Evaluating PID, LQR, and LMPC on Furuta Pendulum with External Disturbance Considerations

Husnul Amri
Faculty of Engineering
Tanjungpura University
Pontianak, Indonesia
amrihusnul@ee.untan.ac.id

Ferry Hadary
Faculty of Engineering
Tanjungpura University
Pontianak, Indonesia
ferry.hadary@ee.untan.ac.id

***Abstract*—The Furuta pendulum is a well-known underactuated and naturally unstable system, which makes stabilization methods very interesting to discuss. This work provides a performance comparison of three different types of controllers, which are proportional-integral-derivative (PID), linear quadratic regulator (LQR), and linear model predictive control (LMPC), on a Furuta pendulum system with external disturbance consideration. From the simulation results, the stabilizing controller was activated after the swing-up process reached its switching threshold. The controller can keep the Furuta pendulum system around its equilibrium states, and bring the system back after a disturbance injection. Each of the controllers has its own advantages and disadvantages evaluated using three performance index parameters, which are integral of square control input (ISCI), integral of time-weighted absolute error (ITAE), and settling time. In the overall performance evaluation, LMPC has the best and reasonable response both in the stabilization and disturbance rejection phase.**

Keywords—Furuta pendulum, PID, LQR, LMPC, Index Performance

I. INTRODUCTION

The inverted pendulum has been a challenging issue for the control research community for decades. It is well known that the system is fully underactuated and naturally unstable. The system is also extremely nonlinear due to the gravitational effects, making stabilization methods interesting to discuss. There's an inverted pendulum problem that has been addressed by prior research, such as wheel-based cart-inverted pendulum system [1], monorail-based cart-inverted pendulum system [2], and rotary inverted pendulum system, also well known as the Furuta pendulum [3,4].

The Furuta pendulum offers an interesting design that avoids the limited travel distance problem that occurs in the cart-based design. Several works have been done to stabilize the system at the equilibrium states. Some relevant works focus on the initial movements to bring the pendulum near the equilibrium states or swing-up strategy [3,5,6], and others focus on the stabilization process.

Various controller types have been introduced to the Furuta pendulum system to stabilize it upright. Some examples are implementation of pole placement control methods [7], proportional-integral-derivative (PID) control [8], linear quadratic regulator (LQR) control [9,10], and many more, including robust and learning-based controls [11,12]. While implementing a single type of controller could demonstrate its ability to stabilize the Furuta pendulum, comparing different types of controllers in a typical environment was also interesting, since the controllers can be evaluated side-by-side to see which aspect needs to be improved.

Some prior works on different control type implementations for the Furuta pendulum were done, for example, in [8,13–15], where Bang Bang-PID-Cascade PID, PID-LQR, LQR - model predictive control (MPC), and PID-LQR with disturbance were compared to evaluate the different types of control methods for the Furuta pendulum. Mainly resulted in better results in cascade PID, LQR, or the MPC control methods. However, it is also interesting to see how a classical control like PID behaves on the Furuta pendulum, followed by an optimal control LQR, and then also compared with an optimal model-based optimization control using Linear MPC (LMPC). Most similar works with this type of focus are done in [16]; however, it has a different kind of actuated system (cart - inverted pendulum system).

Therefore, this paper mainly contributes to providing a performance comparison of three different types of controllers (PID, LQR, and LMPC) on a Furuta pendulum system. Furthermore, external disturbance through the pendulum arm is also introduced to see how different types of controllers react to an unexpected disturbance. Finally, a numerical simulation with an index performance evaluation will be provided to evaluate each controller's performance.

II. SYSTEM MODELLING

A. Non-linear Model of Furuta Pendulum

Consider a schematic image of a Furuta pendulum shown in Figure 1. In general, the system consists of two moving parts, each of which is connected to an inertial body. The first part is the central actuator pillar with a moment of inertia J that connects rigidly to a horizontal arm with length l_a and has a homogenously distributed mass m_a along the arm. The first part connects directly and rigidly to the pendulum arm with length l_p and homogenously distributed mass m_p via the end of the horizontal arm. The pendulum arm has an attached point mass M at its end. The actuator arm and pendulum arm angle positions are represented by ϕ and θ, respectively.

