Proceedings of 2021 International Conference on Advanced Robotics and Intelligent Systems
Taipei Nangang Exhibition Center, Taipei, Taiwan, 17-20 August, 2021

Motion Planning and Navigation Control of a Multifunctional Gyro-Stabilized Two-Wheeled Deformable Robot

Chieh-Wei Tsai, Feng-Chun Tai, and Ching-Chih Tsai
Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan, ROC

***Abstract* —This paper presents techniques for motion planning and navigation control of a multifunctional gyro-stabilized two-wheeled deformable robot (MGSTWDR). Such a robot is mainly with two modes, Segway and bike, which respond to different situations according to their motion characteristics. With capabilities of self-balancing and two-in-one control of the robot, new motion planning and automatic obstacle avoidance are designed for the two-mode operations. At the same time, the mode switching during the movement is also proposed to reduce the moving path of the robot by smoothening its motion trajectory. The simulation results show that the robot can automatically determine obstacles and modify its motion in real time, and choose a suitable mode to navigate from one place to another.**

***Keywords*:** *Gyro-stabilizer, self-balancing control, two-wheeled robot, two-in-one trajectory tracking, obstacle avoidance.*

I. INTRODUCTION

Recently, the electric two-wheeled robot car has been recognized as a powerful personal transportation tool. However, this kind of two-wheeled car or robot has a fatal flaw, namely that it could overturn or even overturn after colliding with a car. In order to overcome the shortcomings of overturning, in recent years, a variety of self-balancing platforms with gyro-stabilizers have been extensively studied to overcome their unstable characteristics. In addition, the self-balancing mobile platforms have gained applications in easily building many two-wheeled self-balancing robots, including Segway from Segway Inc. [1], self-balancing motorcycles from HONDA [2], Google Bike ideas [3], and gyro-stabilization LIT motor C-1[4].

Most of the presented two-wheeled self-balanced robots have been focused on the Segway robot. The study in [5] constructed the kinematic model and shows kinematic control method to achieve trajectory tracking with a Segway robot. Wheelchair is a kind of application of the Segway robot; the study in [6] constructed a wheelchair robot and derived its dynamic model, and even applied a neural network to control it. Moreover, the study in [7] established a steerable wheelchair with a special function of mode transition. The author in [8] designed a multifunctional gyro-stabilized two-wheeled robot (MGSTWR) which consists of two operational modes: Segway and bike; however, the robot needs to have more improvements in control and navigation. The results in [7-8] motivates us to improve the system structure, control and navigation of the multifunctional gyro-stabilized two-wheeled deformable robot (MGSTWDR) and make it more practical.

This paper is aimed to improve the MGSTWDR which is capable of self-balancing at any mode at anytime with a flywheel, and has the ability to move in all directions and navigate in its working environments. The presented contents put emphasis on motion planning, automatic obstacle avoidance and navigation of the designed robot. Meanwhile, the feasibility and effectiveness of the proposed methods will be confirmed through many simulations.

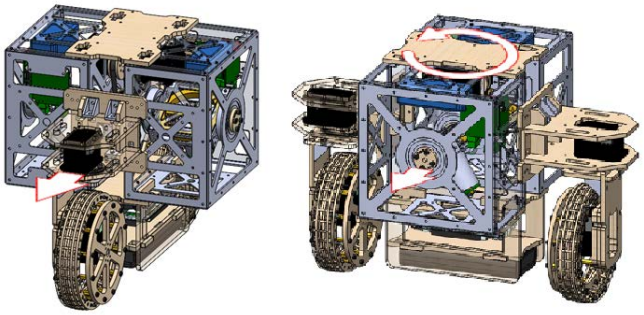
The rest of the paper is organized as follows. Section II briefly recalls the system structure, hardware and two-in-one controller of the robot. In Section III, the design method of smooth and practical motion paths are proposed. Section IV describes the steps and specific ideas of automatic obstacle avoidance and docking. In Section V, simulations are performed for motion planning, obstacle avoidance and navigation. Section VI concludes the paper and states future work.

II. SYSTEM STRUCTURE AND TWO-IN-ONE CONTROL

This section swill briefly describe the system architecture, two motion modes and two-in-one controller of the experimental MGSTWDR. Fig. 1 depicts the hardware structure of the MGSTWDR, where the front and rear wheels are the same and designed as steering wheels. These steering wheels can rotate from -90 degrees to +90 degrees, and vice versa. Unlike ordinary motorcycles or bicycles which are restricted by the front wheel steering, any direction of the robot can be achieved, and the robot always moves in the forward direction. Through the aforementioned design, the robot can move by using different configurations of both front and rear wheels. Through analytical analyses of motion modes of the robot, the robot can basically move by switching between Segway and bicycle modes. This section will focus on the description of the laboratory-built MGSTWDR in which both Segway and bicycle modes are briefly introduced, and then the two-in-one controller of the robot is revisited at last.

