Efficient Modeling and Posture Control of an Inverted-Pendulum Mobile Robot by Control Command Fusion

Taeseok Jin

Department of Mechatronics Engineering, Dongseo University
jints@dongseo.ac.kr

Abstract

Research on inverted-pendulum mobile robots (IPMRs), commonly known as "balancing robots," has gained momentum during the past decade in a number of robotics laboratories worldwide. This paper proposes a hardware design for such a robot, as well as control command fusion based on a new balancing algorithm to minimize the number of sensors necessary for the posture balancing of the inverted mobile robot while maintaining the same level of commercial performance.

The objective of the design is to develop an IPMR as well as a MATLAB interfacing configuration to be used as a flexible platform using an embedded unstable linear plant intended for research and teaching purposes. Such issues as the selection of actuators and sensors, signal processing units, MATLAB Real Time Workshop coding, modeling, and the control scheme are addressed and discussed. The system was tested using a state feedback controller to verify its functionality.

Keywords: Mobile robot, Posture balancing, Gyro sensor, BLDC motor

1. Introduction

Inverted balancing control has been the subject of numerous studies in automatic control since the 1950s. In this paper, the development of a theme inspired by the well-known Segway robot is described. This kind of balancing robot has been a source of much interest recently [1]. This is because it provides rich opportunities for the application of control design, signal processing, distributed control systems, and consideration of real-time constraints for implementation issues. A scaled-down prototype of a digital-signal-processor-controlled two–wheeled vehicle called Quasimoro and JOE based on an inverted pendulum with weights attached to the system to simulate a human driver was considered in Ref. [2]. A linear state space controller utilizing sensory information from a gyroscope and motor encoders was used to stabilize the system.

A small wagon built utilizing this technology enabled humans to travel short distances in a narrow path or factories instead of using cars or buggies that are more polluting [3].

The trajectory control for this type of robot was successfully implemented in two dimensions [4]. The proposed trajectory control algorithm could make the mobile robot move autonomously, albeit slowly. The work was further extended in Ref. [5], where the robot was assumed to receive similar input on both wheels, and the mathematical modeling is represented in a one-dimensional plane system. Based on the successful result described in Ref. [5], similar studies were performed that employed proportional–integral–derivative (PID) and linear-quadratic-regulator controllers [6,7]. The dynamic model of a two-wheeled inverted pendulum was analyzed for controllability, and a ball-driving wheel robot was developed.

This paper reports on multibody simulation and control of an inverted-pendulum mobile robot (IPMR). An approximation model of the system was modeled by using MATLAB

software. The MATLAB model had two inputs (disturbance and control) and two outputs (pendulum angle and wheel position). Pendulum body angle, which is the most important parameter, needs to be controlled by feedback. A PID controller was designed and implemented for balancing control of the pendulum angle. A controller design procedure and the controller efficiency are shown graphically. The IPMR system can be simulated and controlled in MATLAB [8].
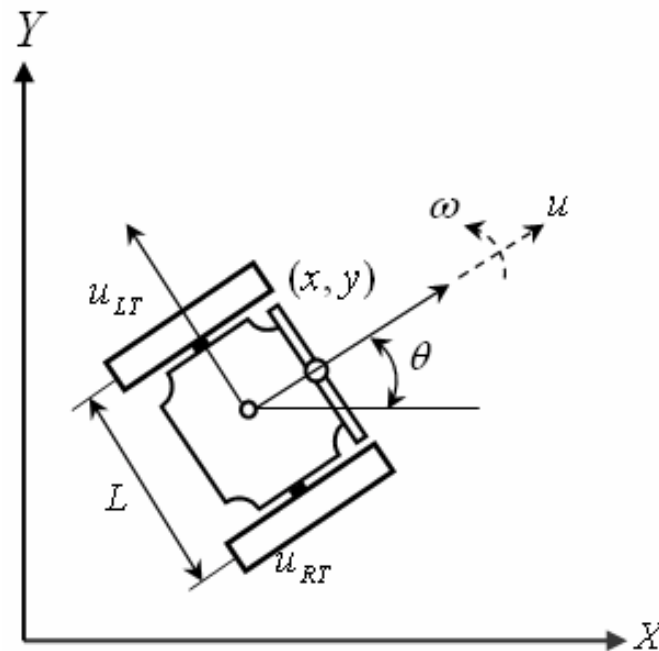
2. Modeling and Control

Figure 1. Kinematics of the Inverted Mobile Pendulum

An inverted mobile pendulum can be modeled as linear and angular velocities, as in Refs. [9-10]:

$$u = \frac{u_{LT} - u_{RT}}{2} \tag{1}$$

$$\omega = \frac{u_{LT} - u_{RT}}{L} \tag{2}$$

Figure 1 illustrates the mobile robot coordinates, and a Jacobian matrix can be derived from the coordinates as follows.