The nonlinear model of this system can be found in [17], which is given by:

$$\frac{d}{dt}\phi = \dot{\phi}$$
$$\frac{d}{dt}\theta = \dot{\theta}$$
$$\frac{d}{dt}\begin{pmatrix}\dot{\phi}\\ \dot{\theta}\end{pmatrix} = \boldsymbol{D}^{-1}(\phi,\theta)\left(\boldsymbol{\tau}(\phi,\theta) - \boldsymbol{C}(\phi,\theta,\dot{\phi},\dot{\theta})\begin{pmatrix}\dot{\phi}\\ \dot{\theta}\end{pmatrix} - \boldsymbol{\mathcal{G}}(\phi,\theta)\right) \quad (1)$$

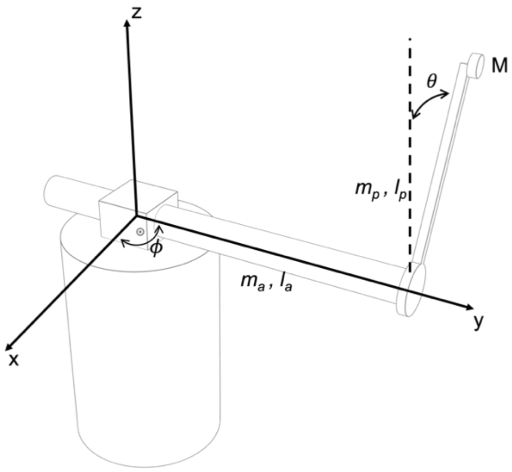

Fig. 1. Furuta Pendulum System

Where $\dot{\phi}$ and $\dot{\theta}$ is the center arm and the pendulum arm rotation speed, respectively, and $\boldsymbol{\tau}(\phi,\theta) = [\tau_\phi, \tau_\theta]^T$ is the force that is applied to the Furuta pendulum system in terms of ϕ and θ. The other parameters ***D***, ***C*** and $\boldsymbol{\mathcal{G}}$ are described by:

$$\boldsymbol{D}(\phi,\theta) \triangleq \begin{pmatrix} \alpha + \beta \sin^2\theta & \gamma \cos\theta \\ \gamma \cos\theta & \beta \end{pmatrix}$$
$$\boldsymbol{C}(\phi,\theta,\dot{\phi},\dot{\theta}) \triangleq \begin{pmatrix} \beta cos\theta sin\theta\dot{\theta} & \beta cos\theta sin\theta\dot{\phi} - \gamma sin\theta\dot{\theta} \\ -\beta cos\theta sin\theta\dot{\phi} & 0 \end{pmatrix}$$
$$\boldsymbol{\mathcal{G}}(\phi,\theta) \triangleq \begin{pmatrix} 0 \\ -\delta \sin\theta \end{pmatrix} \qquad (2)$$

α, β, γ and δ is a form of constant parameter simplification that is defined by:

$$\alpha \triangleq J + \left(M + \frac{1}{3}m_a + m_p\right) l_a^2$$
$$\beta \triangleq \left(M + \frac{1}{3}m_p\right) l_p^2$$
$$\gamma \triangleq \left(M + \frac{1}{2}m_p\right) l_a l_p$$
$$\delta \triangleq \left(M + \frac{1}{2}m_p\right) g l_p \qquad (3)$$

With g denotes the earth gravitational force. The initial non-linear model still does not accommodate the friction term. Therefore, in this paper, the viscous friction term that comes from the main and pendulum joint is considered. Let us represent the viscous friction term for θ and ϕ in this form [17]:

$$\tau_{F,\phi} = \eta_{v,\phi}\dot{\phi}$$
$$\tau_{F,\theta} = \eta_{v,\theta}\dot{\theta} \qquad (4)$$

where $\tau_{F,\phi}$ and $\tau_{F,\theta}$ stand for viscous friction force for ϕ and θ, respectively. Then, $\eta_{v,\phi}$ and $\eta_{v,\theta}$ is the friction constant for ϕ and θ. Inserting the viscous friction factor to the third term of equation (1) yields the new nonlinear equation of the Furuta pendulum system.

$$\frac{d}{dt}\phi = \dot{\phi}$$
$$\frac{d}{dt}\theta = \dot{\theta}$$
$$\frac{d}{dt}\begin{pmatrix}\dot{\phi} \\ \dot{\theta}\end{pmatrix} = \boldsymbol{D}^{-1}(\phi,\theta)\left(\boldsymbol{\tau}(\phi,\theta) - \boldsymbol{C}(\phi,\theta,\dot{\phi},\dot{\theta})\begin{pmatrix}\dot{\phi} \\ \dot{\theta}\end{pmatrix} - \boldsymbol{\mathcal{G}}(\phi,\theta) - \boldsymbol{\tau_F}(\dot{\phi},\dot{\theta})\right) \qquad (5)$$

With $\boldsymbol{\tau_F}(\phi,\theta) = [\tau_{F,\phi}, \tau_{F,\theta}]^T$.

