A SURVEY OF LQG OVER MPC AND LQR CONTROL FOR ROTARY INVERTED PENDULUM

Phuc-Hoang Huynh, Cong-Duy Pham, Nam-Binh Vu, Trung-Kien Duong, Duc-Hoang-Khanh Vo, Truong-Giang Le, Duy-Khanh Nguyen, Thi-Thanh-Hoang Le *

Ho Chi Minh City University of Technology and Education (HCMUTE)
Vo Van Ngan Street, No. 01, Ho Chi Minh City, Vietnam
***** Corresponding author. E-mail: hoangltt@hcmute.edu.vn

Abstract: In this paper, we examine the theoretical cost function equivalence between Model Predictive Control (MPC) and Linear-Quadratic Gaussian (LQG) control, as well as Linear-Quadratic Regulator (LQR) control under specific conditions. Specifically, we linearize the Rotary Inverted Pendulum (RIP) system and construct a Kalman filter state estimator for application in both the LQG and MPC controllers with input and output constraints. We also assume measurable and computable states when designing the LQR controller. Through simulation and experimentation, we demonstrate that, despite the equivalence in cost functions, the output response of MPC is significantly better than that of both LQG and LQR. Our findings not only substantially bridge gaps in control theory but also emphasize the robustness of MPC in complex real-world applications. These insights pave the way for more effective and reliable control strategies across various engineering fields.

Keywords: Rotary inverted pendulum, Model Predictive Control, LQR, LQG, Kalman filter.

1. Introduction

A control system designed to operate in its optimal mode is one that always remains in an optimal state according to a certain quality criterion (achieving an extreme value) [1]. Therefore, from the 1980s to present, numerous research studies have been conducted [2], including methods such as classical Euler-Lagrange variable transformation [3], Bellman's dynamic programming [4], Pontryagin's minimum principle [1], LQR control [5], LQG control [6], MPC [7].

In 1960, Kalman introduced LQR and LQG, which represented a significant advancement, as MIMO problems had previously been designed using the "successive loop-closure" approach, often yielding suboptimal results, such as poorly coordinated controls that interfere with each other, wasting control authority [2], Since then, these methods have found numerous practical applications, such as aircraft system control [8], control of the longitudinal flight dynamics of a fixed-wing UAV [9] and oxygen stoichiometry control [10],…

Unlike LQR and LQG, at each time step, an MPC controller receives or estimates the plant's current state. It then calculates a sequence of control actions that minimize the cost over the horizon by solving a constrained optimization problem based on an internal plant model and the current system state. The controller then applies only the first computed control action to the plant, discarding the rest. This process repeats at each subsequent time step [11, 12]. Due to this approach, MPC algorithms provide excellent control quality, particularly for Multiple-Input Multiple-Output (MIMO) systems [13]. This explains why various MPC algorithms have been applied in numerous fields on an industrial scale, including process industries, oil refineries [12], process control [14], autonomous vehicles [15], robotics [16], and even in medical/health sciences [17].

When cost function is quadratic, plant is linear and without constraints, and the horizon tends to infinity, MPC is equivalent to LQG control (or to an LQR control if the plant states are measured and no estimator is used) [11, 18]. Because of these characteristics, our research team aims to compare these three controllers on a RIP – a system that is relatively easy to construct yet exhibits nonlinear, unstable characteristics that are challenging to control [19, 20]. For these three controllers, we use a linearized model at the operating point for controller design. Although our MPC design employs a finite prediction horizon with input and output constraints, which may lead to a trade-off in controller performance to avoid violating soft output constraints [21], we have designed all three controllers with identical cost function parameters to compare the superior performance of the MPC controller relative to LQG and LQR in both simulations and experiments.

