Husam A. NEAMAH[1,2,3*],
Rawitch BUTDEE[1.]

B-SPLINED TRAJECTORY MODIFIED GENERATION TO MAXIMIZE SPEED OF THE NONHOLONOMIC AMR ROBOT

Trajectory path generation is critical for the Autonomous Mobile Robot (AMR) when moving frequently in the working environment in the shop floor to transport loads from one work station to another continuously. Traditionally, the AMR moves from one point to stop at the next point or turn which is inefficient and consumes much energy. This paper proposes the new concept of AMR trajectory path planning with curvature driven by maximizing speed control of the differential drive at each curve to move smoothly. B-splined is commonly applied to CAD and CAM machining effectively for tool path trajectory. Therefore, the B-splined curvature is studied and validated by simulation together with energy consumption. The simulation is on Matlab Simulink with numerical model. It is investigated that the B-splined trajectory is efficient in animating the AMR's actual system. The velocity can obtain both linear and angular velocity of the AMR movements on forward and backward directions as well as the acceleration. The trajectory can be selected based on the degree of closeness and used to generate speed and velocity control for the AMR system.

1. INTRODUCTION

Mobile robot trajectory planning involves using an intelligent system to move efficiently in dynamic motion when changing path occurs or to avoid a collision of obstacles. Trajectory planning follows the path planning from the starting point to the end under the constraints of the geometric waypoints involving with velocity, acceleration and torque. Trajectory planning deals with smooth path motion with maximum or optimal velocity. This research aims to improve the efficiency of the AMR trajectory planning with smooth motions. In addition, the AMR robot must be able to generate a trajectory to avoid obstacles with less time and energy [1]. The velocity must be reduced when the AMR moves on the curve with velocity control of the left wheel and right wheel under the differential drive control. It moves fast on the straight path and slow-down on the curve. This also help with Organic Human-Robot Interaction (O-HRI) [2]. Traditionally, the AMR trajectory planning follows the waypoint

_____

[1] Department of Electrical Engineering and Mechatronics, Faculty of Engineering, University of Debrecen, Hungary
[2] Technical Engineering College, Al-Ayen University, Thi-Qar 64001, Iraq
[3] Department of Business Management, Al-imam University College, Balad, Iraq
[*] E-mail: husam@eng.unideb.hu

path planning which requires reducing velocity and decreasing acceleration, even to the point of stopping before starting to move again and increasing velocity [3]. Movement on the curve geometry is extremely smooth and more efficient than on the point-to-point of the linear line. Path planning with SLAM is mostly based on the point to point motion. It causes inefficient trajectory and low performance on the AMR. This paper proposes on the B-spline trajectory modified generation instead of using traditional the path waypoint planning to maximize speed of the nonholonomic AMR robot. B-splined was introduced to build up the joint trajectories for the robot motion and mission based on a sequence of Cartesian knots of positions and orientation for a six-joint arm robot [4–7]. B-spline trajectory planning was proposed for a mobile robot with curvature constraints in a two-dimensional plane [8]. They can be defined the problem as a nonlinear semi-infinite optimization.