A. System Structure

As shown in Fig.1, the system structure of this robot is divided into two subsystems: flywheel balancing and wheel locomotion. In order to achieve corresponding functions, both subsystems are functionally deployed at the upper and lower parts of the robot. Such a flywheel balancing subsystem is composed of two flywheels, which can generate much greater torque than a single flywheel does, so that the body has a better balancing capability. In order to offset the unbalanced force generated when the two flywheels change, the rotation of the pair of scissors must be reversed and the angle change is the same. In doing so, two high-precision servo motors are the best and simplest choice for the subsystem such that the robot uses Dynamixel MX-64AR motor with the error being only 0.088 degrees.



(a) (b)

Fig. 1. Multifunctional gyro-stabilized two-wheeled deformable robot (MGSTWDR). (a) Bike mode. (b) Segway mode.

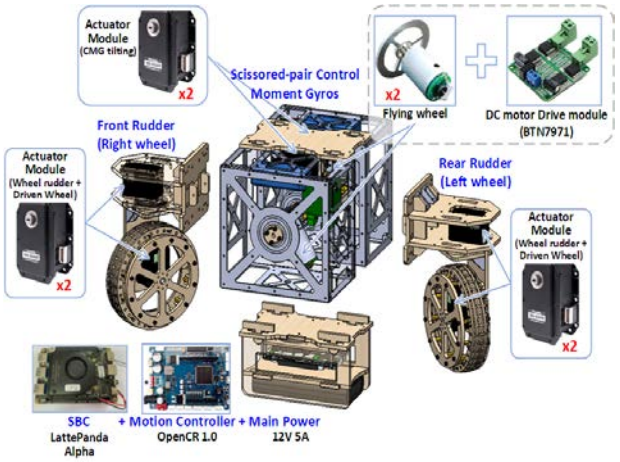
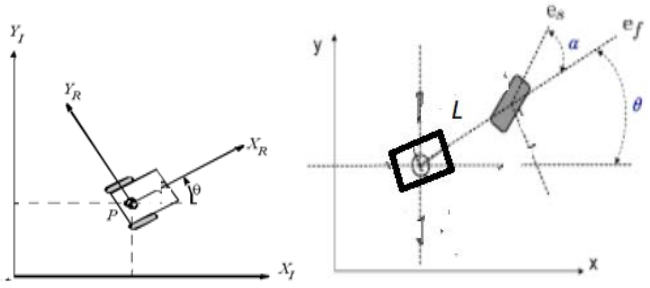


Fig. 2. Key components of the MGSTWDR.

On the other hand, the wheel motion subsystem also consists of two wheels which are front and rear wheels, respectively. Both front and rear wheels can serve as steering wheels, and they can be operated independently. With two independent steering wheels, the locomotion of the robot is omnidirectional, namely that the robot can reach at and position and attain any orientation by properly using both steering wheels. Hence, the coordination of the two wheels is very important. For example, if the forward moving directions of the two wheels are different or their speed differences are too large, then the body will overturn due to the force generated by the increasing wheel difference. Therefore, to ensure that the robot moves in the correct direction and speed, both front and rear steering wheels must be appropriately driven. Through the above-mentioned design, it is easier to let the flywheel balancing and wheel motion subsystems work together, thereby enabling the robot to move by switching between Segway and bicycle modes.

Fig. 2 displays the key components of the constructed MGSTWDR. For the used Dynamixel MX-64AR motors, newly implemented PID controllers can be used to maintain the shaft position by individually manipulating each servomotor, thereby controlling the tracking speeds and directions of the motors. All MX series servos use nominal 12VDC, so other MX-R series Dynamixel motors can be easily used in combination without worrying about a separate power supply. All sensor management and position control are handled by the servo's built-in microcontroller. This distributed method allows the main controller to perform other functions freely. Finally, the system architecture of the entire control system is shown in Figure 2, where the Lattepanda [9] controller commands the OpenCR, and then the OpenCR controls the Dynamixel motors to actuate the flywheel balancing and locomotion subsystems. .



(a) **(b)**

Fig. 3. Schematic diagrams of kinematic models of the robot. (a) Segway-mode. (b) Bike-mode.

B. Two Motion Modes

This subsection is devoted to recalling the two basic motion modes and two-in-one motion control law of the robot. As mentioned previously, the omnidirectional locomotion of the robot can be done by using different configurations of both steering wheels. It was reported in [8] that four motion modes and corresponding kinematic models of the robot were rigorously derived. In this section, the four motion modes can be further reduced to two fundamental modes: Segway and bike, as depicted in Fig. 3. To recall the two kinematic models of the two motion modes, we assume that the inclination angles of both motion modes can be maintained at zero by using appropriate self-balancing control of the gyro-stabilized. On basis of this assumption, the kinematics model of the Segway mode of the robot with a gyro stabilizer is given by a unicycle model [8]:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}\frac{(V_F+V_R)}{2}\cos\theta\\ \frac{(V_F+V_R)}{2}\sin\theta\\ \frac{(V_F-V_R)}{d_w}\end{bmatrix}=\begin{bmatrix}v\cos\theta\\ v\sin\theta\\ \omega\end{bmatrix}=\begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} \quad (1)$$

where $v=(V_F+V_R)/2=R\cdot(\omega_F+\omega_R)/2$, $\omega=(V_F-V_R)/d_w=R\cdot(\omega_F+\omega_R)/d_w$ respectively denote the linear and angular velocities of the robot. Moreover, V_F and V_R are respectively the velocities the front and rear wheels, R denotes the radius of the front and rear wheel, and d_w denotes the distance of both wheels.