To estimate a process with nonlinear difference and measurement relationships, new governing linear equations can be derived based on the nonlinear equations in Eqs. (1) and (2).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} u \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{4}$$

where u is the linear velocity of the mobile robot and ω is the angular velocity of the mobile robot along the center axis.

The linear and angular velocity of the mobile robot can also be represented in terms of the left- and right-wheel velocities.

$$\begin{bmatrix} u \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} u_{LT} \\ u_{RT} \end{bmatrix} \tag{5}$$

where L represents the distance between two wheels. The position of the mobile robot can basically be calculated by using the encoder signal. The linear velocity and angular velocity of the mobile robot are transformed back to each wheel's velocity through the inverse kinematics.

The modeling and controlling strategy and system model can be seen in Figure 2. To obtain the dynamic model of the IPMR, computer-aided design (CAD) and MATLAB software were used. The boundary conditions and dynamic loadings were determined in Adams. Thus, a more realistic model was derived. It was important that the CAD cooperate with MATLAB.

A general block diagram including system modeling and controller design can be seen in Figure 2. The system is a multiple-input multiple-output structure that consists of two inputs and two outputs. The inputs are defined as controller (torque, N.m) and disturbance (tilt, θ_{TILT}). Similarly, outputs are defined as the pendulum slant angle (deg) and wheel position (m). Here, the pendulum angle, which is the most important parameter, needs to be controlled by feedback. Wheel position is analyzed to view only the motion trajectory.
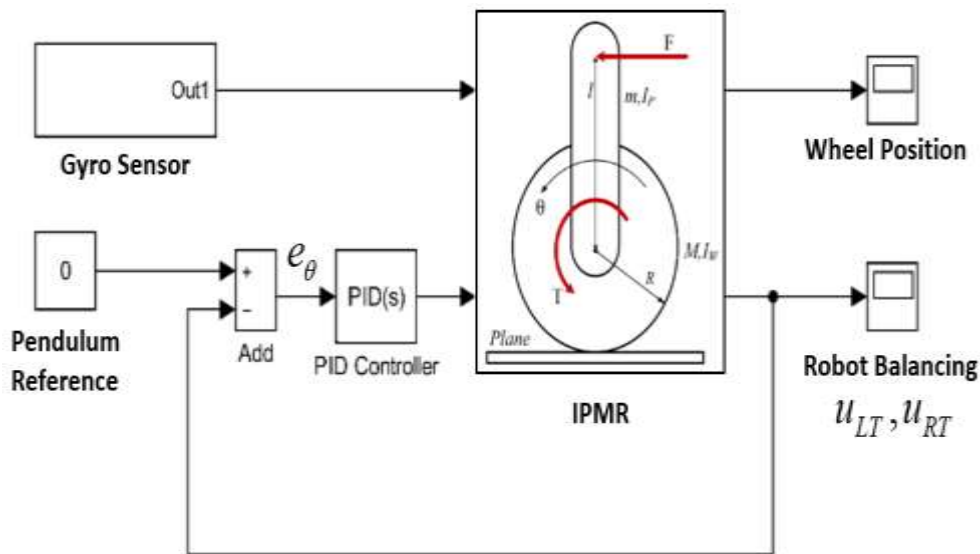

Figure 2. Block Diagram of IPMR System

3. Balancing Control System

The controller detects the dynamic tilt of the inverted mobile pendulum using a gyro sensor and maintains the horizontal balance by rotating the brushless DC (BLDC) motor to the tilted direction to obtain the reaction force. An intelligent tilting and steering algorithm was implemented to improve the safety when a human is loaded on the inverted mobile pendulum.