The constraints included velocity and acceleration. Fast traversal of a path for the four wheel vehicles was previously interest of using automatic generation of a smooth planar for efficient guidance using B-spline curves [9]. The efficiency increases average 32.13%. Vertex movement and rapid velocity control in aerial robots was presented to achieve stable and continuous movements [10]. It can be the curves of B-spline and Bezier. Curved surfaces of the B-spline-based framework enabled the working of on the surface normal and position data [11]. Smoothing trajectories using NURBS was introduced for the jerk minimization and gradient-based optimization [7]. There are different kinds of smoothing trajectory for the various robot types. The mobile robot needs specific and proper trajectory planning to maneuver accurately at low speeds by deriving control parameters. Position and time are critical aspects to be taken into account when a perturbed wheeled mobile robot is designed the robust position-tracking controllers by using a super-twisting sliding mode control [12]. The offline trajectory planning method was proposed by using parabolic and cubic curves. A discontinuous state transformation was introduced using point to point tracking algorithms for trajectory planning with a set of way-point [13]. The closed-loop system was simulated using MATLAB environment and validated by turtlebot3 simulation. Position and localization were solved by leader-based bat algorithm which can operate with particle swarm optimization [14, 15]. The method was proven that it had less error than 21% compared with the adaptive Monte Carlo localization algorithm. A Gaussian mixture spline trajectory was used as a dataset to generate trajectory and learn from previous experiences. Trajectory planning for the mobile robot needs to move smoothly and rapidly to the target path planning safely and economically. It can be defined as global and local path planning according to the mastery of environmental information. The global environment includes the grid method, topology method, geometric feature method, and mixed representation method. On the other hand, the local method includes sensor detection, visual sensor, lidar scanning sensor and radar sensor. The mobile robot path planning algorithm can be divided into three types; classical algorithms, bionic algorithms and artificial algorithm.

2. RELATED WORKS

Autonomous mobile robot (AMR) has received highly attention all over the world because of the industrial revolution and digital transformation. The AMR two-wheel robot is

mainly responsible for services and transportation. In this research, the virtual non-holonomic of the AMR 2 wheel robot with differential drive system is studied. Trajectory planning is developed to move smoothly, rapidly, and economically. The previous paper reviews are concerned only with mobile robots, summarized and criticized to draw up the new concept of trajectory planning. A mobile robot might designed by functionalities and loads. It can be four steered wheels, which is controlled by pc-based controller [16], or two wheels with differential drive control [17], fuzzy logic control [18], and PID control in dynamic environments [19] whereas kinematics models are used to link with mechanical structural movement. Odometry is used for control wheels moved in real-time. A sliding mode dynamic controller has been applied to control wheeled mobile robots to move forward in the trajectory tracking target [20]. Adaptive controller can be used to guide mobile robot for trajectory tracking [21] with linear and angular velocities by kinematic and dynamic model. The concept can reduce error but increase performance for load transportation conditions. The stability was analysed using Lyapunov theory. Nonlinear control was applied to AGVs and AMR. The trajectory tracking objective was to describe and reach to the accomplished tasks. Heavy load transportation and high-speed movements need vehicle dynamics motion combined kinematic and torque control law for nonholonomic mobile. A robust adaptive controller is based on neural networks to deal with disturbances and non-modeled dynamics[22]. An adaptive fuzzy logic system-based controller was used to deal with uncertainty which can tune parameter on-line. The dynamic model included the actuator dynamics controlled with voltages of the motors. Neural Network based approach used to compensate fault parameters of the robot movement as well as identifying and controlling signals with linear and angular velocities. Real-time implementation required a high performance computer based on a multiprocessor system. Robot learning control used a multiprocessor system for real-time processing to address tracking problems in mobile robots operation to deal with environmental disturbances and parameter uncertainties [3, 23]. Radial Basis Function Networks (RBFNs) are considered as a non-linear approximation capacity for modeling the kinematic behaviour and reducing unmodeled contributions to tracking errors. Non-linear optimization techniques are used for learning algorithm for multilayer neural networks. Precise localization of the mobile wheeled robot used sensor fusion of odometry, visual artificial landmarks and inertial sensors. Gaussian process regression was applied as a batch nonlinear for estimation trajectory to obtain maximize accuracy [24], [25] as well as combined with Bayesian filtering method [26], [27]. Accurate localization is critical for mobile robot navigation. The simulation localization and mapping (SLAM) method track signals from laser radar or sonar to form a probabilistic map of points of obstacles. A mobile robot has two sensors for inertial measurement units and wheel encoders. Sensor modality such as ultrasonic and sonar sensor, visual sensor, vision sensor and RGB-D sensor was used to solve SLAM problems in dynamic environments [3], [28]. The SLAM was rapidly created in 3D by Lidar system by planar surface segments which can extract 360° field of view point clouds to provide a map [29]. SLAM coordination can create global maps by merging the local planar surface segments. It consists of concurrent construction of a model of the environment by mapping and estimation of the robot's moving state [30], [31]. SLAM was used for temporal pose interpolation using continuous time framework for creating trajectory smoothly and using the parametric method of finite element representation [31]. The Kalman filters model is used for fault-tolerant sensor fusion in mobile