Similarly, if the inclination angle of the robot is assumed to be keep at its upright angle, while working as a bicycle, then the kinematic behavior of the robot in the bike mode is modeled by following matrix equation.

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}v\cos\theta\\ v\sin\theta\\ 2v\tan\alpha/L\end{bmatrix}=\begin{bmatrix}v\cos\theta\\ v\sin\theta\\ \omega\end{bmatrix}=\begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} \quad (2)$$

where v and ω respectively denote the linear and angular velocities of the robot, L is the wheel base and α is the

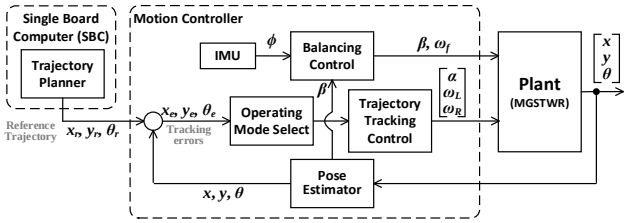


Fig. 4. Block diagram of the proposed MGSTWR.

steering angle of the robot serving as a bike. In (2), if $2v\tan\alpha / L$ is replaced by ω, then the kinematic models of both modes can be made identical mathematically, namely that it can be seen from (1) and (3) that the two modes of the robot have the same kinematics models.

C. Two-in-One Control

Since the kinematic models of the two motion modes of the robots can be made identical mathematically, it is easy to obtain trajectory tracking or stabilization control law for the robot since the control problem of a unicycle model has been extensively investigated by many researchers. In what follows, well-known nonlinear control laws of v and ω are given as follows [8];

$$\begin{aligned} v &= k_1 e_1 + v_r \cos e_3 \ , \quad k_1 > 0 \\ \omega &= \omega_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \ , \quad k_2 > 0,\ k_3 > 0. \end{aligned} \tag{3}$$

This control law can be applied to steer the Segway-model control in a straightforward manner. For the bike mode, the control law is modified from (2-3) and given by

$$\begin{aligned} v &= k_1 e_1 + v_r \cos e_3 \ , \quad k_1 > 0 \\ \alpha &= \tan^{-1}\left(\frac{d\left(\omega_r + k_2 v_r e_2 + k_3 v_r \sin e_3\right)}{v} \right) \ , \quad k_2 > 0,\ k_3 > 0. \end{aligned} \tag{4}$$

Fig. 4 shows the block diagram of the proposed two-in-one trajectory tracking controller for the MGSTWR. Once the motion mode has been chosen, the two-in-one controller will be activated so as to send out the control signals to the steering wheels, and, at the same time, the self-balancing controller receives the inclination information from the IMU mounted on the robot and sends out the control signals to make the robot maintain at its upright angle.

D. Overall Navigation Control

Fig. 5 depicts the overall flowchart of the proposed navigation controller where a 360-degree laser scanner is used to localize the robot moving around its working space, and also achieve obstacle avoidance. After the parameter settings, initial SLAM, the system proceeds with task planning, mode selection and path planning, and then carries out motion control until the goal or destination is arrived.

III. TASK PLANNING, MODE SELECTION AND PATH PLANNING

This section is dedicated to the proposal of techniques for task planning, mode selection and path planning. The task planner is aimed at finding a sequence of execution items and a good operational mode and a set of via points. The goal of mode selection is to select a better operation mode

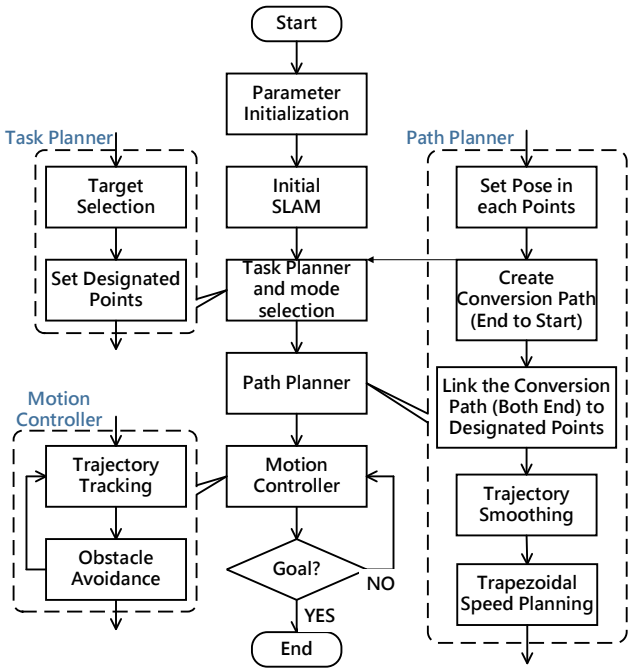


Fig. 5. Overall flowchart of the proposed navigation controller.

for the robot travelling around its working environments since the robot can be operated in the two basic modes. The path planner is mainly classified into four parts: conversion paths, path smoothing and trapezoidal speed planning. Below are the detailed description of these techniques in some detail.