B. Linearization

Linear-type controllers (LQR and LMPC) will be used as the stabilizing controller in this paper. To use this controller, linearization of the non-linear model around the equilibrium state is needed. Initially, we introduce the state variable of the system as:

$$\boldsymbol{x} \triangleq [\phi, \dot{\phi}, \theta, \dot{\theta}]^T \qquad (6)$$

The state equation is basically the time derivation of the state $\boldsymbol{x}$ in the form of: $d\boldsymbol{x}/dt = f(\boldsymbol{x},\boldsymbol{\tau})$. Expanding the state equation using the Taylor series at the equilibrium points ($\boldsymbol{x_0} = [\phi_0, \dot{\phi_0}, \theta_0, \dot{\theta_0}]^T$ and $\boldsymbol{\tau_0} = [0,0]^T$) yields:

$$\frac{d(\boldsymbol{\delta x})}{dt} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_0 \boldsymbol{\delta}\mathbf{x} + \left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{\tau}}\right|_0 \boldsymbol{\tau} \triangleq \boldsymbol{A}\delta x + \boldsymbol{B}\tau \qquad (7)$$

Where δx is defined by $\boldsymbol{\delta x} = \boldsymbol{x} - \boldsymbol{x_0}$. Solving the Jacobian terms for (5) with respect to $\boldsymbol{x}$ and $\boldsymbol{\tau}$ for upward position $\boldsymbol{x_0} = [0,0,0,0]^T$ will give us the $\boldsymbol{A} \in \mathbb{R}^{r\times r}$ and $\boldsymbol{B} \in \mathbb{R}^{s\times s}$ matrix in the form of:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-\beta\tau_{v,\phi}}{\alpha\beta - \gamma^2} & \frac{-\delta\gamma}{\alpha\beta - \gamma^2} & \frac{\gamma\tau_{v,\theta}}{\alpha\beta - \gamma^2} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{\gamma\tau_{v,\phi}}{\alpha\beta - \gamma^2} & \frac{\alpha\delta}{\alpha\beta - \gamma^2} & \frac{\alpha\tau_{v,\theta}}{\alpha\beta - \gamma^2} \end{bmatrix}$$
$$\boldsymbol{B} = \begin{bmatrix} 0 \\ \frac{\beta}{\alpha\beta - \gamma^2} \\ 0 \\ \frac{-\gamma}{\alpha\beta - \gamma^2} \end{bmatrix} \qquad (8)$$

Controllability of the system is mandatory since the LQR control requires a controllable system to have a unique positive definite solution of the Ricatti equation. Controllability is determined based on the rank calculation using the formula $N_c = rank[\boldsymbol{B}\ \boldsymbol{AB}\ \boldsymbol{A}^2\boldsymbol{B}\ \boldsymbol{A}^3\boldsymbol{B}]$. Considering the ***A*** and ***B*** matrix shown in the equation (8), and by using MATLAB rank function, the resulting value of N_c is 4, showing that the system is controllable and satisfies the controller's requirement. The non-linear equation (5) and linearized state (8) can then be used for designing the controller that will be used in this work, which is PID, LQR, and LMPC.

III. CONTROL ALGORITHM

A. PID Control

Proportional-Integral-Derivative (PID) control is a form of classical control method for a wide range of applications. This type of controller can be applied to both a non-linear and linear system, and has a lower implementation complexity, making it a fast and popular choice in the control system research. This control is basically giving an input signal based on the deviation of the process value to the target set point (error signal, e). In this work, PID control will be used to control the arm angle (ϕ) by using the pendulum angle error information ($e_\theta = \theta_t - \theta_g$) where θ_t is the value of the θ on time step t and θ_g is the value of the θ target. The general form of the PID control equation is:

$$\tau_{\phi,PID} = K_p e_\theta(t) + K_i \int e_\theta(t)\,\mathrm{dt} + K_d \frac{de\theta(t)}{dt} \quad (9)$$

Where the K_p, K_i and K_d is the proportional, integral and derivative gain parameter, respectively. Those gains in value are manually tuned in this work, which works with the primary objective of achieving the desired pendulum angle (θ_g as quickly as possible. In the controlling process, often PID will face some accumulating error issues in the integral error terms, making the controller unstable for an extended period of time. To overcome this issue, a simple form of anti-wind-up integral error using conditional integration as discussed in [18] is implemented in this work.

