Model Based Design of Cyber Physical Systems

A Case Study: Two-Wheeled Self-Balancing Robot

Dilşad Engin
Ege University Ege Higher Vocational School
Department of Electronics and Automation
Bornova, izmir, Turkey

Mustafa Engin
Ege University Ege Higher Vocational School
Department of Electronics and Automation
Bornova, izmir, Turkey

***Abstract*—In this study, the model-based design steps in the design of cyber physical systems requiring multi-disciplinary study is shown on a two-wheeled self-balancing robot. A 3D model of the two-wheeled robot is designed and implemented on a 3D CAD software; consequently, the required components and mechanical parameters are determined. The obtained data is transferred to MATLAB™ and forward and inverse kinematic analysis is performed. The system is modelled in Simulink and the control algorithm is optimized. The 3D simulation is run by using the obtained data and the control algorithms.**

Keywords—cyber physical system; model based design; two-wheeled self-balancing robot; embedded system.

I. INTRODUCTION

A cyber-physical system (CPS) is a system that cooperating computational parts with physical components work together. In our daily lives, CPS comes up as smart buildings, smart networks, air traffic control, advanced production machines, defense systems, smart transportation, smart medical treatment and diagnostic devices, autonomous cars and robots [1], [2]. Regardless of the product produced, today, the companies had to launch their products in the shortest time and at the lowest cost if they wanted to be successful in the competitive market. The way to do this is to give the engineering organization an innovative structure that can follow technological developments [3].

Due to the customer requirements, tightening safety and environmental regulations, and market competition, the number of components in each product continues to rise. Making all the components work together becomes increasingly difficult, hindering the design and implementation of innovative features [3], [4].

The key challenges of innovation and complexity place pressure on engineering organizations from every perspective—not only technical, but also organizational, administrative, and cultural. Engineers must design systems comprising many parts so that all the parts work together. Often they must do so within shrinking development schedules, working with geographically scattered teams, and using development methodologies rooted in an Industrial Age culture, with its bureaucratic corporate structure and hard boundaries between departments [5].

Some organizations tackle system complexity by removing features or by simply accepting lower performance. In other words, requirements are changed to fit what has been made, and innovations are deferred or canceled [2], [6]. Other organizations tackle the problem by hiring more engineers. This approach can only work to a degree. As an organization grows, there arise new challenges such as communication and knowledge sharing between departments and groups [7].

The two-wheeled self-balancing robot attracted attention as a solution to this issue. Thus, it has been the subject of many studies. The main reason it perceives is that there is a balance problem. This balance problem is solved based on the pendulum model. Therefore, one should understand the mathematical model of the robot built upon the pendulum model in detail. The inverted pendulum problem has been a complex problem. Motion equations are non-linear. In obtaining these equations from the Newton motion law, Lagrange energy method or similar dynamic approaches must be utilized. In the study of Bugeja the system for the pendulum has become a reference feedback linearization technique, which also takes into account the energy of the pendulum, a hybrid approach by analyzing the situation feedback methods for balancing [8],[9].

This paper provides arguments to enable engineers to use model-based design within their organization. It provides a road map to the major concepts of Model-Based Design, and shows how these concepts used together or individually, can help to make any organization more efficient and better prepared to meet the challenges of change, complexity, and innovation.

II. MODEL BASED DESIGN

Model-Based Design is established on eight core concepts: executable specification, system-level simulation, what-if analysis, model elaboration, virtual prototyping, continuous test and verification, automation, knowledge capture and management [7], [10].

An executable specification is a model that encapsulates all design information, including requirements, system components, and intellectual property (IP), and test scenarios. It can be a model of the environment with uses cases that the embedded software needs to manage, or a high-level algorithm model that specifies the implementation’s exact behavior. In system-level simulation, a model of the entire system is simulated to investigate system performance and component interactions. It can use to validate requirements, check the feasibility of a project, and conduct early test and verification. What-if analysis is a simulation method used to test ideas and learn about the system. What-if analysis can be performed to test a single component or to investigate the interactions of all components in the system [2].

