Sensors and Materials, Vol. 35, No. # (2023) 1–##
MYU Tokyo

S & M ####

Design of Self-balancing Model-size Electrical Motorbike Robot Using Control Moment Gyroscope

Rodi Hartono,[1] Seung Bin Kim,[1] Patrick Tshibang A Kalend,[1]
Nam Kyun Baik,[2*] and Kyoo Jae Shin[1]

[1]Department of Artificial Intelligence Convergence, Busan University of Foreign Studies,
65, Geumsaem-ro 485 Beon-gil, Geumjeong-gu, Busan, 46234, Republic of Korea
[2]Department of Smart Convergence Security, Busan University of Foreign Studies,
65, Geumsaem-ro 485 Beon-gil, Geumjeong-gu, Busan, 46234, Republic of Korea

(Received October 18, 2022; accepted November 28, 2022; online published November 30, 2022)

Keywords: electrical motorbike robot, control moment gyroscope, pole placement controller, linear-quadratic regulator controller

In recent years, a significant amount of study has been conducted on how to control the equilibrium of an electrical motorbike robot (EMBot). This is essential because a stable EMBot must be controlled by a reliable system. In this study, we explored a number of approaches for stabilizing an EMBot by using a control moment gyroscope (CMG). The momentum from a flywheel's rotation at a set speed is employed to counteract gravity and keep the EMBot balanced. To measure the roll angle of the EMBot, an inertial sensor is used. From generalized coordinate systems, a nonlinear mathematical model of the EMBot is derived using Lagrange's equation. Pole placement (PP) and linear-quadratic regulator (LQR) controllers are utilized to test and observe the system's stability. By determining the appropriate locations of the closed-loop poles in the PP controller and finding the optimal *K*-matrix value by selecting closed-loop characteristics in the LQR method, the performance of the system in maintaining stability can be identified.

1. Introduction

Single-track vehicles, such as an electrical motorbike robot (EMBot), offer a number of benefits, one of which is that they are more convenient to use than multi-track vehicles. On the other hand, due to the nonlinear and unstable nature of the system, controlling an EMBot can be challenging.[1–3] One of the difficulties is in maintaining vertical stability or balance control when an EMBot is both in a static state and moving through the environment.[1–4] Therefore, a significant amount of research has been conducted in an attempt to discover a method that will stabilize an EMBot.

The method chosen to control the vertical stability of an EMBot is highly dependent on the EMBot's movement conditions. When an EMBot is not moving, the external torque affects the direction of the wheels and makes it more difficult for the EMBot to remain balanced. This is

*Corresponding author: e-mail: namkyun@bufs.ac.kr

ISSN 0914-4935

because when the wheels are not rotating, there is no angular momentum. In general, there are three different approaches for controlling the vertical stability of an EMBot, each with benefits and drawbacks. The approach of steering control has excellent performance when an EMBot is traveling at medium and high speeds but not when it is traveling at low speeds or stationary. This approach also has disadvantages, including ground friction and a lack of resistance when the EMBot is tilted at a steep angle. To balance the EMBot, the approach of shifting the center of mass can also be used.[5,6] However, a disadvantage of this approach is that the torque output cannot be very high. The final approach is the use of a control moment gyroscope (CMG).[7–16] The large torque that can be generated by a CMG allows it to overcome the need to stabilize the EMBot both at rest and at medium and high speeds. The CMG functions by rotating a flywheel that serves as a rotor and is mechanically connected to a motorized gimbal. The shift in the angular momentum in gaining the torque that results from the tilting of the spinning flywheel by the gimbal motor generates a gyroscopic torque effect, which can help the EMBot maintain its equilibrium.

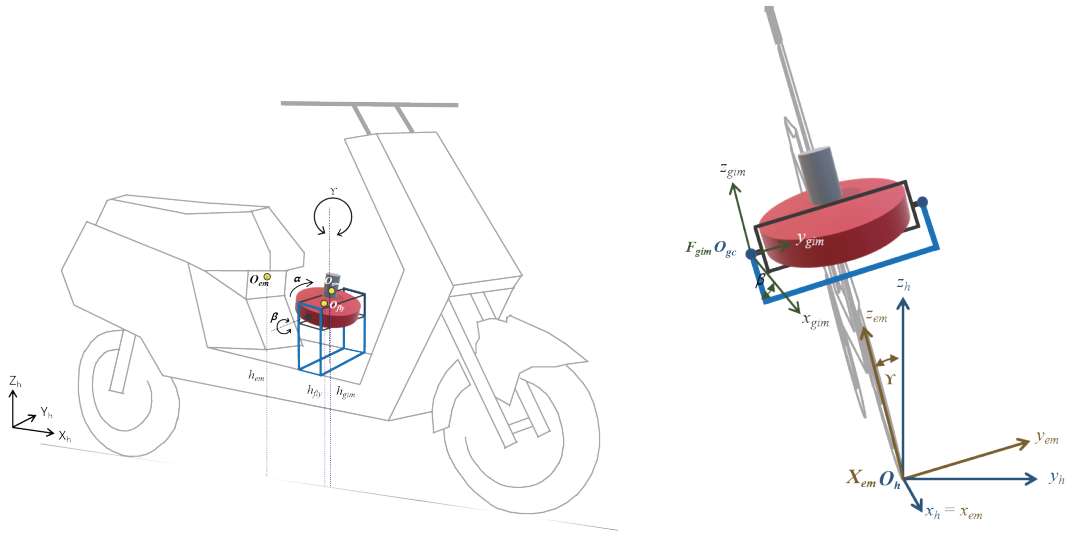
As described in the literature on the application of gyroscopic effects to actively stabilize a single-track vehicle, we conducted research on a model-size EMBot by employing the gyroscopic torque of a spinning flywheel to maintain the stability of the EMBot when stationary. Two DC motors were utilized to rotate the flywheel and the actuator used for twisting the gimbal that controlled the roll angle of the EMBot. As a sensor input, an inertial sensor is employed to monitor the roll angle of the EMBot. The roll angle data is then fed into a control system to generate the motor torque output as a drive actuator for tilting a spinning flywheel. Pole placement (PP) and linear-quadratic regulator (LQR) controllers are utilized to test and observe the system's stability. In this paper, we focus on system modeling, simulation, and observing the response of the EMBot system by implementing PP and LQR controllers based on real hardware parameters employed in maintaining stability. By determining the appropriate locations of the closed-loop poles in the PP control system and finding the optimal *K*-matrix value by selecting closed-loop characteristics in the LQR method, we can determine the performance of the system.