In [22], the authors modeled and linearized the RIP at the equilibrium operating point to design both MPC and LQR controllers. In that study, the controller parameters were set with the goal of achieving the same response in the arm angle to demonstrate that MPC controller's tracking performance in arm angle reference signal was superior to that of LQR. Building on these findings, in this paper, we aim to compare the response of three controllers when they share the same cost function weights. We will redesign parameters of MPC and LQR controllers to ensure that they have identical cost function weights. Additionally, we will specify Kalman state estimator used in MPC and apply this estimator to LQR controller, converting it into an LQG controller. We then conduct simulations and experiments on RIP to verify responses of three controllers.

2. RIP Model

RIP is a system consisting of an arm and a pendulum, with a DC motor mounted at the end of the arm. The pendulum is always stable in the downward position but unstable in the upright position. Therefore, the objective is to design a controller to keep the pendulum upright and move the arm along a predefined trajectory. The model structure is shown in **Fig. 1**, For details a reader can be referenced to [22].

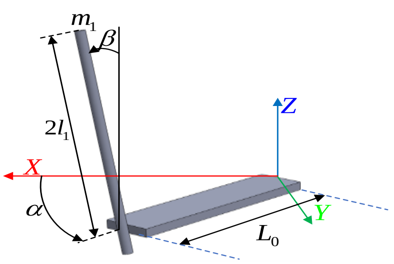

Fig. 1. RIP Structure.

3. Design of MPC controller

MPC uses plant, disturbance, and noise models for prediction and state estimation. The model structure used for the MPC controller is illustrated in Fig. 2 [18].

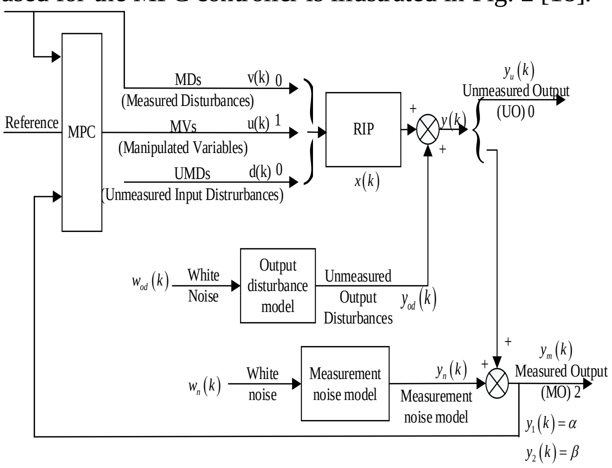

Fig. 2. The model structure used in an MPC controller.

➢ **Controller parameters**

- Sample time: Ts = 0.01s
- Prediction horizon: P = 50
- Control horizon: m = 3
- $L_{s_y} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; L_{s_u} = 10$
- $w_{i,1}^y = 25; w_{i,2}^y = 50$
- $w_{i,1}^u = 0; w_{i,1}^{\Delta u} = 1$
- $-12 \le u(k+i) \le 12$
- $\frac{-\pi}{6} \le \alpha(k+i\,|\,k) \le \frac{\pi}{6}$
- $\rho_\varepsilon = 100$

The state equations describing the disturbances shown in Fig. 2 are described as follows[18, 23]:

➢ **Output disturbance model**

Since the most common disturbance is an unmeasured step disturbance added to the output of the system, the input to the disturbance model will be white noise with a mean value of zero and unit variance [24]

Therefore, we choose the output disturbance model of the system as follows:

$$\begin{cases} x_{od}(k+1) = A_{od}x_{od}(k) + B_{od}w_{od}(k) \\ y_{od} = C_{od}x_{od}(k) + D_{od}w_{od}(k) \end{cases} \quad (1)$$

$$\Rightarrow \begin{cases} x_{od}(k+1) = x_{od} + 0.01w_{od}(k) \\ y_{od} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x_{od}(k) + \begin{bmatrix} 0 \\ 0 \end{bmatrix} w_{od}(k) \end{cases}$$

Where:

- y_{od} : is disturbance added to output of the system.
- w_{od} : which is white noise with a mean value of zero and unit variance.