B. LQR Control

LQR control is part of an optimal control which optimize a performance index shown L in the form of:

$$L = \frac{1}{2}\int_0^\infty \left(\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \tau_{\boldsymbol{\phi},\mathbf{LQR}}^{\boldsymbol{T}} \boldsymbol{R}\, \tau_{\boldsymbol{\phi},\mathbf{LQR}}\right) dt \quad (10)$$

With $\boldsymbol{Q} \in \mathbb{S}_+^r$ and $\mathbf{R} \in \mathbb{S}_{++}^s$ are symmetric, positive semi-definite matrix parameters that used to determine the relative weights of the state variables and inputs. To optimize those values, an appropriate value of $\tau_{\boldsymbol{\phi},\boldsymbol{LQR}}$ is needed. This value is determined based on the state feedback matrix calculations shown in (11).

$$\tau_{\boldsymbol{\phi},\boldsymbol{LQR}} = -\mathbf{K}\,\mathbf{x} \quad (11)$$

Where $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{P}$, with $\mathbf{P} \in \mathbb{S}_+^r$ represents the solution of the algebraic Riccati equation [19] in the form of:

$$0 = \mathbf{PA} + \mathbf{A}^{\mathrm{T}}\mathbf{P} - \mathbf{PBR}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P} + \mathbf{Q} \quad (12)$$

The value $\mathbf{K}$ significantly depends on the combination of $\mathbf{Q}$ and $\mathbf{R}$ matrix. Several studies have been conducted to determine those values based on trial and error, or another optimization-based method [20]. Selection of $\mathbf{Q}$ and $\mathbf{R}$ matrix is also often done based on the response time from swing up to the stabilization process [19], which is also used in this work to keep simplicity.

C. MPC Control

Model Predictive Control (MPC) is one of the modern model-based optimization methods to control a linear or nonlinear system. Using this type of controller, the system model, along with the current state, will be used to predict the future behavior of the system for a short period of time (receding horizon predictions). A cost function will evaluate the prediction, and then the appropriate control input sequence will be chosen based on an optimization algorithm. First, the optimal input will then apply to the real plant/system, and then the algorithm will continue to calculate the next movement prediction of the system, and so on.

Generally, to bring any system to a referenced point, the set point tracking Linear MPC (LMPC) algorithm can be effectively used to generate an optimal control sequence. However, in this work, the Furuta pendulum system is targeted to settle at the equilibrium with zero angle position ($\phi_g = \dot{\phi}_g = \theta_g = \dot{\theta}_g = 0$), therefore, the set point tracking will fall back to a traditional LMPC algorithm.

LMPC is implemented on a discrete optimization over N a horizon step by solving this optimization problem [21]:

$$\begin{aligned} &\inf_{\tau_{\phi,LMPC,i}} \; H_i \\ &\text{subject to: } \tau_{\phi,LMPC,i} \in \mathbb{U}^N, \boldsymbol{X}_i \in \mathbb{X}^{N+1}, \\ &\qquad \mathbf{x}_{h+1,\mathrm{i}} = \mathbf{A}\mathbf{x}_{h,i} + \mathbf{B}\tau_{\phi,LMPC,h,i}, i \in \mathbb{T}, \\ &\qquad \mathbb{T} \in \{0,1,\ldots,\mathrm{N}-1\} \end{aligned} \quad (13)$$

Where $\tau_{\phi,LMPC}$ and X_i is a set of input control and state sequence over the time horizon step i, where a minimum control will be evaluated in every prediction horizon step $h \in \{\mathrm{i},\ldots,\mathrm{i}+\mathrm{N}-1\}\mathbb{U} = \tau_{\phi,LMPC} \in \mathbb{R}^s\colon \boldsymbol{F}_\tau \tau_{\phi,LMPC} \le \boldsymbol{g}_\tau$ and $\mathbb{X} = \mathbf{x} \in \mathbb{R}^{\mathrm{r}}\colon \mathbf{F}_{\mathrm{x}}\mathbf{x} \le \mathbf{g}_{\mathrm{x}}$ are the constraint sets which basically states that every state $\boldsymbol{x}$ and input $\tau_{\phi,LMPC}$ multiplied by the constrained state and input matrices $\boldsymbol{F}_x$, $\boldsymbol{F}_\tau$ should be less than the constrained tracking matrices $\boldsymbol{g}_x$ and $\boldsymbol{g}_u$. The cost function H_i, which is optimized by the equation (13), is a form of quadratic sum of the states and control inputs stated in the equation (14).