2. Dynamic Model of EMBot

2.1 Coordinates, parameters, and assumptions

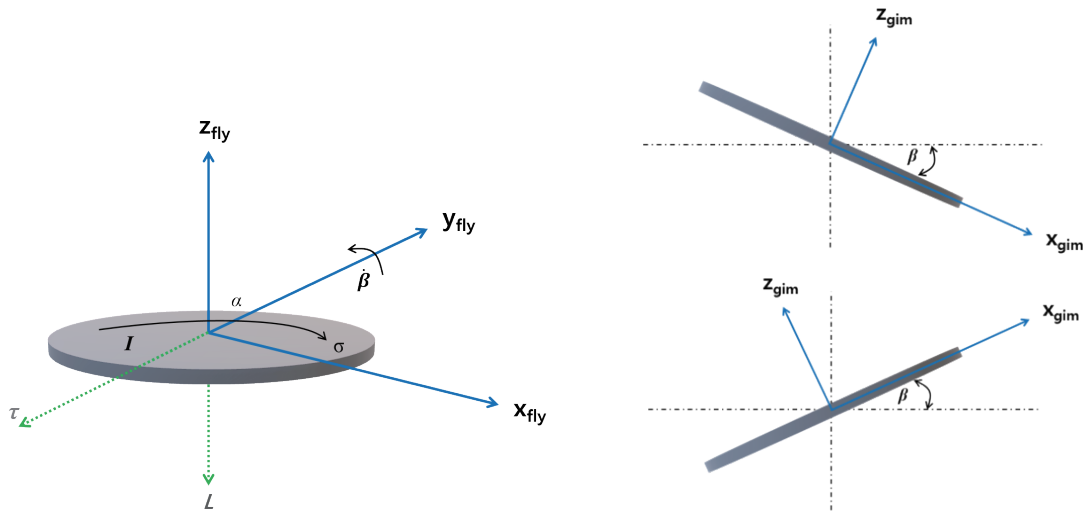
Figure 1(a) is an illustration of an EMBot with a gyroscope attached to control its balance, and Fig. 1(b) shows a front view of the EMBot coordinate system. The coordinate reference or roll angle of the EMBot is denoted by O_h [x_h, y_h, z_h]; the centers of mass of the EMBot, gimbal, and flywheel are O_{em}, O_{gim}, and O_{fly} with heights above the ground of h_{em}, h_{gim}, and h_{fly}, respectively; and the roll angle of the EMBot is denoted by γ. The coordinate system for the gimbal is denoted by O_{gc} [x_{gim}, y_{gim}, z_{gim}], the coordinate system for the moving EMBot is X_{em} [x_{em}, y_{em}, z_{em}], and the coordinate reference for the gimbal angle is denoted by β.

When a flywheel spins at a speed of σ, it has angular momentum denoted by l. The direction of the angular momentum is based on the rotation direction of the flywheel α. As shown in Fig. 2(a), the direction of the angular momentum is along the z_{fly}-axis when the flywheel rotates in



(a) (b)

Fig. 1. (Color online) EMBot with a gyroscope. (a) Definition of coordinate system. (b) Front view of EMBot coordinate system.



(a) (b)

Fig. 2. (Color online) Gyroscopic effect of spinning flywheel. (a) Flywheel coordinate system. (b) Gimbal precession on its axis.

a clockwise direction. The angular momentum of the flywheel will change when a force or a torque is applied to it. We use τ to denote the torque. Then if we apply an external torque along the y_{gim}-axis, the direction of the torque τ will be along the y_{fly}-axis perpendicular to both the force and the radius. Because the CMG can produce torque in the direction of the x_{fly}-axis, the precession torque that is generated by the angular momentum rate of the gimbal $\dot{\beta}$ makes it possible for the EMBot to self-stabilize on its own. Tables 1 and 2 respectively show the definitions for the EMBot rotation system and the notation for the parameters of the system.

Table 1
Definitions for EMBot rotation system.

Parameter	Symbol	Unit
Flywheel spin direction	α	rad
Gimbal angle	β	rad
Roll angle	γ	rad

Table 2
Notation for the parameters.

Parameter	Symbol	Unit
Flywheel mass moment of inertia	I_{fly}	kgm^2
Gimbal mass moment of inertia	I_{gim}	kgm^2
EMBot mass moment of inertia	I_{em}	kgm^2
Flywheel mass & height	m_{fly} and h_{fly}	kg & m
Gimbal mass & height	m_{gim} and h_{gim}	kg & m
EMBot mass & height	m_{em} and h_{em}	kg & m
Flywheel spin velocity	σ	rad/s
Angular velocity of EMBot	ω_{em}	rad/s
Angular velocity of gimbal	ω_{gim}	rad/s
Angular velocity of flywheel	ω_{fly}	rad/s
Gravitational acceleration	g	m/s^2

The following assumptions are taken into consideration in Lagrange's method to obtain a close approximation of a mathematical representation: 1. The EMBot has no longitudinal velocity caused by the gyroscope torque on the vertical axis. 2. The EMBot has no lateral slip at the contact base. 3. The EMBot, gimbal, and flywheel are all point masses at their centers of gravity.

2.2 Modeling of EMBot motion with CMG

The gyroscopic effect is a result of the principle of angular momentum conservation. When the flywheel spins, it generates an angular momentum. The angular momentum l is the product of the moment of inertia I and the spin velocity σ. In particular, the curve, the axis of rotation, and the mass of the flywheel influence the moment of inertia. Thus, the angular momentum l is

$$l = I\sigma. \tag{1}$$

Another important factor is the torque. The angular momentum dl only changes when a torque is applied to it. The torque will cause a change in the angular momentum with respect to time. The torque τ is the cross product of the distance vector and the force vector applied perpendicular to the radius. Thus, the magnitude of the torque τ on the horizontal axis is

$$\tau = \dot{\beta} l \cos\beta. \tag{2}$$

Figure 1 shows the definition of the coordinate system of the EMBot used to describe the system's motion. The coordinate X_{em} rotates about the x_h-axis of O_h through angle γ. The transformation between coordinate X_{em} and O_h is

$$R_{em} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & \sin\gamma \\ 0 & -\sin\gamma & \cos\gamma \end{bmatrix}. \tag{3}$$

The gimbal consists of a frame that holds the flywheel and the gimbal motor, and the frame is molded with the main body of the EMBot. O_{gc} is the center of rotation of the gimbal and F_{gim} rotates about its y_{gim}-axis through angle β, where the transformation is

$$R_{fgim} = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix}. \tag{4}$$

The position of the center of mass of the EMBot O_{em} in the coordinate system X_{em} is

$$r_{em} = \begin{bmatrix} x_{em} & y_{em} & z_{em} \end{bmatrix}^T. \tag{5}$$

In X_{em}, the position of the gimbal joint O_{gc} is

$$r_{gc} = \begin{bmatrix} x_{gc} & y_{gc} & z_{gc} \end{bmatrix}^T. \tag{6}$$

The positions of the gimbal frame's center of mass of O_{gim} and O_{fly} in the F_{gim} coordinate system are, respectively,

$$\begin{aligned} r_{gim} &= \begin{bmatrix} x_{gim} & y_{gim} & z_{gim} \end{bmatrix}^T, \\ r_{fly} &= \begin{bmatrix} x_{fly} & y_{fly} & z_{fly} \end{bmatrix}^T. \end{aligned} \tag{7}$$