Model elaboration begins once you have simulated the high-level system model to verify requirements. When the model yields the desired results, details and refinements are added, and the model is simulated again. Common refinements include converting from floating point to fixed point, converting from continuous time to discrete time, replacing a behavioral actuator model with a detailed actuator model, and adding signals for diagnostics. Virtual prototyping is a technique that uses simulation to validate a design before hardware is available. In cases where the plant and environment are not yet fully known or understood, such as a mechanical construction, it may be necessary to use a hardware prototype for experiments to build the model. The knowledge acquired from these experiments is then stored in the model, where it can be transferred to other developers, departments, suppliers, and customers.

Virtual prototypes save development time because building a model is usually much faster than building a physical prototype. Virtual prototypes also reduce cost and increase innovation as they enable a team quickly and safely try out new concepts. In many situations, a model can replace a test rig. Using a model reduces development bottlenecks, since test rigs are often a scarce resource. Continuous test and verification is the practice of simulating a design at every stage of development. It is used to identify faults as soon as they are introduced into the design. Continuous test and verification can take different forms, and it can be conducted at different levels, depending on the complexity of the system and the stage of development.

Automation is the practice of using scripts and tools to perform repetitive tasks or tasks that are error-prone when performed manually. In model-based design, models are the primary source of project information. That knowledge includes not only design specifications and details about the system under development, but also product knowledge, team members’ design expertise, past experience, and design best practices. The models become a common language for the transfer of information within teams and with customers and suppliers. Because the models can be executed or simulated, the knowledge they contain increases as understanding of the system grows.

III. CASE STUDY TWO-WHELED SELF BALANCING ROBOT

Many robots use wheels to move because they are much simpler to develop than legs. They only require a motor to drive the wheels directly, and sometimes to steer the wheels, whereas legs often have complex mechanisms in order to make contact with the ground and provide a driving force.

Many robots will have three wheels for perfectly constrained static stability. The third wheel is often a holonomic wheel such as a castor wheel. This allows a robot to use differential control of two main driving wheels to track a reference trajectory. The advantage of differential drive is its simplicity. For higher speeds in motor vehicles, automobiles use 4 wheels because a support polygon of twice the area can be obtained, so that stability can be maintained in the presence of higher accelerations which result in higher moments making tip-over more likely. However, with more wheels, more complex mechanisms for driving are required, such as control of steering and, sometimes, individual control of wheel speeds.

A parallel two-wheeled robot has two co-axial wheels attached what is referred as an intermediate body.

This configuration of wheels is highly maneuverable because of its ability to turn on the spot by differential drive control of the two wheels.

A. *Mechanical Design*

In any cyber physical system (CPS) design, there is always room for mechanical improvements to make the design more efficient, especially in terms of strength-to-weight properties of any structural components. This requires both a thorough understanding of the applied loads and a thorough analysis of the stress developed under the loads [11].

Two-wheeled mobile robot chassis should be designed with minimal consideration for both of these requirements because the geometric interfaces are the critical design factor, making it the most likely candidate for design improvement and mass reduction. This highly integrated chassis design would likely improve the system performance, although at the expense of design time, machining time, and increased cost. Alternatively, carbon fiber or aluminum honeycomb panels could be used for some of the structural plates in the chassis design, which weigh less than solid aluminum plates. In addition, the bearings and support plates at the boom connection joint make up a substantial percentage of the chassis weight and are likely oversized. An analysis of the chassis loads can lead to reduction in the boom joint size and another considerable weight savings [12].

In this study, design process has been handled by SOLIDWORKS, a commercial 3D Cad (Computer Aided Design) software. The main purpose of the study was to construct a mathematical model of the manipulator and analyze it, so it was preferred to create a basic and simple CAD model instead of a complicated one.

The completed two-wheeled mobile robot 3D model side view shown in Fig.1, consisting all of the subsystems, with the addition of the sensors and control electronics. Two-wheeled mobile robot is a highly integrated system, and the assembly procedures are not quite as simple as joining all the subassemblies together. A complete documentation package ensures that any parts on two-wheeled mobile robot can be replaced and that the system can be disassembled and reassembled simply by referencing the documentation.