$$\begin{aligned} H_i = \mathbf{x}_{i+N,i}^T \mathbf{Q}_{\mathrm{N}} \mathbf{x}_{i+N,i} &+ \sum_{h=i}^{h+N-1} \boldsymbol{x}_{h,i}^T \boldsymbol{Q}_f \boldsymbol{x}_{h,i} \\ &+ \tau_{\phi,LMPC,h,i}^T \boldsymbol{R}_N \tau_{\phi,LMPC,h,i} \end{aligned} \quad (14)$$

Where $\boldsymbol{Q_N} \in \mathbb{R}^{r\times r}, \boldsymbol{Q_f} \in \mathbb{R}^{r\times r}, \boldsymbol{R}_N \in \mathbb{R}^{s\times s}$ are the weighting that is used to penalize the tracking state, error at the end of prediction horizon and the control effort, respectively. Further step-by-step numerical methods to solve the optimization problem in (13) are not the scope of this paper, and we directly use the same step that is used in [21].

It is worth mentioning that the problem (13) assumes that the state space equation used is in the form of a discrete-time. Therefore, in this work, the state A B is initially converted from continuous to discrete using zero-order-hold methods [22]. The conversion process is done using the MATLAB algorithm *c2d*.

D. Disturbance Modelling

During stabilization periods, every control system algorithm has its own sensitivity against external disturbance. Therefore, exact same value of disturbance applied to the system will give a different deviation effect.

In this work, each controller is subjected to a disturbance torque adjusted to achieve the same pendulum angular

TABLE I. SIMULATION PARAMETERS

Parameter	Value	Parameter	Value	Parameter	Value
J	0.1531 kgm^2	$\eta_{v,\phi}$	0.337 Nm s/rad	$\boldsymbol{Q}_N$	diag(10,5,100,5)
l_a	0.3 m	$\eta_{v,\theta}$	0.035 Nm s/rad	$\boldsymbol{Q}_f$	diag(10,5,100,5)
m_a	1 kg	K_p	-17	$\boldsymbol{R}_N$	0.1
l_p	0.205 m	K_i	-18	$\boldsymbol{F}_\tau$	[1;1]
m_p	0.3 kg	K_d	-2.5	$\boldsymbol{g}_\tau$	[10; -10]
M	1 kg	$\boldsymbol{Q}$	diag(10,1,10,1)	$\mathbf{F}_x, \boldsymbol{g}_x$	[]
g	9.81 m/s^2	$\boldsymbol{R}$	1	N	10
$\tau_{\theta d,PID}$	0.623 Nm	$\tau_{\theta d,LQR}$	4.2 Nm	$\tau_{\theta d,LMPC}$	1.081 Nm