Therefore, the positions of the EMBot, gimbal, and flywheel centers of mass in O_h are, respectively,

$$\begin{aligned} r_{hem} &= R_{em}^T r_{em}, \\ r_{hgim} &= R_{em}^T \left(r_{gc} + {R_{fgim}}^T r_{gim} \right), \\ r_{hfly} &= R_{em}^T \left(r_{gc} + {R_{fgim}}^T r_{fly} \right). \end{aligned} \tag{8}$$

The angular velocities of the EMBot, gimbal, and flywheel are calculated as

$$\begin{aligned}\omega_{em} &= \begin{bmatrix}\dot{\gamma} & 0 & 0\end{bmatrix}^T = \dot{\gamma},\\ \omega_{gim} &= R_{fgim}\omega_{em} + \begin{bmatrix}0 & \dot{\beta} & 0\end{bmatrix}^T = \begin{bmatrix}\dot{\gamma}\cos\beta & \dot{\beta} & \dot{\gamma}\sin\beta\end{bmatrix},\\ \omega_{fly} &= \omega_{gim} + \begin{bmatrix}0 & 0 & \sigma\end{bmatrix}^T = \begin{bmatrix}\dot{\gamma}\cos\beta & \dot{\beta} & \sigma + \dot{\gamma}\sin\beta\end{bmatrix}.\end{aligned} \tag{9}$$

The velocities of the EMBot, gimbal, and flywheel are computed as

$$\begin{aligned}v_{em} &= \omega_{em} r_{em},\\ v_{gim} &= \omega_{em} r_{gc} + R_{fgim}{}^T\left(\omega_{gim} r_{gim}\right),\\ v_{fly} &= \omega_{em} r_{gc} + R_{fgim}{}^T\left(\omega_{gim} r_{fly}\right).\end{aligned} \tag{10}$$

From Eqs. (9) and (10), we can derive the kinetic energy of the system as

$$\begin{aligned}KE_{em} &= \frac{1}{2} m_{em} v_{em}{}^T v_{em} + \frac{1}{2}\omega_{em}{}^T I_{em}\omega_{em} = \frac{1}{2}\left(m_{em} v_{em}{}^2 + I_{em}\right)\dot{\gamma}^2,\\ KE_{gim} &= \frac{1}{2} m_{gim} v_{gim}{}^T v_{gim} + \frac{1}{2}\omega_{gim}{}^T I_{gim}\omega_{gim}\\ &= m_{gim} v_{gim}{}^2\dot{\gamma}^2 + I_{gimx}\dot{\gamma}^2\cos^2\beta + I_{gimz}\dot{\gamma}^2\sin^2\beta + I_{gimy}\dot{\beta}^2,\\ KE_{fly} &= \frac{1}{2} m_{fly} v_{fly}{}^T v_{fly} + \frac{1}{2}\omega_{fly}{}^T I_{fly}\omega_{fly}\\ &= \frac{1}{2}\left(m_{fly} v_{fly}{}^2\dot{\gamma}^2 + I_{flyx}\dot{\gamma}^2\cos^2\beta + I_{flyz}\left(\sigma + \dot{\gamma}^2\sin^2\beta\right) + I_{flyy}\beta^2\right).\end{aligned} \tag{11}$$

The total kinetic energy of the system is

$$KE = KE_{em} + KE_{gim} + KE_{fly}, \tag{12}$$

where I_{em} is the moment of inertia of the body mass of the EMBot about its tire contact line (x_h-axis), I_{gim} [I_{gimx}, I_{gimy}, I_{gimz}] is the moment of inertia of the body mass of the gimbal about its x_{gim}, y_{gim}, and z_{gim} axes, and I_{fly} [I_{flyx}, I_{flyy}, I_{flyz}] is the moment of inertia of the body mass of the flywheel about its x_{fly}, y_{fly}, and z_{fly} axes.

Potential energy exists in nonmoving objects and is converted to kinetic energy when a force such as gravity is applied and the object is set in motion by interacting with another object. The potential energies of the EMBot, the gimbal, and the flywheel are described as follows:

$$\begin{aligned}PE_{em} &= m_{em} r_{hem} g\cos\gamma,\\ PE_{gim} &= m_{gim} r_{hgim} g\cos\gamma,\\ PE_{fly} &= m_{fly} r_{hfly} g\cos\gamma.\end{aligned} \tag{13}$$

The total potential energy of the system is

$$PE = PE_{em} + PE_{gim} + PE_{fly}. \tag{14}$$

To take the entire system as a single entity, we applied the Lagrangian equation to Eqs. (12) and (14). The mathematical model of the system is obtained as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i, \tag{15}$$

where L is the Lagrangian function and $L = KE - PE$, q_i is the generalized coordinate of the object, and Q_i is the generalized force applied at each rotating joint. Then we obtain the equations of motion by solving the Lagrangian equation for $q_i = \gamma$. The roll angle dynamic of the EMBot is thus obtained as

$$\ddot{\gamma} = \frac{\left(m_{em}r_{hem} + m_{gim}r_{hgim} + m_{fly}r_{hfly}\right)g\sin\gamma + 2\left(I_{gimx} + I_{flyx} - I_{gimz} - I_{flyz}\right)\dot{\gamma}\dot{\beta}\sin\beta\cos\beta - I_{flyz}\sigma\cos\beta\dot{\beta}}{I_{em} + \left(m_{em}r_{hem}^{\ 2} + m_{gim}r_{hgim}^{\ 2} + m_{fly}r_{hfly}^{\ 2}\right) + \left(I_{gimx} + I_{flyx}\right)\cos^2\beta + \left(I_{gimz} + I_{flyz}\right)\sin^2\beta}, \tag{16}$$

and for $q_i = \beta$, the gimbal angle dynamic of the CMG is

$$\ddot{\beta} = \frac{\left(I_{gimz} + I_{flyz}\right)\dot{\gamma}^2\sin\beta\cos\beta - \left(I_{gimx} + I_{flyx}\right)\dot{\gamma}^2\sin\beta\cos\beta + I_{flyz}\dot{\gamma}\sigma\cos\beta}{\left(I_{gimy} + I_{flyy}\right)}. \tag{17}$$