robots. The fault tolerance scheme operates with the IMU and odometry sensors which are received by the Kalman filter based on EKF and UKF [27]. A differential drive control system offers advantages of rapid speed and movement in various directions. If both wheels are driven at the same direction and speed, the robot moves in a straight line. If both wheels are turned at equal speed in opposite directions, the robot rotates by the central point of the axis as shown in Fig.1. The robot can move precisely in any direction due to the rate and rotational direction control of the two driven wheels.

Trajectory path planning is a significant task for the AMR working together with mapping or SLAM. Kohonen's self-organizing map was studied by [32] using artificial neural networks. An autonomous tracking control system was developed by using a teleoperation approach with great flexibility and adaptability [33]. It can provide compensation information in the dynamic environment and connect to interactive controlled devices that cooperate with navigation. AMR control applies non-holonomic constraints from the kinematics links using a Lyapunov asymptotically stable tracking controller. The robot's path serves to guide the robot to its desired initial state. However, there are numerous possible paths that are given the work space in which the AMR can move.

Local and global path planning are designed and used to associate the path planning algorithms such as graph search, C-space search, and heuristic search to obtain optimization. The AMR is driven from one position to another within a certain region of the workspace [34]. Evolutionary algorithms, such as ant colony algorithm, are currently of interest for optimization and graph creation. The objective of the path planner is to minimize displacement by producing the optimal path and ensuring collision avoidance in path planning. A cost map is used to assign different values of cost to different accessible areas. Modeling methods are used to search for an optimal path. The robot surface interaction model works with kinematic structure and dynamic behaviour when the AMR moves. The model can work with differential drive of the motors [35]. AMR motion needs optimal trajectories with a curvature pattern which was used in the Hamiton-Jacobi formulation [36]. Optimal times to reach the goal state within a constrained environment can be achieved using a nonlinear model [37]. Genetic algorithm path planning was used for optimal AMR path in terms of time and displacement using a directed acyclic graph [38]. A swarm optimization algorithm using vector coevolving particles was developed with two swarms and two layers [39]. The top layer consists of all particles, whereas the bottom layer includes particles. The following equation shows the dynamic programming to obtain the static Hamilton-Jacobi-Bellman and the first order of a nonlinear partial differential equation. Path smoothing is the ultimate goal for AMR movement working with robot navigation to deal with static and dynamic obstacles [40]. Energy consumption is also concerned by AMR design [41] which is varied by different trajectory planning. It concerns on an optimal control formulation for energy-conscientious trajectory generation for the AMR using numerical control.

3. METHOGOLOGY

According to the previous reviewed paper, many articles presented the advanced and modern models and methods for the trajectory planning of the AMR mobile robot, two wheels

with differential drive, under the smooth path and motion. It is obviously found that spline trajectory enables AMR to moves moothly. This paper studies and proposes a new model of B-spline trajectory modified using math modified generation on the simulation. A mathematical models and formula are created and run on the polynomial trajectory generator. There are four input variables: *q* = positions of trajectory vector, *qd* = velocities of the trajectory vector, *qdd* = acceleration of trajectory, *pp* = piecewise polynomial coefficients that achieve the waypoints. The MATLAB Simulink is created with a block diagram and run on the specific conditions of the input data, which is environment and constraints. It can be 2 dimensions of the *X* and *Y* map given point control and waypoints can be SLAM. Then, the path planning is assigned. In this study environment is given by the 2 dimensions of the *X* and *Y* maps. In the model development, a B-spline trajectory modified generator will create route based on the algorithm with constraints, optimal velocity of the differential drive system for each point and a route through the whole route of AMR motion. The output can be *X* and *Y* positions, velocity of the left and right motors, acceleration and the piecewise polynomial vector.