3.1. Controller Architecture

Figure 3 shows the construction of the controller, which combines three control commands: compensation and error, PID, and balancing control. The compensation and error control command performs the accumulative error compensation function based on the dynamic tilt compensation algorithm in real time, and generates the error between the target and the real values of the tilt for PID control purposes. The second block is the PID controller and control input. The third block is the balancing controller, which determines leftward and rightward rotation ratios according to the knob manipulation angle on the inverted mobile pendulum for motor driving control.
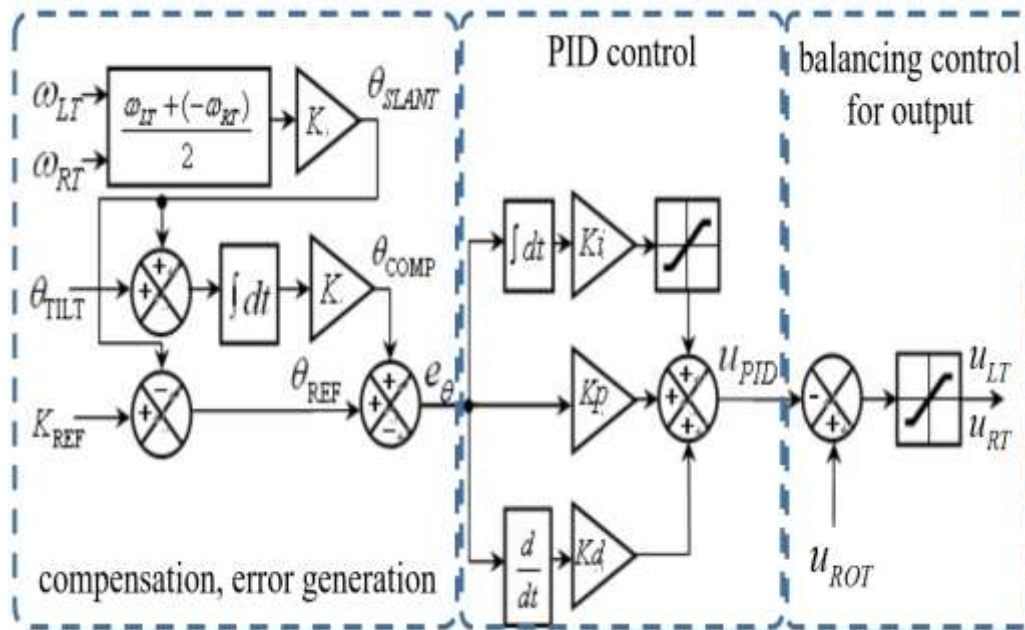

Figure 3. Schematic Diagram of the Controller

3.2. Block of Dynamic Tilt Compensation and Errors

When riding an inverted mobile pendulum, a human is vulnerable to falling forward, because the motor torque becomes weak because of the back electromagnetic frequency when the speed is high. To overcome this problem, the inverted mobile pendulum is designed to be tilted backward when it is moving fast. This controller also limits the maximum velocity by the slant angle, which is calculated as

$$\theta_{SLANT}[k] = K \frac{\omega_{LT}[k] - \omega_{RT}[k]}{2} \tag{7}$$

where $\omega_{LT}[k]$ and $\omega_{RT}[k]$ represent left and right wheel velocities, respectively, and the tilt compensation constant, K, can be determined empirically.

The reference angle for the inverted mobile pendulum, θ_{REF}, is zero degrees at the static state. However, it needs to be adjusted as

$$\theta_{REF}[k] = K_{REF} + \theta_{SLANT}[k] \tag{8}$$

To provide safety in riding, the reference angle constant, K_{REF}, is normally 0 to maintain the horizontal balance of the IPMR. Because there is the nonzero slant angle, the accumulative error for the gyro sensor needs to be modified as

$$\theta_{COMP}[k] = K \sum_{j=1}^{k} (\theta_{TILT}[j] + \theta_{SLANT}[j]) \qquad (9)$$

where the tilt sensor angle, θ_{TILT}, becomes zero when the IPMR keeps horizontal balance, and the gain constant, K, is determined empirically. At the normal state, the slant angle, θ_{SLANT}, and the tilt sensor angle, θ_{TILT}, have the same size and the opposite sign. Therefore, $\theta_{TILT} + \theta_{SLANT} = 0$ when there is no gyro sensor error.

Finally, the error, e_θ, to the target angle, θ_{REF}, for PID control can be defined as

$$e_\theta[k] = \theta_{REF}[k] - \theta_{GYRO}[k] + \theta_{COMP}[k] \qquad (10)$$

where θ_{GYRO} represents the gyro sensor angle, and θ_{COMP} is added to compensate for the accumulative error of the gyro sensor.