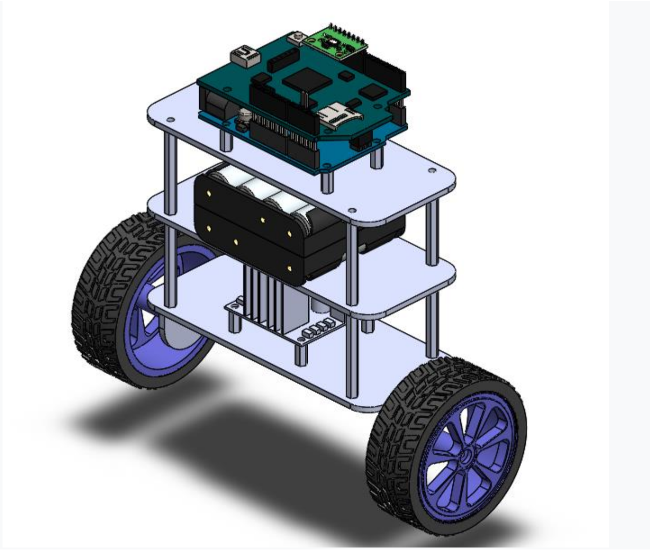

Fig. 1. *Generel view of two-wheeled mobile robot 3D Model.*

B. *Mathematical Model*

Before studying traction control of the two-wheeled robot, a suitable model for the robot should be determined for simulation purposes so that model-based controllers can be designed. There exist a number of models of differing fidelity, and accounting for various dynamics. Some group of researchers prefer using Newton's equations of motion [13][14][15][16]. The others prefer using Euler-Lagrange equations [17][18][19][20]. These two methods use differential and integral calculus. Kane proposes new method which is based on vector approach [21].

The two-wheeled balance robot is similar to the reverse pendulum in many aspects, thus it can be modeled based on the reverse pendulum problem, which is a fully solved system. The inverted pendulum, as seen in Figure 3 has two main parts, pendulum and car. The pendulum, which is free in the vertical axis, the place affecting its weight it cannot keep its balance with the effect of gravity. For this reason, between the pendulum and the vertical axis an angle occurs. The force, F, acting on the carriage carrying the pendulum, controls the angle θ. The dynamic correlation between the force and the angle, θ, can be obtained by various mathematical methods. The most common method they use is the Lagrange energy method. Thus, mathematical relations can define obtained system dynamics.

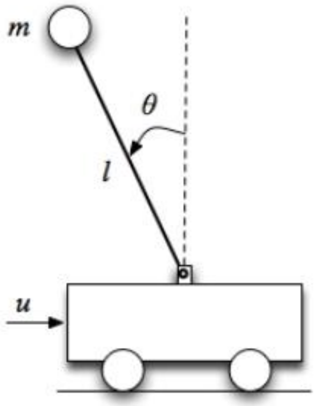

Fig. 2. *Inverted pendulum on a chart.*

The energy equation of the balancing robot according to the physical parameters is given in (1).

$$T = \frac{1}{2}mv^2 + \frac{1}{2}I\omega^2 \quad (1)$$

The position of the wheel's center of gravity, the relation of the pendulum's center of gravity, and the related general kinetic energy equations are given in (2)-(6) respectively.

$$x_2 = (k + lsin\theta)i + (2a + lcos\theta)j \quad (2)$$

$$v_1 = \dot{x}_1 = \dot{k}i \quad (3)$$

$$v_2 = \dot{x}_2 = \left(\dot{k} + l\dot{\theta}cos\theta\right)i + \left(-l\dot{\theta}sin\theta\right)j \quad (4)$$

$$T = \frac{1}{2}Mv_1{}^2 + \frac{1}{2}mv_2{}^2 + \frac{1}{2}I\dot{\theta}^2 \quad (5)$$

$$T = \frac{1}{2}M\dot{k}^2 + \frac{1}{2}m\left[\dot{k}^2 + 2\dot{k}\dot{\theta}lcos\theta + l^2\dot{\theta}^2\right] + \frac{1}{2}I\dot{\theta}^2 \quad (6)$$

We obtained the potential energy as in (7) and (8).