➢ **Measurement Noise Model**

Assuming that the measurable disturbance added to the system is white noise, we have the disturbance model for the system as follows:

$$\begin{cases} x_n(k+1) = A_n x_n(k) + B_n w_n(k) \\ y_n(k) = C_n x_n(k) + D_n w_n(k) \end{cases} \quad (2)$$

$$\Rightarrow y_n(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} w_n(k)$$

Where:

- y_n: which is the measurable noise added to the output of the system.
- w_n: which is white noise with a mean value of zero and unit variance.

➢ **State observer**

Combination of the models shown in Fig. 2 yields the state observer[25]:

$$\begin{cases} x_c(k+1) = Ax_c(k) + Bu_o(k) \\ y(k) = Cx_c(k) + Du_o(k) \end{cases} \quad (3)$$

Where:

- $u_0^T(k) = \begin{bmatrix} u^T(k) & w_{od}^T(k) & w_n^T(k) \end{bmatrix}$
- $A = \begin{bmatrix} A_d & 0 \\ 0 & A_{od} \end{bmatrix} = \begin{bmatrix} 1 & 0.001 & 0.0008 & 0 & 0 \\ 0 & 0.9832 & 0.1636 & -0.00143 & 0 \\ 0 & 0 & 1.003 & 0.01 & 0 \\ 0 & -0.0105 & 0.5142 & 0.996 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

- $B=\begin{bmatrix}B_{pu} & 0\\ 0 & B_{od}\end{bmatrix}=\begin{bmatrix}B_d.L_{s_u} & 0\\ 0 & B_{od}\end{bmatrix}=\begin{bmatrix}0.002 & 0 & 0 & 0\\ 0.398 & 0 & 0 & 0\\ 0.001 & 0 & 0 & 0\\ 0.249 & 0 & 0 & 0\\ 0 & 0.01 & 0 & 0\end{bmatrix}$
- $C=\begin{bmatrix}C_p & C_{od}\end{bmatrix}=\begin{bmatrix}L_{s_y}^{-1}.C & C_{od}\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0 & 1\end{bmatrix}$
- $D=\begin{bmatrix}0 & D_{od} & \begin{bmatrix}D_n\\ 0\end{bmatrix}\end{bmatrix}=\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$

➢ **State estimation**

The controller uses a steady-state Kalman filter that is derived from the state observer. MPC estimates the state of the system with the following state equation for the estimator [25]:

$$x_c(k+1|k)=Ax_c(k|k-1)+B_u u(k)+Le(k) \quad (4)$$

The current estimator generates the state estimates $x_c(k|k)$:

$$x_c(k|k)=x_c(k|k-1)+Me(k) \quad (5)$$

With $e(k)$ being calculated as follows:

$$e(k)=y_m(k)-\left[C_m x_c(k|k-1)+D_{mv}v(k)\right] \quad (6)$$

Where:

- $x_c(k|k-1)$: Controller state estimate from previous control interval, *k*–1
- $u(k-1)$: Manipulated variable (MV) actually used in the system from *k*–1 to *k*.
- $v(k)$: Current measured disturbances
- $y_m(k)$: System output measured at current time.
- B_u: The observation parameter column B corresponds to the input $u(k)$.
- C_m: The observation parameter rows C correspond to the output of the measured system.
- D_{mv}: The rows and columns of the observation parameter D correspond to the output of the measured system and the measured noise input.
- L,M: Constant Kalman gain matrices.

➢ **Calculating the gain for the Kalman filter**

Consider a discrete RIP with known inputs *u*, white process noise *w*, and white measurement noise *v*:

$$\begin{cases}x(k+1)=Ax(k)+B_u u(k)+Gw(k)\\ y(k)=C_m x(k)+D_{pu}u(k)+Hw(k)+v(k)\end{cases} \quad (7)$$