Equation (16) shows that the movement of the flywheel strongly affects the stability of the vertical position of the EMBot. Also, it can be deduced that as the EMBot's tilt increases, the flywheel is likely to change direction, enabling it to generate more reactive torque. This means that stabilization can occur at a greater tilt angle or if there are larger disturbances. Moreover, from the viewpoint of control, the system dynamics can be explained by the following nonlinear state model:

$$\begin{aligned} \ddot{\gamma} &= f\left(\gamma, \dot{\gamma}, \beta\right) + g\left(u, \dot{\gamma}\right), \\ y &= \left[\gamma, \beta\right], \end{aligned} \tag{18}$$

where

$$f = \frac{\left(m_{em}r_{hem} + m_{gim}r_{hgim} + m_{fly}r_{hfly}\right)g\sin\gamma}{I_{em} + \left(m_{em}r_{hem}^{\ 2} + m_{gim}r_{hgim}^{\ 2} + m_{fly}r_{hfly}^{\ 2}\right) + \left(I_{gimx} + I_{flyx}\right)\cos^2\beta + \left(I_{gimz} + I_{flyz}\right)\sin^2\beta}, \tag{19}$$

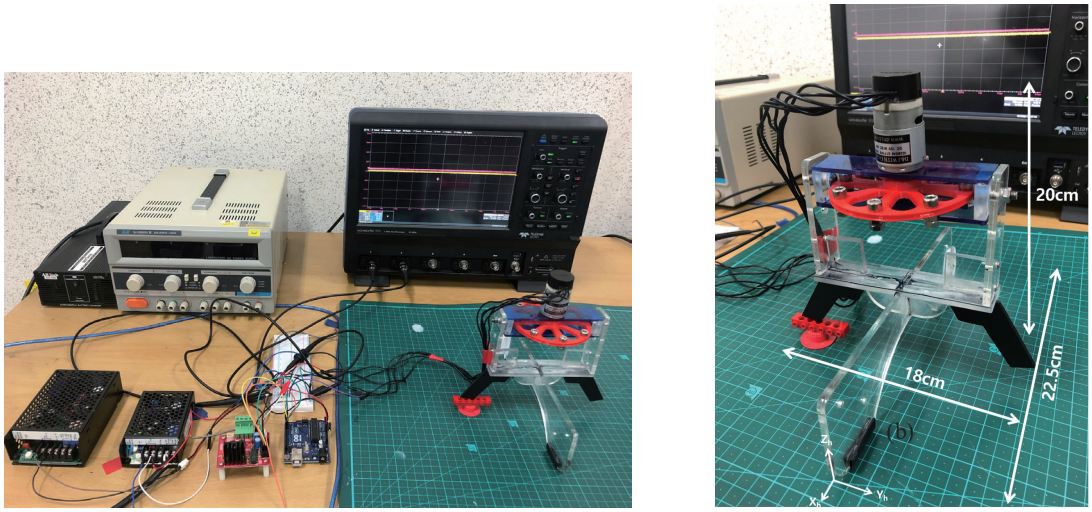
$$g = \frac{2\left(I_{gimx} + I_{flyx} - I_{gimz} - I_{flyz}\right)\dot{\gamma}\dot{\beta}\sin\beta\cos\beta - I_{flyz}\sigma\cos\beta\dot{\beta}}{I_{em} + \left(m_{em}r_{hem}^{\ 2} + m_{gim}r_{hgim}^{\ 2} + m_{fly}r_{hfly}^{\ 2}\right) + \left(I_{gimx} + I_{flyx}\right)\cos^2\beta + \left(I_{gimz} + I_{flyz}\right)\sin^2\beta}u. \tag{20}$$

3. Description of Model-size EMBot and Linearization

3.1 Description of model-size EMBot robot

We built a model-size EMBot, as illustrated in Fig. 3, which is currently being utilized to evaluate various control strategies. The model is 22.5 cm long, 18 cm wide, and 20 cm high to the top of the flywheel motor. Its total weight is 0.319 kg, including the gimbal and the flywheel. The body and the gimbal frame are made of acrylic, whereas the flywheel is made of 3D-printed plastic with a bolt included as ballast. The flywheel is a cylinder with a radius of 4.5 cm and a height of 1.1 cm. Table 3 shows the specifications of the model-size EMBot system.

The base of the acrylic material used to structure the model-size EMBot is shaped like a bicycle tire, and it is coated with rubber to prevent it from sliding. The diameter of the wheels of



(a)

Fig. 3. (Color online) Model-size EMBot and the experimental setup. (a) Experimental setup. (b) Dimensions of EMBot.

Table 3
Specifications of model-size EMBot system.

Parameter	Value	Unit
I_{flyx}, I_{flyy}, I_{flyz}	0.00004472, 0.00004472, 0.000089375	kgm^2
I_{gimx}, I_{gimy}, I_{gimz}	0.0001701, 0.0003402, 0.0001701	kgm^2
I_{em}	0.00638	kgm^2
h_{fly}	0.135	m
h_{gim}	0.148	m
h_{em}	0.11849	m
m_{fly}	0.055	kg
m_{gim}	0.134	kg
m_{em}	0.13	kg
σ	314.159	rad/s
g	9.806	m/s^2

the EMBot is 5 cm and the distance between the axles is 16.5 cm. To prevent the EMBot from tilting by more than 30°, a limiter is placed on each side of the EMBot. To rotate both the flywheel and the gimbal frame, two DC motors with fused gear boxes and integrated encoders are installed. The inclination of the EMBot is read by an inertial sensor connected to a microprocessor for processing. The flywheel, the gimbal motor, and the data processing system including the sensor system are powered by their own independent power sources. Other electronic components, including the processor and power supply, are located outside the EMBot and connected to it with a cable.

3.2 Model linearization

Linearization is a technique for calculating linear approximations of a nonlinear system function that are valid within a restricted range of values around the operational point. Using the Taylor series expansion of the first order, the linear approximation is evaluated to ensure local stability and the steady-state condition at the equilibrium point. This aids in the analysis of the stability and disturbance rejection properties of a system. After linearizing the movement equation around the equilibrium point, the following equation of motion is obtained:

$$\ddot{\gamma} = \frac{\left(m_{em}r_{hem} + m_{gim}r_{hgim} + m_{fly}r_{hf\ ly}\right)g\gamma - I_{flyz}\sigma\dot{\beta}}{I_{em} + \left(m_{em}r_{hem}{}^2 + m_{gim}r_{hgim}{}^2 + m_{fly}r_{hfly}{}^2\right) + \left(I_{gimx} + I_{flyx}\right)}. \tag{21}$$