$$V = mgh \quad (7)$$

$$V = mgl(1 - cos\theta) \quad (8)$$

Lagrange equations were calculated as in (9)-(15).

$$L = T - V \quad (9)$$

$$L = \frac{1}{2}M\dot{k}^2 + \frac{1}{2}m\left[\dot{k}^2 + 2\dot{k}\dot{\theta}lcos\theta + l^2\dot{\theta}^2\right] + \frac{1}{2}I\dot{\theta}^2 - mgl(1 - cos\theta) \quad (10)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i \quad (11)$$

$$q_1 = k \;\; q_2 = \theta \quad (12)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{k}} = (M + m)\ddot{k} + mlcos\theta\ddot{\theta} - mlsin\theta\dot{\theta}^2 \quad (13)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} = (ml^2 + I)\ddot{\theta} + m\ddot{k}lcos\theta - ml\dot{k}sin\theta\dot{\theta} \quad (14)$$

$$\frac{\partial L}{\partial \theta} = m\dot{k}\dot{\theta}lsin\theta + mglsin\theta \quad (15)$$

After obtaining Lagrange expressions, dynamic parameters was established according to generalized forces given in (16)-(19).

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{k}} - \frac{\partial L}{\partial k} = u \quad (16)$$

$$u = F - b\dot{k} \quad (17)$$

$$(M + m)\ddot{k} + b\dot{k} + mlcos\theta\ddot{\theta} - mlsin\theta\dot{\theta}^2 = F \quad (18)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = 0 \qquad (19)$$

Since there is no external force on the body, the equation is equal to zero as given in (20).

$$(ml^2 + I)\ddot{\theta} + m\ddot{k}l\cos\theta - 2ml\dot{k}\sin\theta\dot{\theta} - mgl\sin\theta = 0 \qquad (20)$$

As a result, two parametric equations were found as in (21) and (22).

$$\ddot{k} = \frac{F - b\dot{k} - ml\cos\theta\ddot{\theta} + ml\sin\theta\dot{\theta}^2}{(M+m)} \qquad (21)$$

$$\ddot{\theta} = \frac{mgl\sin\theta + 2ml\dot{k}\sin\theta\dot{\theta} + ml\sin\theta\dot{\theta}^2}{(ml^2+I)} \qquad (22)$$

The above equations were embedded as Matlab functions in Simulink as shown in Fig. 3, and simulated for a certain force and the simulation result is given in Fig. 4 for robot angle response.