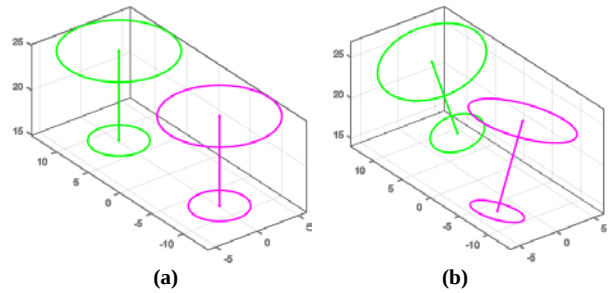
A. Task Planning

During navigation control, task planning is the basic requirement for the robot to navigate from one place to another, in order to achieve its desired mission. For example, the robot can be commanded to perform object searching in an indoor environment, and carry one light object to a target location. Task planning can be done by many approaches; some of them are optimized path planning on basis of the shortest path or minimal time or minimal energy. Next, let us discuss about them separately. In this section, the shortest path planning method is used to find ta global path via the well-known A* approach.

B. Mode Selection

Mode selection is essential to speed up moving or turning speeds for task execution in a more efficient way. Generally speaking, the bike mode is able to move the robot faster than the Segway mode does; this is because of two steering wheels in the bike mode. The Segway mode has a zero radius of turning due to differential driving, but has lower speeds than the bike mode does. On the other hand, the bike mode enables the robot move in a narrower corrido since the front and rear wheels are arranged in a straight line by comparing to the Segway mode.

Hence, mode selection heavily depends upon its travelling environment. If the working space is wide and open, then the bike mode is chosen due to its fast moving, then the bike mode will be suggested. If the execution task could be done in a tortuous path, then the Segway mode will be adopted owing to its smaller turning radius. If the desired



(a) (b)

Fig. 6. Structural Poses of the control moment gyroscope (CMG) scissored-pair. (a) Segway mode. (b) Bike mode. .

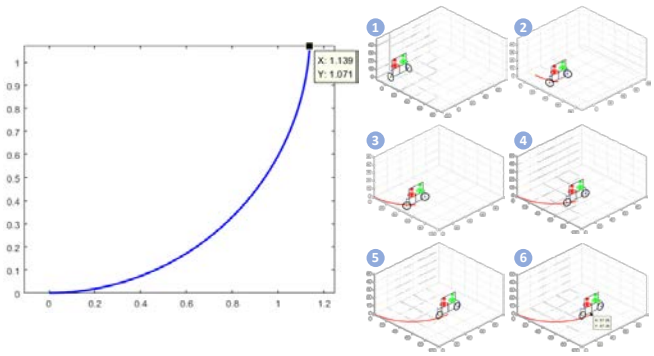


Fig. 7 Mode-transition trajectory from the Segway mode to the bike mode.

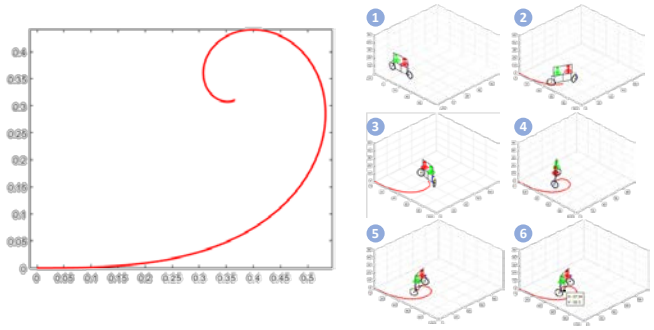
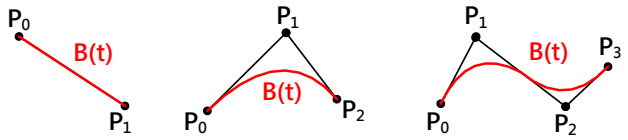


Fig. 8. Mode-transition trajectory from bike mode to Segway mode.

task could be done in a complicated environment, the mode selection will be activated since one of the operational modes is decided according to the wideness of its current working space, where the wideness can be measured by using the laser scanner.

Mode selection can be achieved in a static or dynamic transition. In the static mode selection, the robot has to be stopped before the mode transition, and the gyro stabilizer must be adjusted so that the robot is stabilized at any time. As shown in Fig. 6 (a), the structural pose of the control moment gyroscope (CMG) scissor-pair is suggested by make both gyroscope vertical, in order to generate enough precession force to enable the robot in the Segway mode to keep balancing. For the bike mode, the structural pose of the (CMG) scissor-pair is maintained at a tilting status, even at the inclination of 90 degrees, as shown in Fig. 6(b). Once the robot has been balanced for both operational modes, the steering angles of both steering wheels will easily controlled to reach the desired ones.

