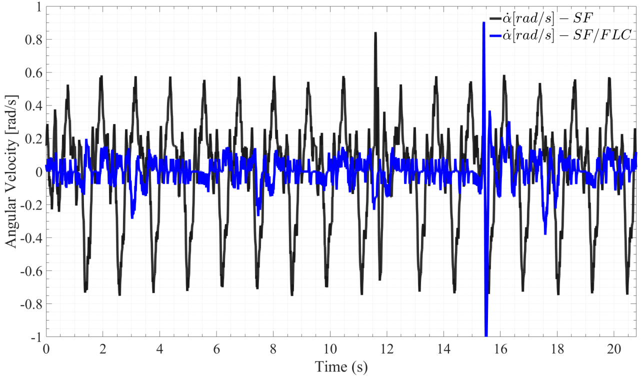


Fig. 12. Pendulum Angular velocity: ━ (SF-LQR), ━ (SF-FLC)

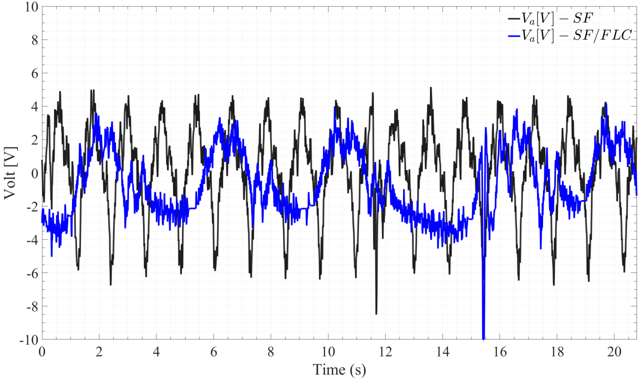


Fig. 13. Input Voltage: ━ (SF-LQR), ━ (SF-FLC)

[4] A. A. Nada and M. A. Bayoumi, "Development of embedded fuzzy control using reconfigurable fpga technology," *Automatika*, vol. 65, no. 2, pp. 609–626, 2024, published online: 14 Feb 2024. [Online]. Available: https://doi.org/10.1080/00051144.2024.2313904

[5] J. C. García-Montalva, J. de la Cruz-Alejo, and J. Díaz-Salgado, "Fuzzy logic control on fpga using labview," *Mechanisms and Machine Science*, vol. 25, pp. 261–271, 2015. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-09858-6_25

[6] S. M. Megahed, *Principles of Robot Modelling and Simulation*. Chichester, New York: John Wiley & Sons, 1993.

[7] L. Wang, *A Course in Fuzzy Systems and Control*. Prentice Hall, 1997.

[8] A. A. Nada, "Simplified procedure of sensitivity-based parameter estimation of multibody systems with experimental validation," *IFAC-PapersOnLine*, vol. 54, no. 14, pp. 84–89, 2021, 3rd IFAC Conference on Modelling, Identification and Control of Nonlinear Systems MICNON 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896321017389

[9] K. C. Sio and C. K. Lee, "Stability of fuzzy pid controllers," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, vol. 28, pp. 490–495, 1998.

[10] S. V. Wong and A. M. Hamouda, "Optimization of fuzzy rules design using genetic algorithm," *Advances in Engineering Software*, vol. 31, pp. 251–262, 4 2000.