3.3. Block of PID Control

The second control command represents the PID controller, and the control input can be obtained as

$$u_{PID}[k] = K_P e_\theta[k] + K_I \sum_{j=1}^{k} e_\theta[j] + K_D (e_\theta[k] - e_\theta[k-1]) \qquad (11)$$

where K_P, K_I, and K_D represent the proportional, integral, and derivative gains, respectively. The gains are determined heuristically during the experiments.

3.4. Block of Balancing Control

The balancing control command determines outputs of the left and right wheel motors to rotate the inverted mobile pendulum. The left- and right-wheel outputs, u_{LT} and u_{RT}, are represented as

$$
\begin{aligned}
u_{LT}[k] &= u_{ROT}[k] + u_{PID}[k] \\
u_{RT}[k] &= u_{ROT}[k] - u_{PID}[k]
\end{aligned}
\qquad (12)
$$

where u_{PID} represents the right-wheel motor PID output, and u_{ROT} represents the rotation output. When the IPMR is driving without rotation, the left output, u_{LT}, is increasing while the right output, u_{RT}, is decreasing, because the left and right wheels are rotating in opposite directions.

When the IPMR is stationary without driving, the PID output, u_{PID}, becomes near zero. If the rotation output, u_{ROT}, is given at this moment, the IPMR can rotate $180°$ at the same location

Finally. The control input is limited within the range of motor maximum torque, and it is sent to the controller through Serial Peripheral Interface bus communication to control the BLDC motor.

Also, this control command represents the steering control algorithm, and the steering sensitivity, $\alpha_{KNOB\ GAIN}$, which ranges from 0 to 1, can be defined as

$$\alpha_{KNOB\ GAIN}[k] = \frac{K_{MAX} - \left(\dfrac{|\omega_{LT}[k]| + \|\omega_{RT}[k]|}{2}\right)}{K_{MAX}} \tag{13}$$

where K_{MAX} represents the maximum speed constant, and it needs to be kept higher than the maximum speed of the IPMR, ω_{MAX}. The absolute values are taken to apply for forward and backward motions. When the velocity is high, the steering sensitivity is low, which keeps the smooth rotation motion when the IPMR is moving fast.

The rotation output of the IPMR, u_{ROT}, can be determined by using the steering sensitivity, $\alpha_{KNOB\ GAIN}$, as

$$u_{ROT}[k] = K\left(\theta_{KNOB}[k] - K_{KNOB\ REF}\right)\alpha_{KNOB\ GAIN} \tag{14}$$

For the left rotation, u_{ROT} has a negative value, whereas, for the right rotation, it has a positive value. The steering angle, θ_{KNOB}, is manually controlled by the knob attached at the upper part of the IPMR.

4. Experiments

IPMR models, both real system and simulation, cannot be validated directly, owing to the open-loop unstable system. It is not possible to analyze the process response using a step change, because the robot cannot stay balanced unless a controller is used. To test the mathematical models against the real robot, a PID controller was implemented in the Beauto Balance robot.

In Figure 4 are depicted the speed of the θ_{SLANT} angle for the robot and mathematical models.
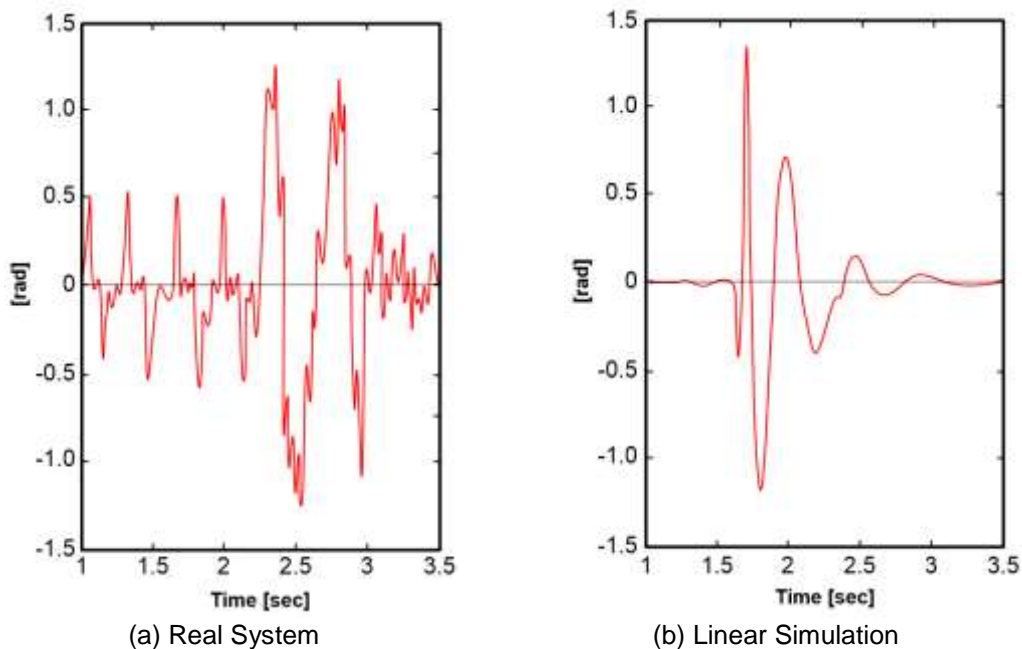