By expanding Eq. (21), the numerical results for a linear model in state-space form are obtained as

$$\begin{aligned}
A &= \begin{bmatrix} 0 & 1 & 0 \\ \dfrac{\begin{pmatrix} m_{em}\cdot r_{hem} + m_{gim}\cdot r_{hgim} \\ +m_{fly}\cdot r_{hfly} \end{pmatrix} g}{I_{em} + \left(m_{em}\cdot r_{hem}{}^2 + m_{gim}\cdot r_{hgim}{}^2 + m_{fly}\cdot r_{hfly}{}^2\right) + \left(I_{gimx} + I_{flyx}\right)} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
B &= \begin{bmatrix} 0 \\ \dfrac{I_{flyz}\cdot\sigma}{I_{em} + \left(m_{em}.r_{hem}{}^2 + m_{gim}.r_{hgim}{}^2 + m_{fly}.r_{hfly}{}^2\right) + \left(I_{gimx} + I_{flyx}\right)} \\ 1 \end{bmatrix} u, \\
C &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \\
D &= \begin{bmatrix} 0 \end{bmatrix}.
\end{aligned} \tag{22}$$

The specifications of the EMBot listed in Table 3 are determined from the model-size EMBot hardware system shown in Fig. 3(b). By substituting these values into Eq. (22), a MATLAB

simulation of the behavior of the system is performed. An examination of poles and zeros found that the uncompensated system has two poles but no zeros. One of the poles is a positive number, which means that the pole on the right side of the imaginary axis causes the system to become unstable.

4. Control System Implementation and Analysis

Since we are attempting to control the position of the EMBot, which should return to the vertical position after the inclination, the reference signal we are tracking should be zero. Various full-state feedback control methods, mainly the PP and LQR control methods, were implemented and analyzed for the control system. The scheme of the PP and LQR control system is shown in Fig. 4(b).

For a classical feedback controller such as a PID controller, when the input u and output y are known, the objective is to build a feedback control system that changes the output y to a desired value. Then, in the process block section, we compare the actual value of the output signal with the reference signal to obtain a new control signal. Then, this new control signal is fed as the input signal to control the plant block section with the aim of reducing the error to zero. Instead of feeding back the output y for the PP control, we feed back every state variable in the state vectors. In this case, we were convinced of the importance of each state value. The state values are then multiplied by a matrix comprised a variety of gain values, also known as the gain matrix. Then the result is subtracted from the scale of the reference signal. This signal is then directly supplied as an input to the system. The controller comprises the entire block of the feedback system.

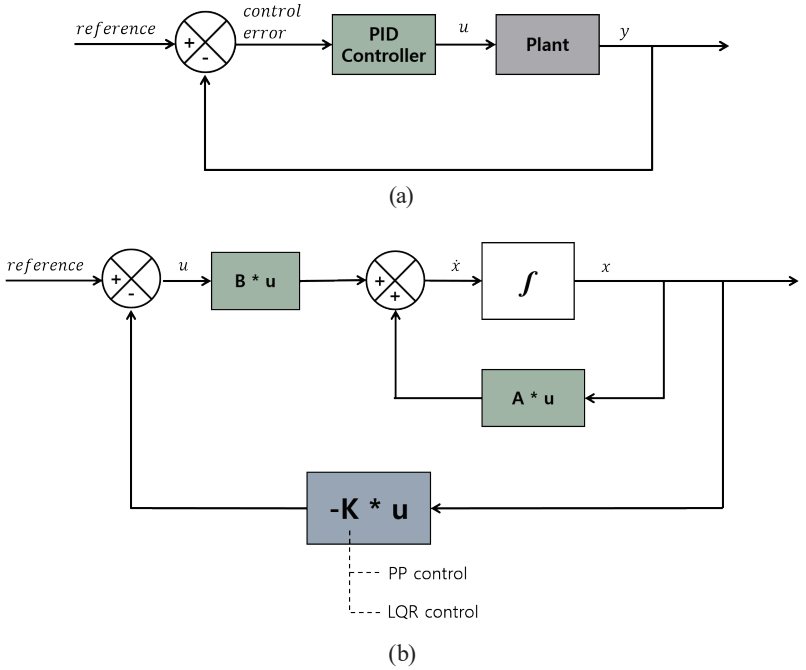


Fig. 4. (Color online) PID vs full-state feedback control scheme. (a) PID controller. (b) PP and LQR controllers.

As previously mentioned, using the PP control method, we can compute the appropriate gain matrix to ensure system stability. A reference scaling term is also utilized to guarantee sufficient steady-state performance. In addition, rather than randomly selecting poles in the PP controller method, the LQR method provides a more systematic approach through finding the optimal *K*-matrix value by selecting the closed-loop characteristics. This makes LQR control one of the most powerful algorithms for controlling a dynamic system at the lowest possible cost, and it is extensively applied in many fields of industry control.

4.1 PP controller

The state-space equation of the system represents the multiple *n* th-order input linear time-invariant systems as described by

$$\dot{x} = Ax + Bu, \tag{23}$$

where Ax describes the dynamics of the system and Bu describes how the system responds to an input. In addition, the Ax term describes how energy is stored and transferred in the system. Therefore, it is reasonable to assume that the matrix Ax is unique for each controller design. This is particularly relevant in terms of system stability, because each feedback controller must update matrix A to adapt to the system's dynamics. The eigenvalues of matrix A are equal to the system's poles, and the location of the poles determines the stability of the linear system. This is the crucial aspect of the PP controller. The poles or the eigenvalues of the closed-loop matrix A are modified to ensure the stability of the closed-loop system. The closed-loop system is created by submitting state variables back through a real constant matrix K:

$$\dot{x} = \left(A - BK\right)x. \tag{24}$$

As discussed previously, after running the system, one of the eigenvalues appears on the positive side of the imaginary axis in the open-loop system, causing the system to become unstable. Using the PP controller by moving the unstable pole to the left half-plane will stabilize the system.