Dynamic mode transition is aimed to save the transition time during the mode transformation where the PD controller is applied to achieve self-balancing [8]. To simplify the dynamic mode transitions, two restrictions are



(a) (b) (c)

Fig. 9. Bezier curve: (a) Linear Bezier curve; (b) Quadratic Bezier curve; (c) Cubic Bezier curve.

considered. Steering angles of both wheels are set identical and both wheels must have the same speeds. Under these two restrictions, there are only two mode transition ways: both wheel steers are in the parallel phase or in anti-phase.

Once the two motion modes haven been activated, it suffices to construct two mode-transition trajectories switching between the Segway and bike modes. A mode-transition trajectory from the Segway mode to the bike mode is shown in Fig. 7, and is described as follows;

$$\begin{aligned} x(t+1) &= x(t) + v\cos(\alpha(t))t \\ y(t+1) &= y(t) + v\sin(\alpha(t))t \\ \alpha(t+1) &= \alpha(t) + kt \end{aligned} \tag{5}$$

where k is the slope of the increasing steering angle.

Similarly, a mode-transition trajectory from the bike mode to the Segway mode is depicted in Fig. 8, and is described as follows:

$$\begin{aligned} x(t+1) &= x(t) + 2v\cos(\alpha(t))\cos(\theta(t))t \\ y(t+1) &= y(t) + 2v\cos(\alpha(t))\cos(\theta(t))t \\ \alpha(t+1) &= \alpha(t) + kt \\ \theta(t+1) &= \theta(t) + 4vt\sin(\alpha(t))/L \end{aligned} \tag{6}$$

where θ is the heading angle or orientation of the MGSTWR, and L is the length between the wheels.

It is obvious that the starting and ending poses of the robot are different after the mode transition. Hence, the differences can be regarded as initial errors when the two-in-one trajectory tracking controller is turned on.

C. Trajectory Smoothing

For trajectory smoothing, a Bezier curve is used here. The Bezier curve is originally an algorithm based on car design, Of course, it can also be applied to our MGSTWDR. First, Below is the introduction to the three kinds of the Bezier curves, as shown in Fig. 9.

Linear Bezier curve: given points P_0 and P_1, the linear Bezier curve is just a straight line between two points. This line is given by:

$$B(t)=P_0+(P_1-P_0)t=(1-t)P_0+tP_1, t\in[0,1] \tag{7}$$

Quadratic Bezier curve: given points P_0, P_1, and P_2, the quadratic Bezier curve is traced by the following function B(t)

$$B(t)=(1-t)^2P_0+2t(1-t)P_1+t^2P_2, t\in[0,1] \tag{8}$$

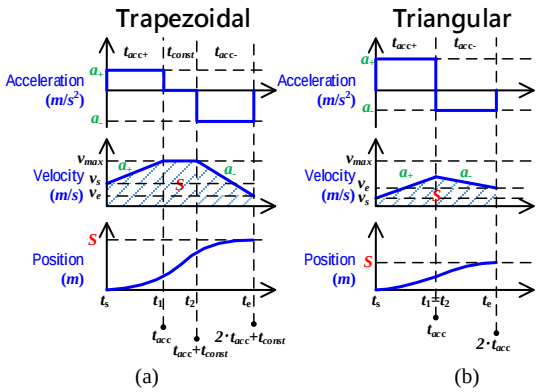


Fig. 10. Trapezoidal velocity planning (TVP): (a) Trapezoidal planning that reach the maximum velocity; (b) Triangular planning that will not reach the maximum velocity.

Cubic Bezier curve: this curve is implemented by choosing four reference points P_0, P_1, P_2, and P_3. The curve starts from P_0 to P_1, and goes from P_2 to P_3. Generally, it will not pass through P_1 or P_2; these two points only provide direction information. The distance between P_0 and P_1 determines the length of the curve in the direction of P_1 before turning to P_2.The parameter form of the curve is given by

$$B(t)=P_0(1-t)^3+3P_1t(1-t)^2+3P_2t^2(1-t)+P_3t^3,\ t\in[0,1] \quad (9)$$

To smoothen a complicated curve, we propose an *n*-order Bezier curve method in the following formula:

$$\begin{aligned} B(t)&=\sum_{i=0}^{n}\binom{n}{i}P_i(1-t)^{n-i}t^i \quad , \quad t\in[0,1] \\ &=\binom{n}{0}P_0(1-t)^n t^0+\binom{n}{0}P_1(1-t)^{n-1}t^1+\cdots \\ &+\binom{n}{n-1}P_{n-1}(1-t)^1 t^{n-1}+\binom{n}{n}P_n(1-t)^0 t^n \end{aligned} \quad (10)$$

Having explained the mathematical expression formulas of Bezier curves, the following delineates how to use it for trajectory planning. Since several via points have been obtained from the task planner, it is easy to apply the n-order Bezier curve method is used to connect all the via points and make the curves smooth.