(a) Real System　　　　　(b) Linear Simulation

Figure 4. θ_{SLANT} for Real Systems and Simulation

When an upset occurs, it affects the angle θ that represents the angular displacement of the tires in an instant of time. The angular displacement of the wheels can be seen in Figure 5.
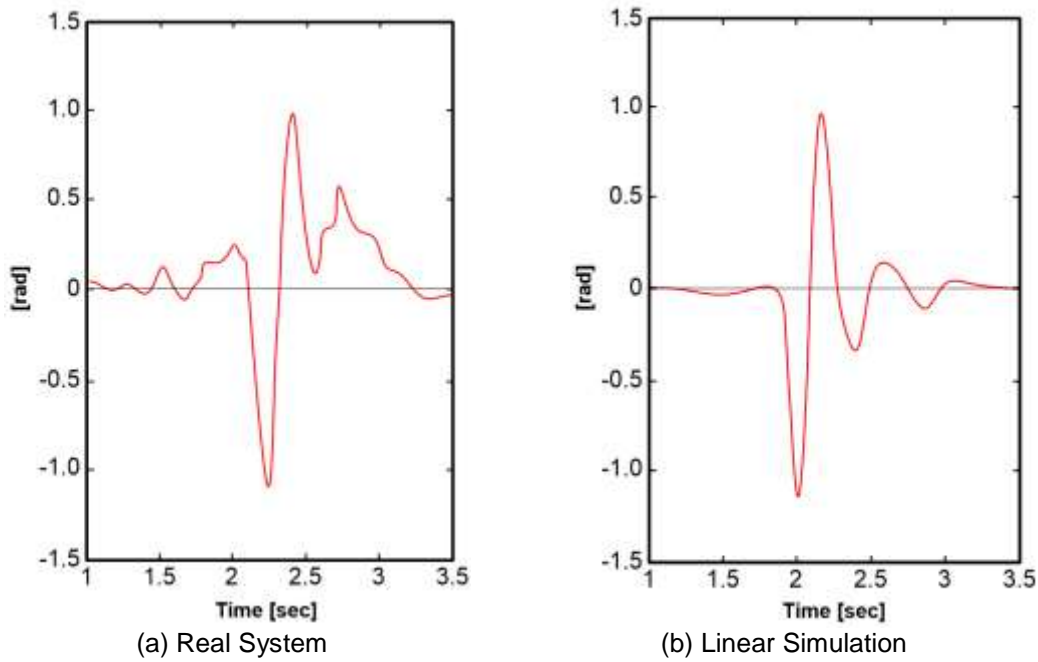