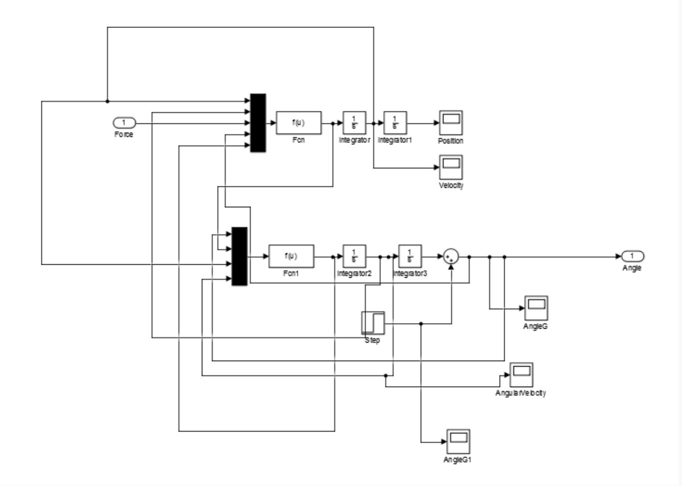

Fig. 3. *Nonlinear robot dynamics block diagram.*

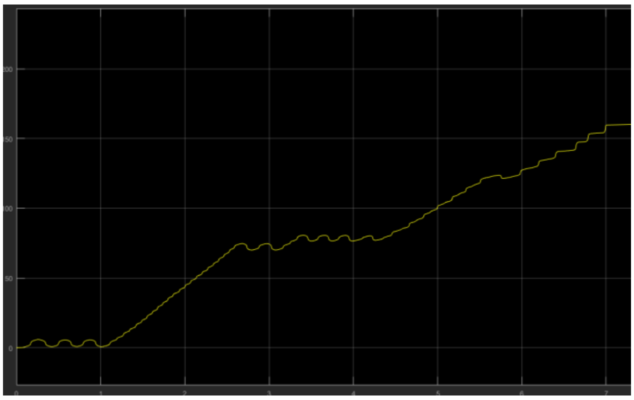

Fig. 4. *Robot angle response to 1N disturbance force.*

C. Controller Design

In this study, the control algorithm for balancing robot was implemented by fuzzy logic. The angle of the body to be controlled was measured with an accelerometer. For this, firstly, PID control was applied to the system and the behavior of the system was given (Fig. 5). After this stage, fuzzy logic was adopted.

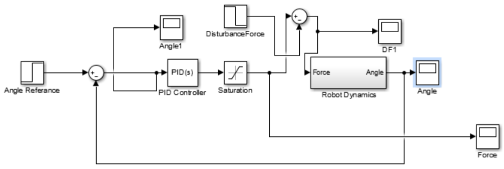

Fig. 5. *Self Balancing Robot Crisp PID Block Diagram*

Matlab Tuning found the gains of the system as above. After that, fuzzy logic P was tried as given in Fig. 6.

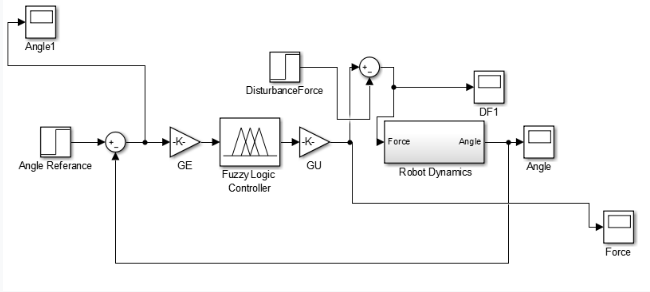

Fig. 6. *Self Balancing Robot Fuzzy P Block Diagram*

The input to the Robot Dynamics block can be calculated as

$$\big(f(e * GE)\big)GU = U \qquad (23)$$

By using a linear approach, we can simulate the fuzzy block to the normal system

$$f(e * GE) \cong e * GE \qquad (24)$$

Accordingly, we obtained the new equation as given in (25).

$$(e(n) * GE)GU = U \qquad (25)$$

We can find the control parameters by typing the classical P equation and simulating (25).

$$K_p(e(n)) = U \qquad (26)$$

$$K_p = GE * GU \qquad (27)$$

We can find fuzzy logic gains according to K_p values that we found in the real system. K_p values must be in the range of error given in (28).

$$e = [-\pi \quad +-\pi] \qquad (28)$$

Accordingly, the gain values found were as follows:

$$GU = 0.03 \qquad (29)$$

$$GE = -1000 \qquad (30)$$

Membership functions of the system are in Fig. 8 and Fig. 9.