From our closed-loop matrix A, which is $(A - BK)x$, and our gain matrix K, which is a 1×3 matrix, the following expression can be derived:

$$\left(A - BK\right)x = \begin{bmatrix} 0 & 1 & 0 \\ 69.98 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 4.697 & 1 \end{bmatrix}^T \cdot \begin{bmatrix} K_1 & K_2 & K_3 \end{bmatrix}. \tag{25}$$

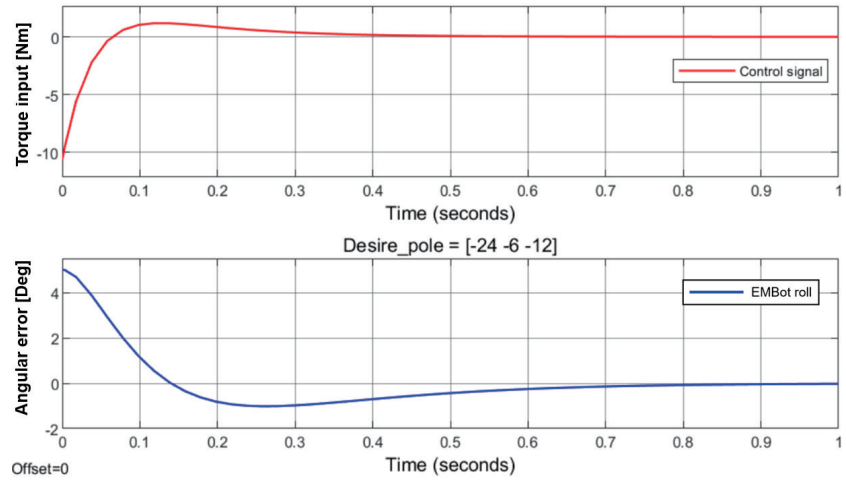
In this case, we choose the eigenvalues $\lambda = \begin{bmatrix} -24 & -6 & -12 \end{bmatrix}^T$. Using these eigenvectors, we can solve the eigenvalues of the closed-loop system by finding the characteristic equation of the three gain values

$$\det\left(\left(A-BK\right)x-\lambda I\right)=0, \tag{26}$$

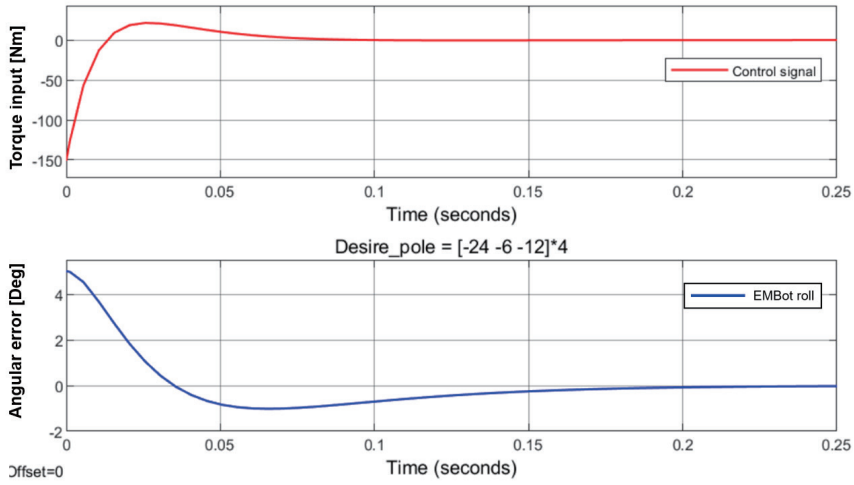
and we find appropriate values of K_1, K_2, and K_3 as follows:

$$K=[122.1954 \quad 14.1979 \quad -24.6913]. \tag{27}$$

Using these three gain values in the path of the system will stabilize the system as shown in Fig. 5. Using the root locus method, it is only possible to move the poles along the locus lines by adjusting the gain. However, using the PP control system, we have a gain matrix with which the poles can be moved anywhere in the complex plane, as shown in Fig. 5. We first choose the desired pole location of [−24 −6 −12], for which we obtain the response system at T_r = 0.120 s and



(a)



(b)

Fig. 5. (Color online) Response system with PP control. (a) Roll angle response at desired pole position [−24 −6 −12]. (b) Roll angle response at desired pole position [−24 −6 −12] × 4.

T_s = 0.899 s, and if we attempt to move a large number of eigenvalues extremely far to the left, we achieve a rapid reaction, as shown in Fig. 5(b).

There are some deficiencies in choosing the poles in the PP method, especially for a high-order system. We wish to retain the poles closest to the imaginary axis, but this causes the system to behave as a conventional second-order system, because the poles move more slowly and typically govern the system's response. Attempting to move a large number of eigenvalues extremely far to the left of the imaginary axis to achieve a rapid reaction revealed that our actuators lack the necessary speed or power to produce the desired response. This is because it requires greater actuator torque to move the eigenvalues from their initial position. Lastly, in our mathematical model, we communicate back every state, but we are unable to do this in practice. To solve this problem, sensors will be added to the system to measure the missing states. In contrast, rather than randomly selecting poles in the PP control method, the LQR control method is expected to be a more systematic process.

4.2 LQR controller

The PP and LQR methods have exactly the same physical structure. Also, the key to their implementation is the same, but how the K value is chosen is different. Using the PP controller, we determined the K value by deciding where we wanted to place the closed-loop poles. However, with this method, it is necessary to find the optimal location for the closed-loop poles, which is essential for high-order systems or systems with multiple actuators. Thus, with the LQR method, we do not choose the locations of the poles. Instead, we find the best K-matrix value by choosing the closed-loop characteristics that we consider important.

As discussed previously, the linear time-invariant system was expressed in the state-space form (Eq. (23)), and the state variable of the feedback control can be found from Eq. (24). Next, the quadratic cost function for the system is defined as

$$J = \int \left(x^T Q x + u^T R u \right) dt, \tag{28}$$

where Q is a positive definite or positive semi-definite symmetric matrix when used in conjunction with R, a symmetric positive definite matrix. Both matrices are weight matrices, and each of them should be diagonal. Each element of the weight matrix has an associated value that reflects its effect on the performance index J. Here, we adjust Q to penalize the performance and adjust R to penalize the amplitude of the actuator.

From the state-space model system described in Eq. (22), we design a full-state feedback control using LQR with the identity matrix as the starting matrix for Q. The first diagonal entry corresponds to the EMBot angular error, and the second is related to the angular rate of the EMBot. There is also an actuation input for the input rotation torque system. These components work together to create the torque command. Therefore, R is a single value. Then the optimal feedback is obtained using the LQR command and from the state-space object that represents the closed-loop dynamics. Finally, using the designed controller, we can observe the response time of the system with an initial tilt position of 0.087 rad or 5°. Figure 6 shows the performance of the control system in keeping the EMBot balanced in the vertical position.