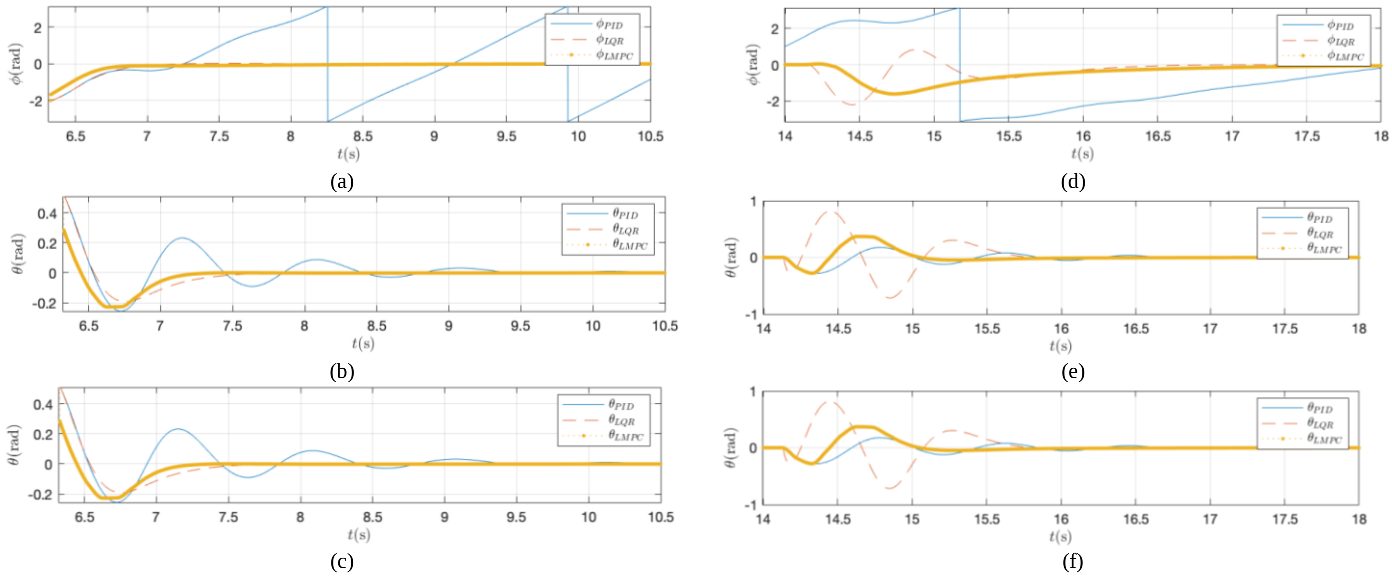

Fig. 2. Simulation data: (a), (b), (c) is the plot of ϕ, θ, and $\tau_{\phi c}$ during stabilization phase; and (d), (e), (f) is the plot of ϕ, θ, and $\tau_{\phi c}$ during disturbance introduction.

TABLE II. CONTROLLER PERFORMANCE EVALUATION PARAMETERS VALUE

Controller	Stabilization			Disturbance Introduction		
	ITAE (rad s)	ISCI ((Nm)^2)	Settling Time (s)	ITAE (rad s)	ISCI ((Nm)^2)	Settling Time (s)
PID	0.27	23.94	3.44	0.71	12.41	3.56
LQR	0.06	7.54	1.42	1.57	298.73	2.87
LMPC	0.04	5.32	0.96	0.63	22.18	2.91

deviation. The use of different disturbance magnitudes does not intend to represent unfair testing, but serves as a simple quantitative robustness assessment across different control strategies. Using this method allows direct evaluation of the controller's capabilities for disturbance rejection, where stronger controllers will obviously require more torque to disturb it. To give a natural disturbance effect, the value of disturbance will be applied as a sudden change in the pendulum arm input (e input control ($\boldsymbol{\tau}$) in (5) can be rewritten to:

$$\boldsymbol{\tau}(\phi,\theta) = [\tau_\phi, \tau_{\theta d}]^T \quad (15)$$

where τ_ϕ is the control input from the controller calculations and $\tau_{\theta d} \in \{\tau_{\theta d,PID}, \tau_{\theta d,LQR}, \tau_{\theta d,LMPC}\}$ is the disturbance input.

E. Swing-Up Strategy

In a real-life scenario, the Furuta pendulum system will naturally stay downward and needs to be brought up until it is near the equilibrium position. Since the controller mostly works on the linearized model that works around the equilibrium state, a switching controller is needed. The swing up controller will be used to bring up the pendulum near its equilibrium state, and then it will be switched to the stabilization control.

$$\tau_\phi = \begin{cases} \tau_{\phi s}, & \text{for } 0 \le |\theta| < \pi/6 \\ \tau_{\phi c}, & \text{for } |\theta| \ge \pi/6 \end{cases} \quad (16)$$

With $\tau_{\phi s}$ is generated using the simple energy-based swing up methods like what is used in [5], and $\tau_{\phi c} \in \{\tau_{\phi,PID}, \tau_{\phi,LQR}, \tau_{\phi,LMPC}\}$ is the input control signal from the controller calculations. Using (16), the system will initially bring up near the equilibrium ($0 \le |\theta| < \pi/6$) then the stabilization algorithm will take place and stabilize the pendulum at the upright positions.

IV. SIMULATION RESULT AND DISCUSSION

To control the furuta pendulum system described in the non-linear equation (5), numerical discrete time-step simulation is done using the MATLAB software. Initially, the system will be positioned at the steady downward state $x_0 = [0,0,\pi,0]^T$, and then bring upward using (16) until $|\theta|$ is reached $\pi/6$. When the switch condition is achieved and τ_ϕ is moved to $\tau_{\phi c}$ last state will be recorded and feed to the three different controller, PID (9), LQR (11) and LMPC (13).

All of the simulation and system parameters can be seen in Table 1. Initially, disturbance ($\tau_{\theta d}$) is assumed to be neglected, and a significant disturbance is only applied around three seconds after the stabilization reaches its steady state conditions for each controller. The controller will calculate each of its control input signals in every time step (0.001s or 1000Hz in this work), and then the input will be fed to the non-linear equation (5) to get the updated state. Updating the state requires some integration process to get the next step state. Therefore, in this work, the Runge-Kutta method of order 4 (RK4) is used to integrate the system output. The resulted states changes are then plotted and analyzed using a performance index evaluation: Integral of Time-weighted Absolute Error (ITAE) to value the time needed to reach the steady state value, Integral of Square Control Input (ISCI) to value the energy usage of the controller, and settling time to see the time needed from the controller start to react until equilibrium is reached.