D. Trapezoidal speed planning

Trapezoidal velocity planning (TVP) method [10] is the simplest way of speed planning used by the motion control system. Depends on whether the top speed v_t reach the maximum speed v_{max} or not, it can be separated to trapezoidal planning or triangular planning, the planning velocity curves are shown in Fig.10.

The TVP method create the speed curve by separating the curve with three different zones: acceleration, constant speed, and deceleration, where the accelerations a_1 and a_2 are constant, i.e., $v_{acc}=a_1 \cdot t$ and $v_{dec}=a_2 \cdot t$. Each accelerating zone can be treated as a right triangle area (startup or end at $v=0$) or trapezoidal area (non-zero startup or end speed), and constant speed zone as rectangle. Sum of the area will equals the moving distance. By setting the relationship between the initial speed v_s, maximum speed v_{max}, and the end speed v_e, with distance S and time t for this trajectory section, one can

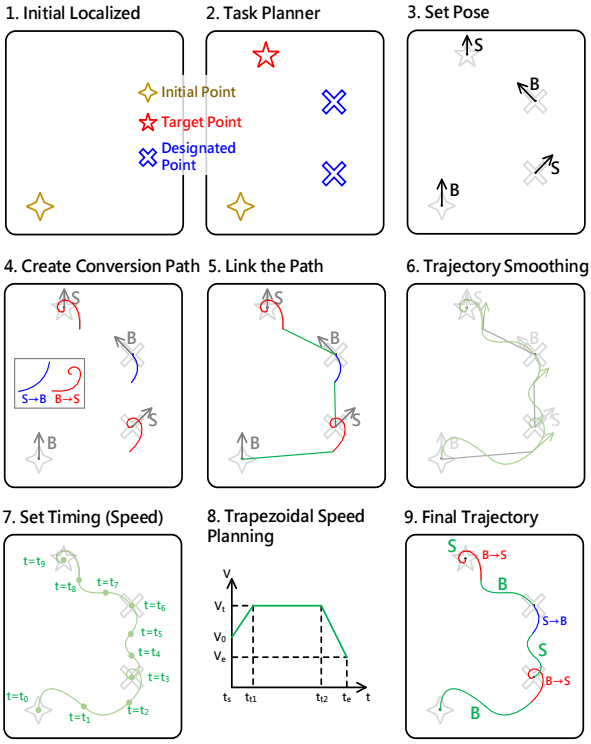


Fig. 11. Resultant task and path planning.

define the top speed and the velocity curve (trajectory) by

$$\begin{aligned} S&=S_{acc}+S_{const}+S_{dec} \\ &=\frac{1}{2}(v_s+v_{\max})t_1+v_{\max}(t_2-t_1)+\frac{1}{2}(v_{\max}+v_e)(t_e-t_2) \\ &=\frac{1}{2}v_st_1+\frac{1}{2}v_e(t_e-t_2)+\frac{1}{2}v_{\max}(t_1+(t_2-t_1)+(t_e-t_2)) \\ &=\frac{1}{2}v_st_1-\frac{1}{2}v_et_2+\frac{1}{2}(v_e+v_{\max})t_e \\ &=\frac{1}{2}(a_1t_1)t_1-\frac{1}{2}a_2(t_e-t_2)t_2+\frac{1}{2}(a_2(t_e-t_2)+v_{\max})t_e \\ &=\frac{1}{2}a_1\cdot t_1^2+\frac{1}{2}a_2\cdot t_2^2+\frac{1}{2}a_2\cdot t_e^2-a_2\cdot t_2\cdot t_e+\frac{1}{2}v_{\max}\cdot t_e \end{aligned} \quad (11)$$

where $v_{max}=v_s+a_1 \cdot t_1=v_e+a_2 \cdot (t_e-t_2)$ and v_t stands for the top speed. Namely, we can have the overall velocity planning as below.

$$v(t)=\begin{cases} a_1\cdot t & \text{if } t_s\le t<t_1 \\ v_{\max}=v_s+a_1\cdot t_1 & \text{if } t_1\le t<t_2 \\ \quad =v_e+a_2\cdot(t_e-t_2) & \text{and } v_t=v_{\max} \\ a_2\cdot t & \text{if } t_2\le t\le t_e \end{cases} \quad (12)$$

Simulrary, the angular velocity planning can be found as

$$\omega(t)=\begin{cases} \alpha_1\cdot t & \text{if } t_s\le t<t_1 \\ \omega_{\max}=\omega_s+\alpha_1\cdot t_1 & \text{if } t_1\le t<t_2 \\ \quad =\omega_e+\alpha_2\cdot(t_e-t_2) & \text{and } \omega_t=\omega_{\max} \\ \alpha_2\cdot t & \text{if } t_2\le t\le t_e \end{cases} \quad (13)$$

where α_1 and α_2 are respectively the angular acceleration and angular deceleration, and ω_t stands for the top angular speed in the speed planning.

In conclusion, Fig. 11 shows the resultant task and path planning in some detail for the robot.