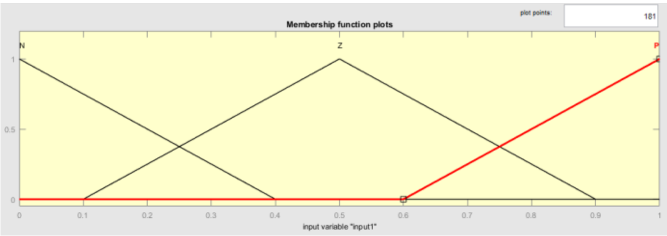

Fig. 7. *Error Membership Function.*

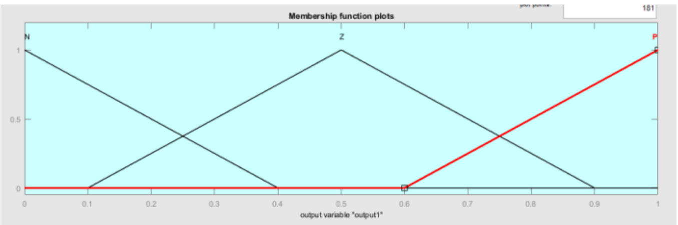

Fig. 8. *Output Membership Function*

Fig. 9. *Robot Angle Response to Disturbance Force*

IV. RESULT AND DISCUSSION

Two 12 V motors with 29:1 gear ratios were used to control the robot. Two 90 mm Pololu wheels were mounted directly to the gearboxes. The battery was attached to the top with Velcro. The Speed Controller, IMU, and microcontroller were mounted to the frame of the robot. The dimensions of the robot are 3.5x7.5x9 cm. The main supports are 3/8 inch threaded stock, and the platforms are ½ mm Plexiglas. A piece of heavy iron was attached to the top of the robot with heat-activated adhesive, where this weight at the top helps the robot stay balanced.

The two-wheeled mobile robot is implemented using a Cortex-M4 microcontroller on an STM32F401 Black pill development board [22]. The microcontroller interfaces directly with the Invensense MPU-6050 IMU via the I2C protocol [23]. The microcontroller also interfaces directly with the LM298N Motor Shield Electronic Speed Controller via Pulse Width Modulation (PWM) [24]. The Motor Shield controls the speed of the DC motors based on the PWM signal it receives from the microcontroller and the direction based on two lines the microcontroller drives. The Motor Shield connects directly to an 11.1V Lithium Polymer 3-cell battery. The motor shield has a 5V output port used to power the microcontroller.

The robot is able to stay upright and balanced for as long as the battery has power. It is also fairly good at recovering from disturbances. The performance and functionality of the design were adequate for the specified requirements.

In Fig. 10, the experimental operation of the designed robot is presented while carrying the full glass without spilling the beverage inside.

Fig. 10. *Experimental set-up of the robot carrying a full glass.*

V. CONCLUSION

This paper presented model-based design approach to develop an autonomous 3D two-wheeled robot. The proposed methods use the advantages offered by computer aided design software package, to determine critical parameters that included the dc motor, the gearbox, the motor drivers, control algorithms and the wheel size. The system parameter mapping process was categorized into five steps that detailed the relationships between the critical parameters and the design variables. Also as part of the mapping process, design constraints were established to limit the search size based on factors that included expected performance, size constraints, and safety concerns, among others. Each configuration's performance was evaluated and the simulations results were presented.

REFERENCES

[1] T. Habib and H. Komoto, “Comparative analysis of design concepts of mechatronics systems with a CAD tool for system architecting,” *Mechatronics*, vol. 24, no. 7, pp. 788–804, Oct. 2014.

[2] A. Ahmad, K. Andersson, and U. Sellgren, “A model-based and simulation-driven methodology for design of haptic devices,” *Mechatronics*, vol. 24, no. 7, pp. 805–818, Oct. 2014.

[3] C. Zheng, M. Bricogne, J. Le Duigou, P. Hehenberger, and B. Eynard, “Knowledge-based engineering for multidisciplinary systems: Integrated design based on interface model,” *Concurr. Eng. Res. Appl.*, vol. 26, no. 2, pp. 157–170, Oct. 2018.

[4] C. Zheng, J. Le Duigou, M. Bricogne, and B. Eynard, “Multidisciplinary interface model for design of mechatronic systems,” *Comput. Ind.*, vol. 76, pp. 24–37, Feb. 2016.

[5] P. Marwedel and M. Engel, “Cyber-physical systems: Opportunities, challenges and (Some) solutions,” in *Internet of Things*, vol. 0, no. 9783319268675, Springer, Cham, 2016, pp. 1–30.

[6] P. G. Larsen *et al.*, “Integrated tool chain for model-based design of Cyber-Physical Systems: The INTO-CPS project,” in *2016 2nd International Workshop on Modelling, Analysis, and Control of Complex CPS, CPS Data 2016*, 2016.

[7] J. Wan, A. Canedo, and M. A. Al Faruque, “Functional Model-Based Design Methodology for Automotive Cyber-Physical Systems,” *IEEE Syst. J.*, vol. 11, no. 4, pp. 2028–2039, Mar. 2015.

[8] M. Bugeja, “Non-linear swing-up and stabilizing control of an inverted pendulum system,” in *IEEE Region 8 EUROCON 2003: Computer as a Tool - Proceedings*, 2003, vol. B, pp. 437–441.