The results of numerical simulations are shown in fig. 2 and Table 2. Initially, swing-up is executed to bring up the pendulum arm from downward positions to reach near equilibrium using (16). The latest state value from the swing up execution ($x_{sf} = [-1.9933, 1.9873, 0.5229, -2.0135]^T$) is then fed to the controller to get stabilized towards the reference state x_g . During the stabilization phase, the control response (ITAE) of LMPC becomes the best related to LQR and PID, although LQR control is the one that has the lowest overshoot angle. In terms of stabilization time, it is also done faster using LMPC (0.96 s), followed by LQR (1.42 s) and PID (3.44 s). While LMPC and LQR can reach a steady arm position at a stable position, PID is constantly moving the arm to the left, caused by a continuous small error that is proportionally compensated by the algorithm. Thus, the energy consumption of the PID in the stabilization phase becomes the biggest (23.94 $(Nm)^2$), while LMPC is the lowest (5.32 $(Nm)^2$), followed by LQR (7.54 $(Nm)^2$).

An interesting pattern was seen when the system is disturbed until it deviates for around 16º (0.279 rad), where LQR is the one that reacts faster, but also overshoots much larger than the other two controllers. Although it converges faster than the others (LQR is 2.87s vs 2.91s for LMPC and 3.56s for PID), the energy consumption during disturbance stabilization of LQR is the largest (298.73 $(Nm)^2$). At the same time, PID and LMPC only take 12.41 $(Nm)^2$ and 22.18 $(Nm)^2$ energy usage, respectively. Control response during the disturbance stabilization also gave a similar pattern, where LQR was the worst, followed by PID and LMPC as the best. However, it can also be seen that the energy used to disturb the LQR control (4.2 Nm) is also higher than LMPC (1.081 Nm) and PID (0.62 Nm). LQR is quite stiff and sensitive to external disturbance, while LMPC is less sensitive since it keeps its minimum effort constraint. PID is more uncomplicated to disturb since it only relies on the state error to generate the control signal.

Overall, the proposed controlling methods can stabilize the Furuta pendulum system to the upright positions with their own advantages and disadvantages, which can be summarized as:

1) Classical PID control is the one that has the smallest complexity in terms of implementation and controller design. It can be set to react faster to every small error and works with a nonlinear system flawlessly. However, the tuning process is tricky for a volatile system like the Furuta pendulum. Also, it converges more slowly to the equilibrium state, making the energy consumption higher.
2) LQR control tackles the needs of fine-tuning the PID and guarantees a stable tracking process. This type of controller can reach the equilibrium using a smaller amount of energy and a faster converging time. However, since it is designed to optimize infinitely with a single gain value, it is not adaptive to any disturbance that happens during the stabilization process.
3) LMPC, on the other hand, offers some optimal control input in every step that always works under the constraint criteria. This type of control gives a reasonable converging speed and energy usage, while keeping the control effort and angle overshoot as low as possible. However, this type of controller requires higher computation resources (1190 μs vs PID 266 μs and LQR 253 μs), since it needs to compute its optimal input in every step and receding horizon prediction step.

V. CONCLUSION

In this paper, three different types of stabilizing controllers were employed on a Furuta pendulum system. The stabilizing controller was activated after the swing-up process reached its switching threshold. The controller can keep the Furuta pendulum system around its equilibrium states, and bring the system back after a disturbance injection. Each of the controllers has its own advantages and disadvantages evaluated using three performance index parameters, which are ISCI, ITAE, and settling time. In the overall performance evaluation, LMPC has the best and reasonable response both in the stabilization and disturbance rejection phases. Future work can be done to refine the controller to have a much better performance, such as using a gain-scheduled controlling system, cascade control, nonlinear MPC, and LQG. It is also interesting to test the proposed algorithm/methods on a real Furuta pendulum system, and give the same exact-measured disturbance effect to the system.

ACKNOWLEDGMENT

This work is supported by the Digital Control and Computation (DCC) Laboratory, Electrical Engineering Department, Faculty of Engineering, Tanjungpura University.

REFERENCES

[1] Rani M, Kamlu SS. Optimal LQG controller design for inverted pendulum systems using a comprehensive approach. Sci Rep 2025;15. https://doi.org/10.1038/s41598-025-85581-3.

[2] Mori S, Nishihara H, Furuta K. Control of unstable mechanical system control of pendulum†. Int J Control 1976;23:673–92. https://doi.org/10.1080/00207177608922192.