Without loss of generality, $u(k)=0; v(k)=0$

$$w^T(k)=\begin{bmatrix}w_{od}^T(k) & w_n^T(k)\end{bmatrix}$$

Covariance matrices are calculated as follows [25]:

$$Q=E\left\{Bww^TB^T\right\}=BB^T$$

$$=\begin{bmatrix}0 & 0.0008 & 0 & 0.0005 & 0\\ 0.008 & 0.159 & 0.0005 & 0.0993 & 0\\ 0 & 0.0005 & 0 & 0.0003 & 0\\ 0.0005 & 0.0993 & 0.0003 & 0.0621 & 0\\ 0 & 0 & 0 & 0 & 0.0001\end{bmatrix} \quad (8)$$

$$R=E\left\{Dww^TD^T\right\}=DD^T=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \quad (9)$$

$$N=E\left\{Bww^TD^T\right\}=BD^T=\begin{bmatrix}0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\end{bmatrix}^T \quad (10)$$

Here, $E\{...\}$ denotes the expectation.

The matrices L and M will be solved from the Riccati equation [6]. For simplicity in calculations, the author uses the 'kalman' command in Matlab, resulting in the gain matrices M and L as follows:

$$\begin{bmatrix}\sim & L & \sim & M\end{bmatrix} =kalman\left(ss\left(A,\begin{bmatrix}B_u & G\end{bmatrix},C_m,\begin{bmatrix}D & H\end{bmatrix},Ts\right),Q,R,N\right) \quad (11)$$

$$L=\begin{bmatrix}0.0723 & 0.0317\\ 0.2659 & 0.2788\\ 0.0214 & 0.1524\\ 0.1681 & 1.0357\\ -0.0001 & -0.0093\end{bmatrix};\ M=\begin{bmatrix}0.0696 & 0.029\\ 0.2674 & 0.2612\\ 0.0198 & 0.1424\\ 0.1615 & 0.9695\\ -0.0001 & -0.0093\end{bmatrix} \quad (12)$$

where

$$A=\begin{bmatrix}1 & 0.001 & 0.0008 & 0 & 0\\ 0 & 0.9832 & 0.1636 & -0.00143 & 0\\ 0 & 0 & 1.003 & 0.01 & 0\\ 0 & -0.0105 & 0.5142 & 0.996 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix};$$

$$B_u=\begin{bmatrix}0.0002\\ 0.0398\\ 0.0001\\ 0.0249\\ 0\end{bmatrix};G=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix};\quad D=D_{pu}=\begin{bmatrix}0\\ 0\end{bmatrix};$$

$$C_m=C=\begin{bmatrix}1 & 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0 & 1\end{bmatrix};H=\begin{bmatrix}0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\end{bmatrix};\ T_s=0.01$$

4. Design of LQR Controller

As mentioned in the previous study on the LQR algorithm [22] the author selects the weight matrices Q, R as follows:

$$Q=\begin{bmatrix}25 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 50 & 0\\ 0 & 0 & 0 & 0\end{bmatrix};\ R=0.1 \quad (13)$$

The solution to the problem to obtain the matrix K is the solution of the Riccati equation [5]:

$$K = \begin{bmatrix} -14.085 & -7.794 & 146.59 & 22.3 \end{bmatrix} \quad (14)$$

5. Design of LQG Controller

LQG control method is a combination of LQR controller and Kalman state estimator, as in Fig. 3[26]:

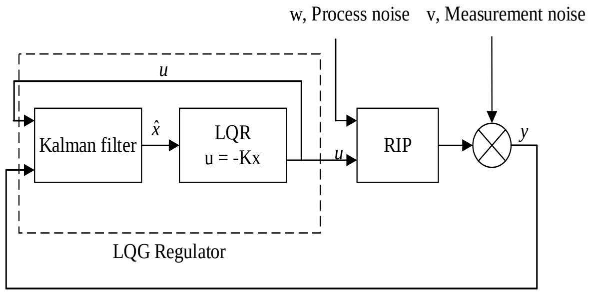

Fig. 3. Diagram of LQG control.

The Kalman state estimator is referenced with the state equation of the system mentioned in (7). State estimation equations are referred to in (4), (5). Gain matrices L and M are calculated in (12). The gain matrix K is calculated in (14).