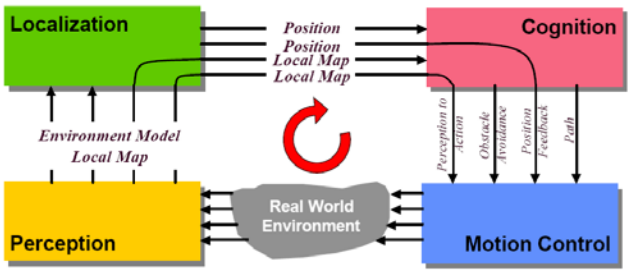


Fig. 12. Block diagram of a general navigation control system.

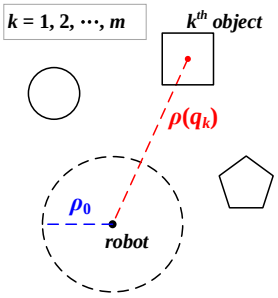


Fig. 13. Illustration of the collision and obstacle avoidance strategies for the robot.

IV. Design of Navigation Controller

A. Navigation Control

This section will describe the navigation control structure of the MGSTWDR and then propose an obstacle avoidance method using artificial potential functions. Fig. 12 depicts the block diagram of a general navigation control system, where the four main modules are perception, localization, cognition, and motion control. In the perception module, the IMU units are used to fulfill self-balancing control of the robot via the PD controller and gyro stabilizer, whereas the 360-degree laser scanner used in the localization module is applied to localize the robot and detect the static or moving obstacle near the robot's surrounding. The cognition module recognizes the objects and scenes of the robot's surroundings, and generates local paths to avoid the detected obstacles. The two-in-one control laws in (3) in the motion control module is employed to carry out the trajectory tracking missions for both modes of the robot.

B. Obstacle Avoidance

To avoid any collisions between robots and between any robot and any static or moving obstacle, it is necessary to add the collision-free or obstacle avoidance ability into the main function of the formation controller. In [10], the collision-free and obstacle avoidance algorithms were separately discussed. Unlike the ideas in [10], this section will propose a unified approach to achieving both collision and obstacle avoidance functions for the multi-ball-riding robot system.

This subsection is aimed to develop the unified collision-free and obstacle-avoidance method between the robot and any obstacle by using the potential field function and treating the robot as a point under the influence of the potential field function. From the viewpoint of the potential field, the objective of the unified obstacle-avoidance method is to generate a repulsive force to push the robot away from any detected obstacle. Thus, the overall repulsive potential U for the aims of collision and obstacle avoidance is defined as follows;

$$U(q) = U_{oa}(q) \tag{14}$$

where the repulsive potential of obstacle avoidance is defined by

$$U_{oa}(q) = \begin{cases} \frac{1}{2} K_{oa} \sum_{k=1}^{M} \left(\frac{1}{\rho(q_k)} - \frac{1}{\rho_0}\right)^2 & \text{if } \rho(q_k) \le \rho_0 \\ 0 & \text{if } \rho(q_k) > \rho_0 \end{cases} \tag{15}$$

In (15), the pair $q=(x, y)$ means the position of the robot; K_{oa} is scaling factor; $\rho(q_k)$ denote the distance from the robot to the k^{th} obstacle, namely that $\rho(q_k)=\|q-q^k_{obstacle}\|_2$, where $q^k_{obstacle}$ is the coordinates of the k^{th} obstacle. Moreover, ρ_0 is the radius of the safe region, and M is the number of the obstacles as shown in Fig. 13. The repulsive potential $U(q)$ is positive or zero when q gets closer to the k^{th} obstacle.

From (16), we have the repulsive force to avoid the obstacle as below.

$$\mathbf{u}_{oa_i} = -\nabla U_{oa}(q_i) = \left[-\frac{\partial U(q_i)}{\partial q_i}^T, 0 \right]^T \in R^3$$

$$= \begin{cases} K_{oa} \sum_{k=1}^{M} \left(\frac{1}{\rho(q_{ik})} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(q_{ik})} \frac{q_i - q^k_{obstacle}}{\rho(q_{ik})} & \text{if } \rho(q_{ik}) \le \rho_0 \\ 0 & \text{if } \rho(q_{ik}) > \rho_0 \end{cases} \tag{16}$$

Hence, the moving trajectory in the environments is given by

$$P_{doa} = P_d + u_{oa_i} \tag{17}$$

where P_{doa} is the current trajectory of the robot. Obviously, the robot will return to the originally planned trajectory after avoiding the obstacles.

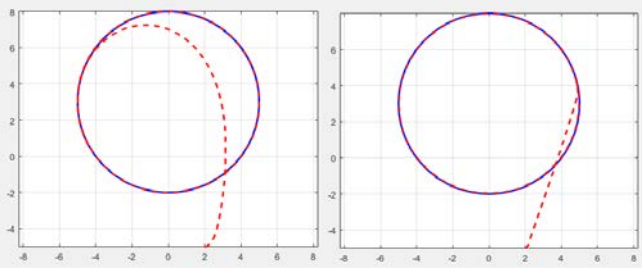
C. Docking

Docking aims to park the robot at a desired location with a wanted orientation once the navigation mission has been completed. In the subsection, docking can be done in either mode of the robot. Depending on the desired posture, the robot is easily controlled to stop at the desired destination position and heading by planning and following the trajectory to move toward the goal.