3.1. SIMULINK MODEL

This section shows the model of the B-spline trajectory modified planning based on the path planning assignment of the way points. There are 8 waypoints starting from *P1–P8*. Firstly, the block control diagram is created with configuration of the dynamic control parameters for each block. The first block is important, as is the contribution of this development. The coding, algorithm, formula and parameters are set inside the model before it can run.

3.1.1. NON-HOLONOMIC AMR DYNAMIC MODEL

The non-holonomic AMR dynamic model is shown in the Fig.1 with two wheels and one caster wheel. The velocity is dependent on the diameter of the wheel and the power of the motor, as well as the capacity of the speed. Differential is optimized for the AMR' speed in various trajectories using the concept of B-spline modified pattern. The assumption is that AMR does not move backward. It can turn left and right. It aims to move fast but stable particularly through the curvature path.
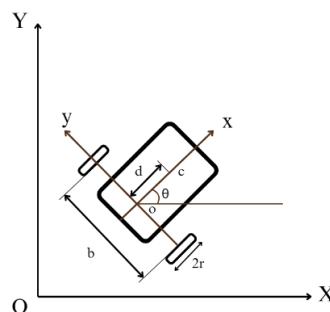

Fig. 1. Model of the non-holonomic wheel mobile robot

The kinematic model of the non-holonomic wheel robot can be written as a linear velocity (v), angular velocity (ω) and angle (θ) as in the (1).

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{1}$$

In case of non-slip, the relationship between linear velocity and angular velocity of right and left wheel can be described as the equation (2) and (3).

$$|v| = \frac{r}{2}(\dot{\emptyset}_R + \dot{\emptyset}_l) \tag{2}$$

$$|\omega| = \frac{r}{2b}(\dot{\emptyset}_R - \dot{\emptyset}_l), \tag{3}$$

Where v is the linear velocity of the WMR, ω is the angular velocity, r is the radius of the wheel, b is distance between the wheels and its center, $(\emptyset_R, \emptyset_l)^\cdot$ is the angular velocity of the right and left wheels respectively.

The dynamic equation is adopted as the general formula of the mobile robot as shown in (4).

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + G_m(q) = B(q)\tau - A(q)^T\lambda \tag{4}$$

Where: $M(q)$ represents the inertia matrix,

$\quad V(q,\dot{q})$ is the matrix of coriolis and centrifugal forces,

$\quad G_m(q)$ is gravitational force τ_d is external and disturbance,

$\quad B(q)$ represents the input of the transformation matrix,

$\quad A(q)^T\lambda$ is the matrix associated with such restriction.

$$q = [x, y, \theta, \phi_r, \phi_l]^T \tag{5}$$

Non- holonomic constraints can be defined as the (6)–(12).

$$\dot{x} + b\dot{\theta}cos(\theta) + r\dot{\emptyset}_r cos(\theta) = 0 \tag{6}$$

$$\dot{y} + b\dot{\theta}sin(\theta) + r\dot{\emptyset}_r sin(\theta) = 0 \tag{7}$$

$$\dot{x} - b\dot{\theta}cos(\theta) + r\dot{\emptyset}_l sin(\theta) = 0 \tag{8}$$

$$\dot{y} - b\dot{\theta}sin(\theta) + r\dot{\emptyset}_l sin(\theta) = 0 \tag{9}$$

$$(3)+(4) \quad \dot{x} = \frac{r}{2}cos(\theta)(\dot{\emptyset}_l - \dot{\emptyset}_r) \tag{10}$$

$$(5)+(6) \quad \dot{y} = \frac{r}{2}sin(\theta)(\dot{\emptyset}_l - \dot{\emptyset}_r) \tag{11}$$

$$(3)-(5) \quad \dot{\theta} = \frac{r}{2b}(\dot{\emptyset}_l - \dot{\emptyset}_r) \tag{12}$$