6. Results of Simulation

To simulate the responses of the three controllers MPC, LQR, and LQG, the author constructed the diagram using MATLAB Simulink as follows:

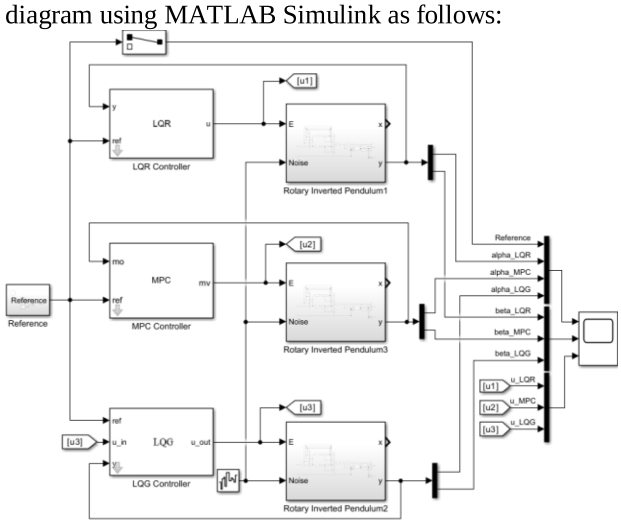

Fig. 4. Simulation diagram of the controllers.

Where LQR controller block is designed as follows:

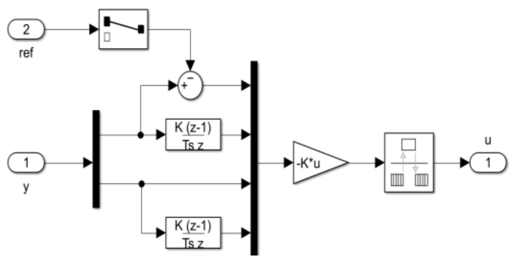

Fig. 5. Internal diagram of the LQR controller.

The LQG controller block is designed as follows:

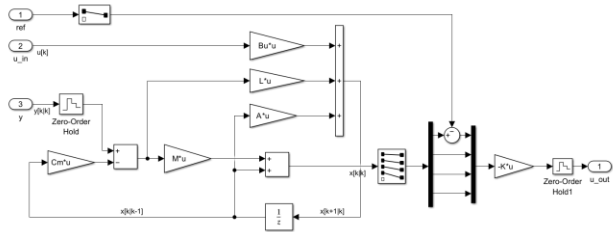

Fig. 6. Internal diagram of the LQG controller.

The MPC controller block is taken from the library provided by MATLAB Simulink.

The author initializes the simulation system with the initial state $x = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$ and allows the arm angle (angle α) to follow the reference signal $f = 0(rad)$ for the first 0.5 seconds and $f = 1(rad)$ after, pendulum angle (angle β) tracks the reference signal $f = 0(rad)$, with the power noise set to 10^{-7}. The simulation results for all three controllers are plotted on the same axes, corresponding to the pendulum angle response, the arm angle response, and the control voltage, respectively from top to bottom:

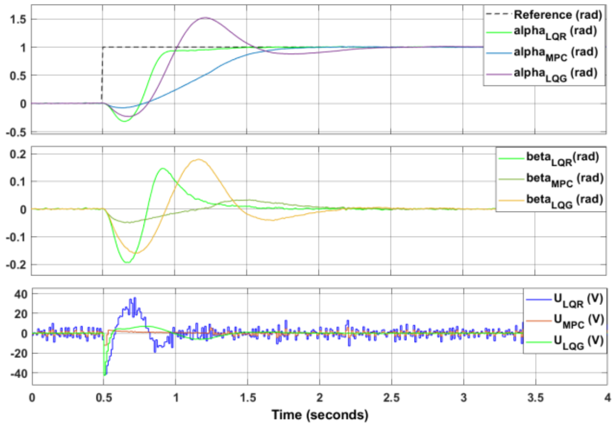

Fig. 7. Responses of three controllers during simulation.