(a) Real System   (b) Linear Simulation

Figure 5. θ for Real Systems and Simulation

An external force was exerted on the back of the robot. The disturbance was considered a constant force to keep a deviation of -0.11 rad in (θ_{SLANT}) angle.
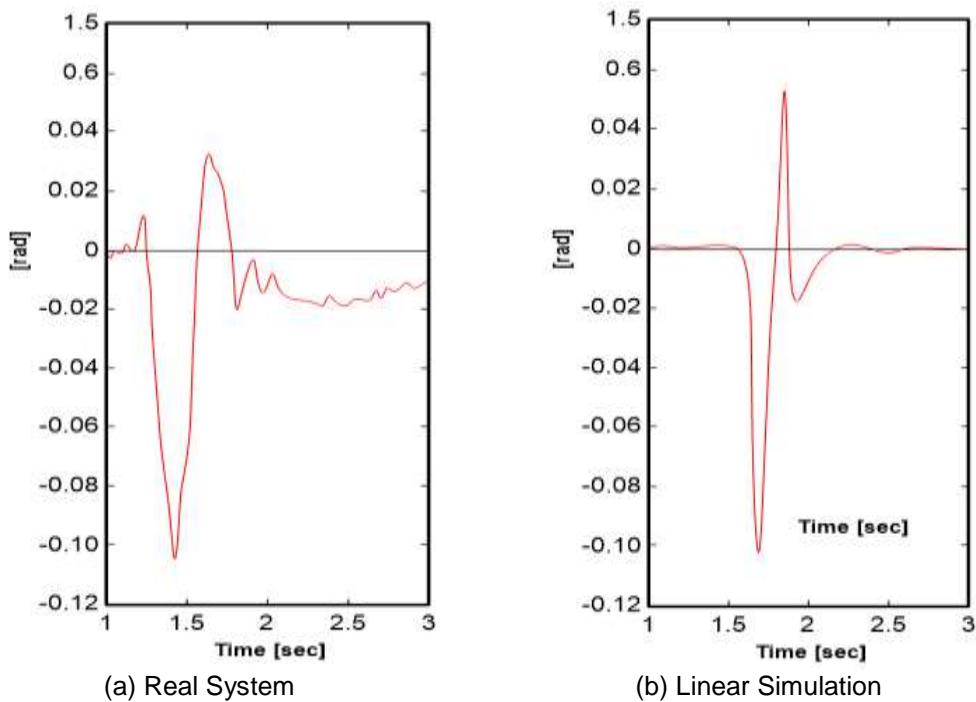

(a) Real System   (b) Linear Simulation

Figure 6. Slant Angle (θ_{SLANT}) for (a) the Real System and (b) Simulation: the Robot Behaviors under an External Back Force

The slant angle (θ_{SLANT}) under the presence of the external disturbance force applied on the back of the robot behaves as presented in Figure 6; the plot (a) represents the real system, and (b) the simulation for dynamic balancing behavior of the robot with an external disturbance force.

The IPMR also demonstrates the ability to recover from external disturbance force applied to the body. Figure 7 shows the sequence of the dynamic behavior of the robot with external disturbance applied by pushing the robot body by hand when the robot moves forward and when it moves back. The plots show the slant angle when the disturbance was applied to the robot, and it was able to recover to its upright position. Because of the limitation of the actuators, the robot can handle a maximum of 0.1-rad tilt and a small amount of force.
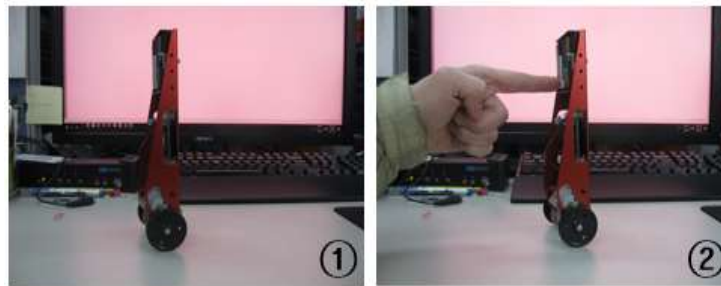
(a) Dynamic Behavior of the Robot when the Robot Moves Forward
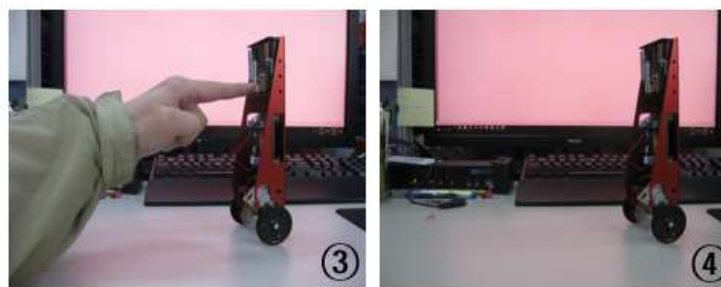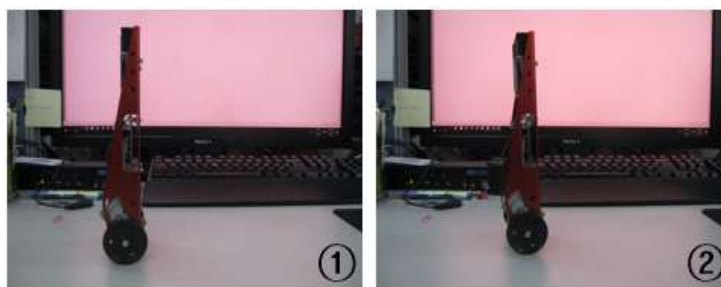
(b) Dynamic Behavior of the Robot when the Robot Moves Back