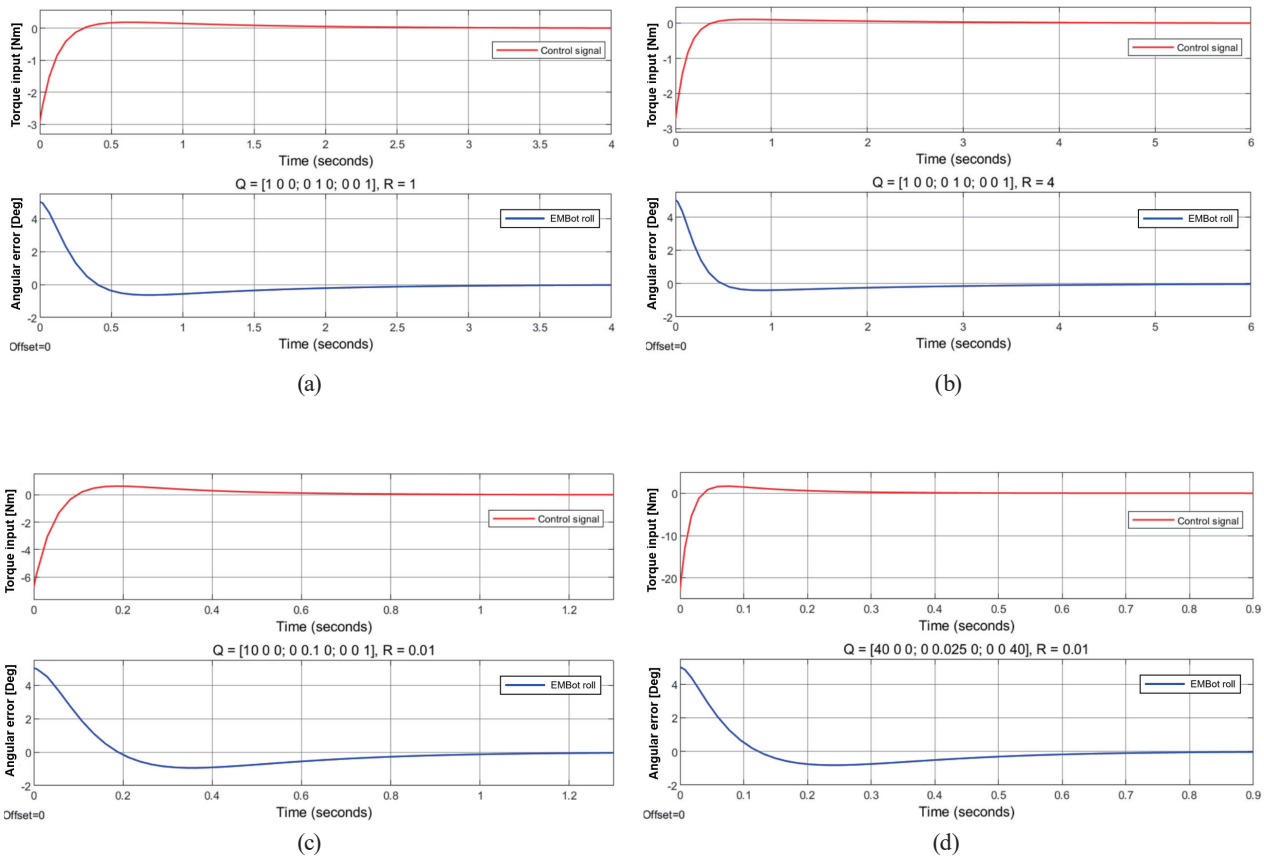


Fig. 6. (Color online) Response system with LQR control showing roll angle responses for (a) Q = [1 0 0; 0 1 0; 0 0 1] and R = 1, (b) Q = [1 0 0; 0 1 0; 0 0 1] and R = 4, (c) Q = [10 0 0; 0 0.1 0; 0 0 1] and R = 0.01, and (d) Q = [40 0 0; 0 0.025 0; 0 0 40] and R = 0.01.

As shown in Figs. 6(a)–6(d), the system with different Q and R values has different response times of T_r = 0.302 s and T_s = 3.602 s, T_r = 0.383 s and T_s = 5.504 s, T_r = 0.161 s and T_s = 1.284 s, and T_r = 0.102 s and T_s = 0.844 s, respectively, with zero steady-state error in each case. When the control signals are active, they generate a torque that accelerates the EMBot over time; therefore, the torque is proportional to the error's integral. The longer the acceleratation, the greater the torque employed. Utilizing less torque and increasing the penalty on R by a factor of four results in a slower response. Moreover, by penalizing the angular rate portion of the Q matrix and the R value in more aggressive tuning, as respectively shown in Figs. 6(c) and 6(d), we obtain a much faster reaction of the EMBot roll response that can meet the requirements of our response system. As shown in Fig. 7, for different initial inclinations, we obtain different EMBot roll responses. Finally, these optimal values of the LQR controller then can be supplied to the embedded processor of the EMBot to maintain the stability.

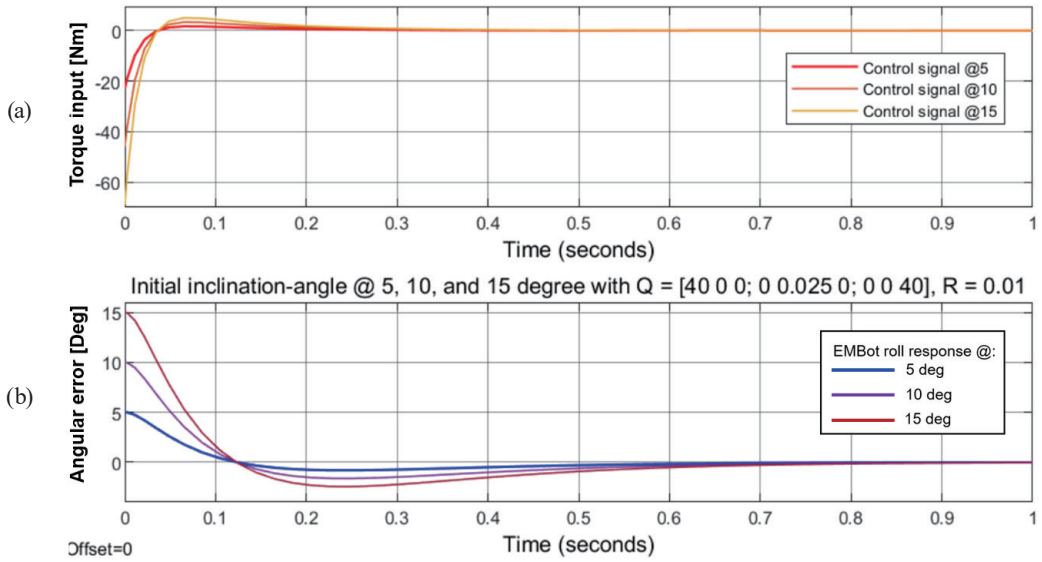


Fig. 7. (Color online) Response system with LQR control for different initial inclination angles.