[9] S. Nundrakwang, T. Benjanarasuth, J. Ngamwiwit, and N. Komine, “Hybrid controller for swinging up inverted pendulum system,” in *2005 Fifth International Conference on Information, Communications and Signal Processing*, 2005, vol. 2005, pp. 488–492.

[10] M. S. Erden, H. Komoto, T. J. Van Beek, V. D’Amelio, E. Echavarria, and T. Tomiyama, “A review of function modeling: Approaches and applications,” *Artif. Intell. Eng. Des. Anal. Manuf. AIEDAM*, vol. 22, no. 2, pp. 147–169, 2008.

[11] D. Ding, Q. L. Han, Z. Wang, and X. Ge, “A Survey on Model-Based Distributed Control and Filtering for Industrial Cyber-Physical Systems,” *IEEE Trans. Ind. Informatics*, vol. 15, no. 5, pp. 2483–2499, May 2019.

[12] C. Bernardeschi, A. Domenici, and P. Masci, “A PVS-Simulink Integrated Environment for Model-Based Analysis of Cyber-Physical Systems,” *IEEE Trans. Softw. Eng.*, vol. 44, no. 6, pp. 512–533, Jun. 2018.

[13] R. Xiaogang and J. Cai, “Fuzzy backstepping controllers for two-wheeled self-balancing robot,” in *Proceedings - 2009 International Asia Conference on Informatics in Control, Automation, and Robotics, CAR 2009*, 2009, pp. 166–169.

[14] P. K. W. Abeygunawardhana, M. Defoort, and T. Murakami, “Self-sustaining control of two-wheel mobile manipulator using sliding mode control,” in *International Workshop on Advanced Motion Control, AMC*, 2010, pp. 792–797.

[15] L. Jingtao, G. Xueshan, H. Qiang, D. Qinjun, and D. Xingguang, “Mechanical design and dynamic modeling of a two-wheeled inverted pendulum mobile robot,” in *Proceedings of the IEEE International Conference on Automation and Logistics, ICAL 2007*, 2007, pp. 1614–1619.

[16] H. Fukushima, K. Muro, and F. Matsuno, “Sliding-Mode Control for Transformation to an Inverted Pendulum Mode of a Mobile Robot With Wheel-Arms,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4257–4266, Jul. 2015.

[17] J. Åkesson, A. Blomdell, and R. Braun, “Design and control of yaip - An inverted pendulum on two wheels robot,” in *Proceedings of the IEEE International Conference on Control Applications*, 2006, pp. 2178–2183.

[18] W. Wei, M. Xiaoning, and W. Jijun, “Intelligent control in two-wheel self-balanced robot,” in *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering, CMCE 2010*, 2010, vol. 3, pp. 470–473.

[19] J. H. Han, S. S. Zhao, J. S. Li, and H. Li, “Research on developed parallel two-wheeled robot and its control system,” in *Proceedings of the IEEE International Conference on Automation and Logistics, ICAL 2008*, 2008, pp. 2471–2475.

[20] K. Furuta, H. Kajiwara, and K. Kosuge, “Digital control of a double inverted pendulum on an inclined rail,” *Int. J. Control*, vol. 32, no. 5, pp. 907–924, 1980.

[21] T. R. Kane and D. A. Levinson, *Dynamics, theory and applications*. McGraw-Hill, 1985.

[22] ST, “STM32F413RH - High-performance access line, Arm Cortex-M4 core with DSP and FPU, 1,5 MByte of Flash memory, 100 MHz CPU, ART Accelerator, DFSDM - STMicroelectronics.” [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32f401cc.html. [Accessed: 23-Jul-2021].

[23] TDK InvenSense, “Mpu-6050 | Tdk,” 2020. [Online]. Available: https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/. [Accessed: 23-Jul-2021].

[24] STMicroelectronics, “L298 - Dual Full Bridge Driver - STMicroelectronics.” [Online]. Available: https://www.st.com/en/motor-drivers/l298.html. [Accessed: 23-Jul-2021].