Figure 7. Robot Behaviors under an External Back Force

5. Conclusion

The development of a new sensor control command fusion for the IPMR was described. This study forms a basis for the control of one-wheel self-balancing systems, such as ball-balancing robots, in the future. A new method was applied for the dynamic tilt compensation algorithm of the IPMR, where only one tilt sensor and a gyro sensor were used to build the horizontal balancing system.

A real system that considers the motor and robot dynamics and a linear one based on the least-squares method were examined. The responses for both the real system and the linear model cannot be compared directly against the sensor errors. To this end, a fusion of three control commands — compensation and error, PID control, and balancing control — should be used to keep the robot balanced, because it is an open-loop, unstable system. The experimental results indicate that the θ_{SLANT} and θ responses for the real system are similar in shape than those presented by the models, but some model parameter adjustments should be made to obtain closer settling times for the models with respect to the real robot.

The experimental results show that the developed control strategy enables the IPMR to balance its body dynamically. The proposed economical horizontal-balancing technique can be applied not only for the IPMR's balancing, but also for personal riding robots, mobile service robots, and automobiles to control the horizontal balance inexpensively and efficiently.

Acknowledgments

References

[1] T. S. Jin, J. M. Lee and H. Hashimoto, "Position Estimation of Mobile Robot using Images of Moving Target in Intelligent Space with Distributed Sensors", Advanced Robotics, The Robotics Society of Japan, vol. 20, no. 6, (2006), pp. 737-762.

[2] H. Mehrjerdi, M. Saad and J. Ghommam, "Hierarchical Fuzzy Cooperative Control and Path Following for a Team of Mobile Robots", In: IEEE/ASME Transactions on Mechatronics, vol. 16, no. 5, (2011), pp. 907-917.

[3] J. H. Lee, H. J. Shin, S. J. Lee and S. Jung, "Balancing control of a single-wheel inverted pendulum system using air blowers: Evolution of Mechatronics capstone design", Mechatronics, vol. 23, (2013), pp. 926-932.

[4] C. Sundin and F. Thorstenson, "Autonomous balancing robot: Design and construction of a balancing robot", Master of Science Thesis. Chalmers University of Technology, Goteborg, (2012).

[5] J. J. Wang, "Simulation studies of inverted pendulum based on PID controllers", Simulation Modelling Practice and Theory, vol. 19, (2011), pp. 440-449.

[6] F. Grasser, A. D'Arrigo, S. Colombi and A. C. Rufer, "JOE: a mobile, inverted pendulum", IEEE Transactions on Industrial Electronics, vol. 49, (2002), pp. 107-114.

[7] W. An and Y. Li, "Simulation and Control of a Two-wheeled Self-balancing Robot", IEEE International Conference on Robotics and Biomimetics (ROBIO), China, (2013), pp. 456-461.

[8] T. S. Jin, "Command Fusion for Navigation of Mobile Robots in Dynamic Environments with Objects", Journal of information and communication convergence engineering, vol. 11, no. 1, (2013). pp. 24-29.

[9] F. Han, Y. Shan, R. Cekander, H. Sawhney and R. Kumar, "A two-stage approach to people and vehicle detection with hog-based SVM", In Performance Metrics for Intelligent Systems Workshop in conjunction with the IEEE Safety, Security, and Rescue Robotics Conference, (2006), pp. 133-140.

[10] J. H. Kim, "Fast Motion Estimation Based on a Modified Median Operation for Efficient Video Compression", Journal of information and communication convergence engineering, vol. 12, no. 1, (2014), pp. 53-59.

Author

Taeseok Jin, he received his Ph.D. degrees from Pusan National University, Busan, Korea, in 2003, in electronics engineering.

He is currently an associate professor at Dongseo University. From 2004 to 2006, he was a Postdoctoral Researcher at the Institute of Industrial Science, The University of Tokyo, Japan. His research interests include mobile robots, computer vision, and intelligent control. Dr. Jin is a Member of the KIIS, KIEE, ICROS, and JRS.