⇨ The simulation results show that:

- Notably, when there is an abrupt change in the reference signal, the output response at the arm angle and the control voltage of the MPC controller does not exceed the constraint limits $-\frac{\pi}{6} \le \beta \le \frac{\pi}{6}$, $12 \le u \le 12(v)$ while still ensuring system stability.

- With a stability criterion of 2%, there are clear differences in the arm angle response among the three controllers. The rise times for the LQR controller $t_s(LQR) = 0.9(s)$,MPC controller $t_s(MPC) = 1.33(s)$, and LQG controller $t_s(LQG) = 1.87(s)$. The MPC has sacrificed control performance to ensure that constraints are not violated. For the LQG controller, due to state estimation and not providing real-time optimal control signals like the MPC, the response in the arm angle has a longer rise time and exhibits an overshoot.

- MPC solves optimization problem at each sampling interval based on estimated output of the system, but estimated values are not entirely accurate. In contrast, the LQR does not have an estimator and is therefore significantly affected by noise. Consequently, we observe that the average control voltage of LQR is the highest, while that of the LQG is the lowest.

7. Results of Experiment

From the basic model structure in Fig. 1 the author developed the RIP system hardware for experimentation, as shown in the following figure:

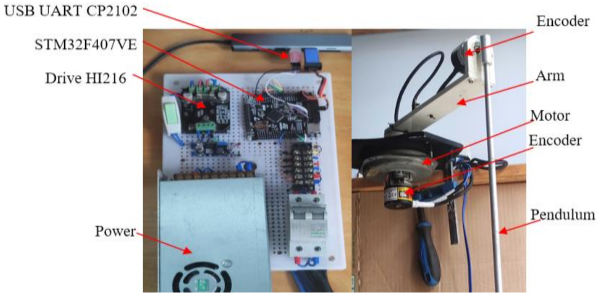

Fig. 8. Hardware platform of RIP.

The model we experimented with is the same model mentioned previously, for details a reader can be referenced to [22].

We set the reference signal for the arm angle to change from 0 rad to 1 rad. The experimental results are plotted on the same response graph, as shown below:

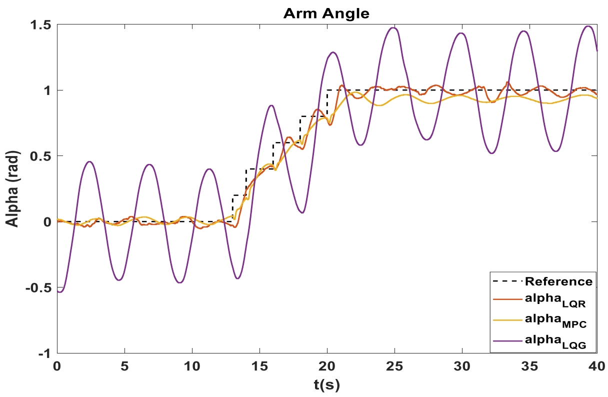

Fig. 9. Arm angle response.

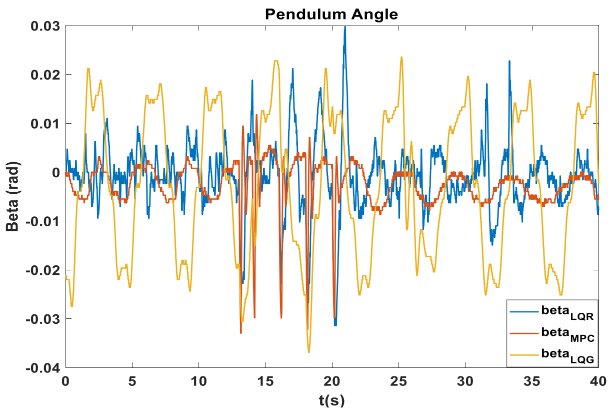

Fig. 10. Pendulum angle response.