V. Simulation Results and Discussion

This section will verify the feasibility of the proposed methods by conducting two simulations based on the aforementioned mathematical model and control method. These two simulations include path planning, navigation control and obstacle avoidance.

A. *Path Planning and Navigation Control*



(a) **(b)**

Fig. 14. Planned path and navigation control . (a) Simulation results of the Segway mode. (b) Simulation results of the bike mode.

This first simulation program runs 1000 iterations, and the sampling time for one iteration is 0.01 seconds. Fig. 14 shows the simulation results of the circular trajectory planned for the Segway mode and bicycle mode. As can be seen from Fig. 14, the blue line represents the planned trajectory, and the red line represents the actual trajectory done by either the Segway mode or bike mode. Obviously, the planned trajectory is well followed by using the proposed navigation control method.

B. *Navigation Control and Obstacle Avoidance*

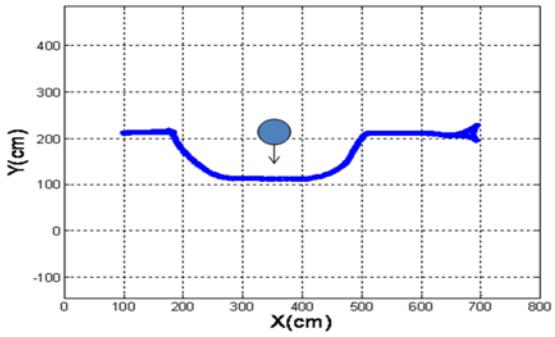


Fig. 15. Illustration of the obstacle avoidance method for the robot.

Fig. 15 shows the simulation result of the obstacle avoidance method for the robot originally moving along a straight line. The result in Fig. 15 indicates that the proposed obstacle avoidance method worked as predicted.

VI. Conclusions and Future Work

This paper has presented methodologies and techniques for motion planning and navigation control of a multifunctional gyro-stabilized two-wheeled deformable robot (MGSTWDR). This robot is basically operated in two modes: Segway and bike, which enable the robot to have different motions based on their motion commands. With capabilities of self-balancing and two-in-one control of the robot, new task and path planning method along with mode selection are designed for the robot. Meanwhile, the mode switching during the movement has also been considered to reduce the moving path of the robot by smoothening its motion trajectory. The navigation control together with obstacle avoidance has been proposed. The simulation results have shown that the robot is capable of proceeding with task and path planning, detecting obstacles and moving along its motion path in real time, and choose a suitable mode to effective navigate from one place to another. Future work will be conducting experiments to show practicability of the proposed method.

Acknowledgements

The authors deeply acknowledge financial support from the MOST, Taiwan, ROC, under contract MOST 109-2221-E-005 -066-MY2.

References

[1] Personal Transportation That Simply Moves You | Segway. Retrieved *Jul 19, 2020 from http://www.segway.com*
[2] Honda riding assist. Retrieved Jul 19, 2020 from *https://global.honda/innovation/CES/2017/002.html*
[3] Google idea self-driving bike, Retrieved Jul 19, 2020 from *https://www.youtube.com/watch?v=LSZPNwZex9s*
[4] Lit Motors self-balanced motor bike, Retrieved Jul 19, 2020 from *https:// www.litmotors.com/product.*
[5] C. C. Tsai and C, H. Tsai, "Direct adaptive fuzzy-wavelet-neural-network control foreElectric two-wheeled robotic vehicles," in *Proc. of 2013 IEEE International Conference on Systems, Man and Cybernatics*, Manchester, UK, October 13-16. 2013
[6] C. C. Tsai, Y. P. Ciou, F. C. Tai, "Indirect adaptive nonlinear self-balancing and station keeping for omnidirectional riding chair," in *Proc. of 2014 IEEE International Conference on Control, Automation and Systems*, KITEX, Gyeonggi-do, Korea, October 22-25. 2014.
[7] S. Hamatani and T. Murakami, "A novel steering mechanism of two-wheeled wheel chair for stability improvement," *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society, Yokohama, 2015, pp. 002154-002159, doi: 10.1109/IECON.2015.7392420.*
[8] C. H. Yang, *System Design, motion control and dynamic mode transition of a novel multifunctional gyro-stabilized two-wheeled robot*, Master Thesis, Department of Electrical Engineering, National Chung Hsing University, July 2020.
[9] DFR0546 LattePanda Alpha 864 – DFRobot. *Retrieved June 20, 2020 from* https://www.mouser.tw/new/dfrobot/dfrobot-dfr0546-lattepanda-alpha/
[10] M. Haddad, W. Khalil, and H. E. Lehtihet, "Trajectory planning of unicycle mobile Robots with a trapezoidal-velocity constraint" *in Proc. IEEE Transactions on Robotics*, vol. 26, no. 5, October 2010.