We can create matrix A(q) which satisfied non-holonomic constraints as (13)

$$A(q)\dot{q} = 0_{[3x1]} \tag{13}$$

$$A(q) = \begin{bmatrix} \boldsymbol{cos(\theta)} & \boldsymbol{sin(\theta)} & \mathbf{0} & \frac{r}{2} & -\frac{r}{2} \\ -\boldsymbol{sin(\theta)} & \boldsymbol{cos(\theta)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \frac{r}{2b} & \frac{r}{2b} \end{bmatrix} \tag{14}$$

From the Euler Lagrangian we can derive

$$\frac{d}{dt}[A(q)\dot{q}] = 0_{[3x1]} \tag{15}$$

We can simplify the derivative as $A(q)\dot{q} + \dot{A}(q)q = 0_{[3x1]}$

From the dynamic equation, it can be presented in the form of $\ddot{q}$ as (16)

$$\ddot{q} = M^{-1}(q)[B(q)\tau - v(q,\dot{q}) + A(q)^T\lambda] \tag{16}$$

From (16), we can substitute $\ddot{q}$ (17) and get Lagrange multiplier as the following equation.

$$\lambda = -[A(q)M^{-1}(q)A(q)^T]^{-1}[A(q)M^{-1}(q)(B(q)\tau - v(q,\dot{q})) + \dot{A}(q)\dot{q}] \tag{17}$$

3.1.2. B-SPLINE CURVE GENERATION

In robotics, a B-spline curve is used in path planning and trajectory generation. When a robot needs to move from one point to another, the generated path may contain sharp turns or abrupt changes in direction, leading to inefficient or jerky motion. By applying B-spline curves to smooth out these paths, robots can achieve smoothness and continuous movements, which are often crucial for tasks requiring precision and efficiency.

$$\dot{\boldsymbol{q}}_r = \left[\dot{\boldsymbol{x}}_r, \dot{\boldsymbol{y}}_r, \dot{\boldsymbol{\theta}}_r, \dot{\varnothing}_{\boldsymbol{r_r}}, \dot{\varnothing}_{l_r}\right]^{\mathbf{T}} \tag{18}$$

From the input of b-spline curve generation, there are 3 dataset which are *q, qd, qdd* which contain position, velocity and acceleration of the robot in x and y coordinates. We can get $\dot{x}_r, \mathbf{v}_r, \dot{\boldsymbol{\theta}}_r$ as shown in the equation (19)–(21).

$$\dot{\boldsymbol{x}}_r = \boldsymbol{qdd}(\mathbf{1}), \dot{\boldsymbol{y}}_r \, \boldsymbol{qdd}(\mathbf{2}) \tag{19}$$

Furthermore, we can define desired linear and angular velocity $\boldsymbol{v}_r, \boldsymbol{w}_r$ as

$$\mathbf{v}_r = \sqrt{\dot{\boldsymbol{x}}_r(t)^2 + \dot{\boldsymbol{y}}_r(t)^2} \tag{20}$$

$$\dot{\boldsymbol{\theta}}_r = \mathbf{w}_r = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \dot{y}_r(t)\ddot{x}_r(t)}{\mathbf{x}_r(t)^2 + \mathbf{y}_r(t)^2} \tag{21}$$

From the relation between robot velocity and wheel velocity, it can be represented by the equation of $AB = X$.

$$X = [\mathbf{v}_r \; \mathbf{w}_r]^T \tag{22}$$

$$A = \begin{bmatrix} \dfrac{r}{2} & \dfrac{r}{2} \\ \dfrac{r}{2b} & -\dfrac{r}{2b} \end{bmatrix} \tag{23}$$

$$B = [\dot{\phi}_r \quad \dot{\phi}_l]^T \tag{24}$$

Then, we can get the angular velocity of each wheel as shown in the (25).