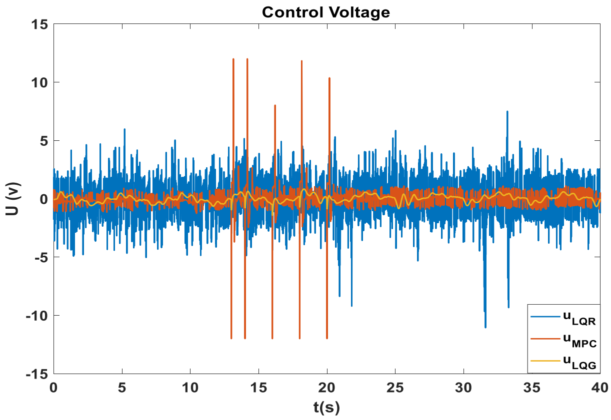

Fig. 11. Control voltage of the RIP system.

We use the Root Mean Square Error (RMSE) criterion to evaluate the expected values against the measured values. The results of the calculations are presented in the table below:

Tab. 1. Quality of the RIP according to the Root Mean Square Error standard.

Controler	$\alpha(rad)$	$\beta(rad)$	$u(v)$
LQG	0.341	0.015	0.308
LQR	0.061	0.007	1.75
MPC	0.073	0.005	0.8

⇨ The experimental results show that:

- The responses in the pendulum angle and arm angle oscillate around the reference value, which is significantly different from the simulation results. This is because the experimental model is affected by signal wiring, which inadvertently creates a torque moment when the arm moves. This indicates that the system is influenced not just by white noise, as in the simulation.
- Arm angle response of LQR controller has the smallest RMSE index; however, in Fig. 9 , it shows poor response at certain moments. This is due to influence of system noise generated, and the control voltage has the highest RMSE value.
- Results also indicate that the RMSE index for the arm angle and pendulum angle responses of LQG controller is significantly worse than that of the other two controllers, with larger amplitude oscillations in the curves. This shows that for the LQG controller, selecting parameters for Kalman state estimator is crucial to closely estimate the actual state of the system. Notably, control voltage for LQG controller has the smallest RMSE index.
- We also observe that the response of the MPC controller is the best. The RMSE response index for the arm angle of the MPC controller is only 0.012 (rad) higher than that of the LQR controller and lower by 0.002 (rad) for the pendulum angle, with the control voltage also significantly lower. Although all three controllers have the same objective function weights, both the MPC and LQG controllers share the same state estimator, yet the MPC exhibits less oscillation in

the curves and is particularly more stable than the LQR. This demonstrates the superiority of MPC when performing runtime optimization and alleviates the difficulties in selecting parameters for the state estimator, unlike LQG.

8. Conclusions

The authors have developed three controllers with controller parameters having the same objective function weights. At the same time, a Kalman state estimator has been constructed for both MPC and LQG control. After simulating and testing all three controllers LQG, LQR, and MPC on the RIP system, in addition to the conclusions mentioned in the previous study [22], we observed that despite LQG and LQR solving the optimization problem over the entire prediction horizon, while MPC solves the optimization problem at each sampling point, MPC yields more favorable experimental results when the same optimization criterion is applied.

Acknowledgement

This paper belongs to the HCMUTE project for students in 2025. We appreciate the support of HCMUTE in funding this project. We also want to thank Ms. Eng. Phong-Luu Nguyen and PhD. Van-Dong-Hai Nguyen for supporting the simulation.