[3] Aracil J, Åström KJ, Pagano DJ. Global Bifurcations in the Furuta Pendulum. IFAC Proceedings Volumes 1998;31:37–42. https://doi.org/10.1016/s1474-6670(17)40307-7.

[4] Furuta K, K.J. Astrom. Swinging up a pendulum by energy control. Automatica 2000;36:287–95.

[5] Wiklund M, Kristenson A, Aström KJ. A New Strategy for Swinging Up an Inverted Pendulum. IFAC Proceedings Volumes 1993;26:757–60. https://doi.org/10.1016/s1474-6670(17)48372-8.

[6] Aracil J, Acosta JÁ, Gordillon F. A nonlinear hybrid controller for swinging-up and stabilizing the Furuta pendulum. Control Eng Pract 2013;21:989–93. https://doi.org/10.1016/j.conengprac.2013.04.001.

[7] Fahmizal, Geonoky, Maghfiroh H. Rotary Inverted Pendulum Control with Pole Placement. Journal of Fuzzy Systems and Control 2023;1:90–6. https://doi.org/10.59247/jfsc.v1i3.152.

[8] Muhammad R, Hadary F. Performance Analysis of Bang-Bang, Single PID, and Cascade PID Controllers for Cart Pendulum Stabilization. CENTIVE 2024;4:187–95.

[9] Chiem NX, Thang LT. Synthesis of LQR Controller Based on BAT Algorithm for Furuta Pendulum Stabilization. Journal of Robotics and Control (JRC) 2023;4:662–9. https://doi.org/10.18196/jrc.v4i5.19661.

[10] R. Muhammad, F. Hadary, and B. Wibowo Sanjaya, “Pemodelan dan Simulasi Sistem Kendali Rotary Inverted Pendulum Berbasis Mikrokontroler STM32”, JITET, vol. 13, no. 3S1, Oct. 2025.

[11] Ma Y, Xu D, Huang J, Li Y. Robust Control of An Inverted Pendulum System Based on Policy Iteration in Reinforcement Learning. Applied Sciences (Switzerland) 2023;13. https://doi.org/10.3390/app132413181.

[12] Yilmaz FC, Onler R, Tatlicioglu E, Zergeroglu E. A transfer learning based deep neural network adaptive controller for the Furuta pendulum subject to uncertain disturbance signals. Sci Rep 2025;15. https://doi.org/10.1038/s41598-025-10021-1.

[13] Guo H, Wu J, Yin Z, Zhang W. Comparative Study of Furuta Pendulum Based on LQR and PID Control. J Phys Conf Ser, vol. 2562, Institute of Physics; 2023. https://doi.org/10.1088/1742-6596/2562/1/012075.

[14] Jezierski A, Mozaryn J, Suski D. A comparison of LQR and MPC control algorithms of an inverted pendulum. Advances in Intelligent Systems and Computing, vol. 577, Springer Verlag; 2017, p. 65–76. https://doi.org/10.1007/978-3-319-60699-6_8.

[15] Prasad LB, Tyagi B, Gupta HO. Optimal control of nonlinear inverted pendulum system using PID controller and LQR: Performance analysis without and with disturbance input. International Journal of Automation and Computing 2014;11:661–70. https://doi.org/10.1007/s11633-014-0818-1.

[16] Varghese ES, Vincent AK, Bagyaveereswaran V. Optimal control of inverted pendulum system using PID controller, LQR and MPC. IOP Conf Ser Mater Sci Eng, vol. 263, Institute of Physics Publishing; 2017. https://doi.org/10.1088/1757-899X/263/5/052007.

[17] Gäfvert, M, Modelling the Furuta Pendulum. (Technical Reports TFRT-7574). Department of Automatic Control, Lund Institute of Technology (LTH), 1998.

[18] Li XL, Park JG, Shin HB. Comparison and evaluation of anti-windup PI controllers. Journal of Power Electronics 2011;11:45–50. https://doi.org/10.6113/JPE.2011.11.1.045.

[19] Murray RM, LQR Control. California Institute of Technology, Control and Dynamical Systems, 2006.

[20] Önen Ü. Particle Swarm Optimization Based LQR Control of an Inverted Pendulum. Engineering and Technology Journal 2017;2. https://doi.org/10.18535/etj/v2i5.01.

[21] Augustine MT, Model Predictive Control using MATLAB. Department of Electrical Engineering. Indian Institute of Technology Delhi, India, 2023.

[22] Astrom KJ, Witten B, Computer-Controlled Systems Theory and Design THIRD EDITION. Tsinghua University: Prentice Hall, 1997.