5. Conclusions

In this study, we designed the posture control stabilization system of the EMBot using a CMG. A reduced mechanical model of the EMBot was designed, which included an inertial sensor to provide the input data of the EMBot's roll angle with the necessary components of the system. The EMBot system was also modeled, where Lagrange's equation was used to derive the nonlinear mathematical model of the EMBot in a generalized coordinate system. As a control system, the PP and LQR control methods were designed to simulate and evaluate the performance of the EMBot system. The experimental results showed that both control methods have a good response and the ability to return the EMBot to the equilibrium position after it has been tilted. However, the proposed LQR controller ensures that the EMBot has less vibration, a faster response, and greater robustness than the PP method. This is because the optimal gain value was determined by determining the closed-loop characteristic rather than by choosing the pole position. The LQR controller designed in this study is expected to be applied to the actual EMBot.

Acknowledgments

This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the ICT Challenge and Advanced Network of HRD (ICAN) program (IITP-2022-2020-0-01825) supervised by the Institute of Information & Communications Technology Planning & Evaluation (IITP).

References

1 K. J. Astrom, R. E. Klein, and A. Lennartsson: IEEE Control Syst. Mag. **25** (2005) 26. https://doi.org/10.1109/MCS.2005.1499389
2 H. Fu: Bull. Jpn. Soc. Mech. Eng. **9** (1966) 284. https://doi.org/10.1299/jsme1958.9.284
3 L. Guo, Q. Liao, S. Wei, and Y. Huang: The 2010 IEEE Int. Conf. Information and Automation (IEEE, 2010) 1613-1617. https://doi.org/10.1109/ICINFA.2010.5512251
4 C. Xiong, Z. Huang, W. Gu, Q. Pan, Y. Liu, X. Li, and E. X. Wang: 2018 3rd Int. Conf. Robotics and Automation Engineering (IEEE, 2018) 24-28. https://doi.org/10.1109/ICRAE.2018.8586765
5 C. H. Chiu and C. Y. Wu: IEEE Access **8** (2020) 84837. https://doi.org/10.1109/ACCESS.2020.2992792
6 H. Jin, Y. Zhang, H. Zhang, Z. Liu, Y. Liu, Y. Zhu, and J. Zhao: Math. Problems Eng. (2018). https://doi.org/10.1155/2018/5240594
7 N. K. Vu, H. Q. Nguyen: Math. Problems in Eng. (2020). https://doi.org/10.1155/2020/6724382
8 D. A. Bravo M., C. F. Rengifo R., and J. F. Díaz O.: IEEE 4th Colombian Conf. Automatic Control (IEEE, 2019) 1–6. https://doi.org/10.1109/CCAC.2019.8920870
9 S. C. Spry and A. R. Girard: Vehicle Syst. Dynamics **46** (2008) 247. https://doi.org/10.1080/00423110801935863
10 M. H. Shafiei, and M. Emami: Syst. Sci. Control Eng. **7** (2019) 12. https://doi.org/10.1080/21642583.2018.1555062
11 S. H. Park and S. Y. Yi: Int. J. Control Autom. Syst. **18** (2020) 217. https://doi.org/10.1007/s12555-018-0749-7
12 D. Karnopp: Vehicle Syst. Dynamics. **37** (2002) 145. https://doi.org/10.1076/vesd.37.2.145.3535
13 R. Lot and J. Fleming: Vehicle Syst. Dynamics. **57** (2018) 1. https://doi.org/10.1080/00423114.2018.1506588
14 P. Gogoi, M. Nath, B. T. Doley, A. Boruah, and H. J. Barman: Int. J. Eng. Tech. **9** (2017) 2051. https://doi.org/10.21817/ijet/2017/v9i3/1709030206
15 D. Horla and A. Owczarkowski: 2015 Int. Conf. Industrial Eng. and Syst. Management (IEEE, 2015) 998–1003. https://doi.org/10.1109/IESM.2015.7380276
16 P. Y. Lam: 2011 IEEE 5th Int. Conf. Cybernetics and Intelligent Syst. (IEEE, 2011) 247–252. https://doi.org/10.1109/ICCIS.2011.6070336

About the Authors

Rodi Hartono is an electrical engineer with a strong interest and ability in AI convergence, control systems, robotics, image recognition, deep learning, and automation systems. He has worked extensively in the field of robotics and intelligent systems and has more than 12 years of experience working as lecturer and a full-time researcher. He is also responsible for managing and supervising the Indonesian Computer University robotics team to perform research and build computing robots that have been entered in regional, national, and international robot competitions. He also designs, researches, and builds robotics products required by the industrial community. He received his bachelor's degree from Indonesian Computer University, Indonesia, in 2010 and his master's degree from the School of Electrical and Informatics Engineering, Bandung Institute of Technology, Indonesia, in 2014. Since 2021, he has been pursuing a doctor's degree at the Department of Artificial Intelligence Convergence, Busan University of Foreign Studies, Republic of Korea. (rodihartonoo@gmail.com)

Seung Bin Kim received his bachelor's degree from Busan University of Foreign Studies, Republic of Korea, in 2021. Since 2021, he has been pursuing his master's program at the Department of Artificial Intelligence Convergence, Busan University of Foreign Studies. He has a strong interest in AI convergence, image recognition, deep learning, and IoT. (skbbq123@naver.com)

Patrick Tshibang A Kalend received his bachelor's degree in computer science engineering, focusing on the field of information system engineering, from Université Protestante de Lubumbashi, Democratic Republic of Congo, in 2016. After his graduation, he presented lectures and led students in their projects for graduation. Since 2022, he has been pursuing his master's degree at the Department of Artificial Intelligence Convergence, Busan University of Foreign Studies. He has a strong interest in AI, computer vision, robotics, and IoT. (patricktshibang7@gmail.com)

Nam Kyun Baik is a professor in the Department of Smart Convergence Security, Busan University of Foreign Studies. He received his B.S., M.S., and Ph.D. degrees from the School of Electronic Engineering, Soongsil University. From 2000 to 2017, he was a senior researcher at Korea Internet & Security Agency. He has a strong interest in convergence security, security consulting, information security management systems, AI security, and IoT security. (namkyun@bufs.ac.kr)

Kyoo Jae Shin is a professor of intelligent robot science at Busan University of Foreign Studies, South Korea. He is the director of Future Creative Science Research Institute at BUFS. He received his B.S. degree in electronics engineering in 1985, his M.S. degree in electrical engineering from Cheonbuk National University in 1988, and his Ph.D. degree in electrical science from Pusan National University in 2009. Dr. Shin has been a professor at Navy Technical Education School and a main director and research associate of dynamic stabilization systems at Busan Defense Weapon Research Institute. He has also researched and developed a fish robot, a submarine robot, an automatic dog spaying robot operating in a glass room, an automatic milking robot using a manipulator, a personal electrical vehicle, a smart accumulated aquarium using a heat pump, a solar tracking system, a 3D hologram system, and a gun/turret stabilization system. He is interested in intelligent robots, image signal processing application systems, smart farms, and aquariums using new energy and IoT technology. (kyoojae@bufs.ac.kr)