8. References

[1] Nguyen T.P.H.: "Modern Control Theory", Ho Chi Minh City National University, 2009.
[2] Bryson A.E.: "Optimal control-1950 to 1985", IEEE Control Systems Magazine, vol. 16, no. 3, pp. 26-33, 1996.
[3] Ortega R. et al: "Euler-Lagrange systems", Communications and Control Engineering. Springer, London, 1998.
[4] Bellman R.: ""Dynamic programming", Science, vol. 153, no. 3731, pp. 34-37, 1966.
[5] Kalman R. E.: "Contributions to the theory of optimal control", Bol. soc. mat. mexicana, vol. 5, no. 2, pp. 102-119, 1960.
[6] Kalman R.E.: "A new approach to linear filtering and prediction problems", J. Fluids Eng., vol. 82, no. 1, pp. 35-45, 1960.
[7] Garcia C.E. et al: "Model predictive control: Theory and practice—A survey", Automatica, vol. 25, no. 3, pp. 335-348, 1989.
[8] Chrif L., Kadda Z. M.: "Aircraft control system using LQG and LQR controller with optimal estimation-Kalman filter design", Procedia Engineering, vol. 80, pp. 245-257, 2014.
[9] Ingabire A. and Sklyarov A. A.: "Control of longitudinal flight dynamics of a fixedwing UAV using LQR, LQG and nonlinear control", in E3S Web of Conferences, vol. 104: EDP Sciences, p. 02001, 2019.
[10] Niknezhadi A. et al: "Design and implementation of LQR/LQG strategies for oxygen stoichiometry control in PEM fuel cells based systems", Journal of Power Sources, vol. 196, no. 9, pp. 4277-4282, 2011.
[11] "What is Model Predictive Control?" MathWorks. https://www.mathworks.com/help/mpc/gs/what-is-mpc.html (accessed 30/8, 2024).
[12] Qin S.J., Badgwell T.A.: "A survey of industrial model predictive control technology", Control Engineering Practice, vol. 11, no. 7, pp. 733-764, 2003.
[13] Chaber P., Lawrynczuk M.: "Fast Analytical Model Predictive Controllers and Their Implementation for STM32 ARM Microcontroller", IEEE Transactions on Industrial Informatics, vol. 15, no. 8, pp. 4580-4590, 2019.
[14] Shi G. et al: "A process-model-free method for model predictive control via a reference model-based proportional-integral-derivative controller with application to a thermal power plant", Frontiers Control Eng, vol. 4, 2023.
[15] Yu S. et al: "Model predictive control for autonomous ground vehicles: A review", Auto. Intell. Syst., vol. 1, no. 1, pp. 1-17, Aug. 2021.
[16] Gold T., Volz A., Graichen K.: "Model predictive interaction control for industrial robots", IFAC-PapersOnLine, vol. 53, no. 2, pp. 9891–9898, 2020.
[17] Parihar S. et al: "Model predictive control and its role in biomedical therapeutic automation: A brief review", Appl. Syst. Innov., vol. 5, no. 6, p. 118, 2022.
[18] Bemporad A. et al, "Model Predictive Control Toolbox", User's Guide, Version, vol. 2, 2004. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=f189074fadc648c7c4c29c0467afdc64965ad787.
[19] Huynh T.H., "Intelligent Control Systems", Ho Chi Minh City National University, 2006.
[20] Nguyen V.D.H. et al: "PID-neuron controller design for rotary inverted pendulum system," Journal of Technical Education Science, pp. 37–43, 2012.
[21] "MPC Control of an Inverted Pendulum on a Cart", MathWorks. https://www.mathworks.com/help/mpc/ug/mpc-control-of-an-inverted-pendulum-on-a-cart.html (accessed November 1st, 2024).
[22] Huynh P.-H. et al, "Model Predictive Control for Rotary Inverted Pendulum: Simulation and Experiment", Journal of Fuzzy Systems and Control, vol. 2, no. 3, pp. 215-222, Nov. 2024.
[23] "MPC Prediction Models", MathWorks. https://www.mathworks.com/help/mpc/gs/mpc-modeling.html (accessed November 1st, 2024).
[24] "Implement Custom State Estimator Equivalent to Built-In Kalman Filter", MathWorks. https://www.mathworks.com/help/mpc/ug/design-estimator-equivalent-to-mpc-built-in-kf.html (accessed 9/10, 2024).
[25] "Controller State Estimation", MathWorks. https://www.mathworks.com/help/mpc/ug/controller-state-estimation.html (accessed November 1st, 2024).
[26] Lee K. et al: "Optimal path tracking control of autonomous vehicle: Adaptive full-state linear quadratic Gaussian (LQG) control", IEEE Access, vol. 7, pp. 109120 - 109133, 2019.