$$B = A^{-1}X \tag{25}$$

3.1.3. CONTROLLED DESIGN

Controller design is started by the schemetic concept of AMR functions. It concerns with velocity, positions, torque and trajectory. The concept is tested by Matlab Simulink. Figure 2 shows the AMR control system design. PID is applied to control stability. Differential drive is used with encoders to build odometry information. The data capture is velocity for the AMR of the left and right motor. Positions are obtained by 2 planes of X and Y axes as well as the motor's positions on the left and right. Figure 2 shows the dynamic model controller with simulink to B-spline modified generator and outputs. They are position (ourput1), velocity (output2) and acceleration (output3) as well as the output 4.
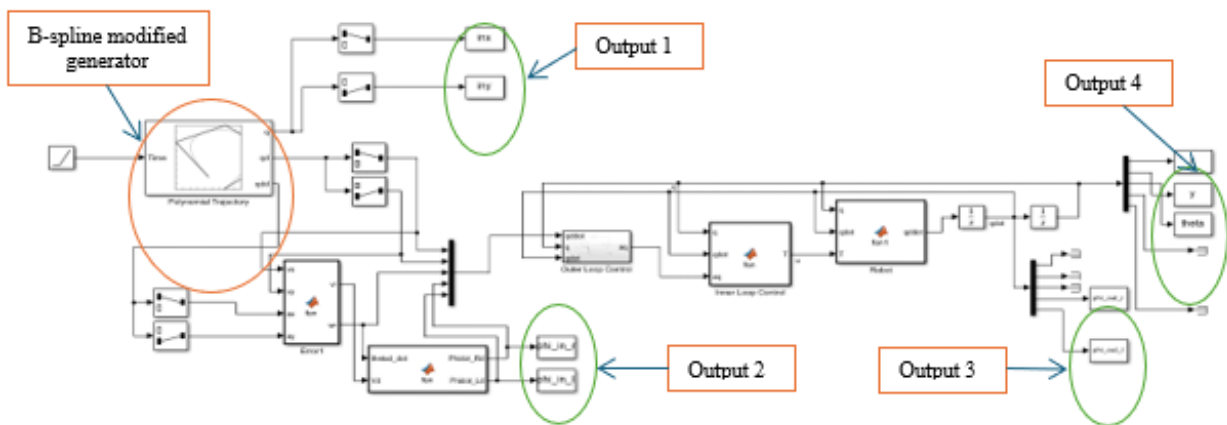

Fig. 2. AMR control system Simulink model

3.1.4. INNER CONTROL LOOP

In the equivalent control law, the inner control loop is shown in (26)

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + g(q) = \tau_{eq} \tag{26}$$

$V(q, \dot{q})\dot{q}, g(q)$ represents the feedback from $q, \dot{q}$ we can cancel the nonlinear component and re-derive as

$$M(q)\ddot{q} = M(q)\mathrm{A}_q \tag{27}$$

The output of the inner control loop with torque can be defined as the following.

$$\boldsymbol{\tau_{out} = mA_q + v(q,\dot{q})} \tag{28}$$

From the general dynamic equation, it can substitute previous equation since gravitation is negligible and can get the torque as in the (29).

$$\boldsymbol{B(q)\tau = M\ddot{q} + v(q,\dot{q}) - A(q)^T[A(q)M^{-1}(q)A(q)^T]^{-1}[A(q)M^{-1}(q)(B(q)\tau - v(q,\dot{q}) + \dot{A}(q)\dot{q}]} \tag{29}$$

Then, it can reduce the torque equation at the right hand side and it can get the (30), (31)

$$\boldsymbol{\tau = B^{-1}(q)A(q)^T[M^{-1}(q)A^T(q)]A(q)M^{-1}(q) + I_{5x5}]^{-1}[M\ddot{q} + V(q,\dot{q}) - H(q)\dot{A}(q)\dot{q} + H(q)A(q)M^{-1}(q)V(q,\dot{q})]} \tag{30}$$

$$\boldsymbol{H(q) = A(q)^T[A(q)M^{-1}(q)A(q)^T]^{-1}} \tag{31}$$

It can substitute (31) in to (30), then it can get (32)

$$\boldsymbol{\tau = B^{-1}(q)[A(q)^T[M^{-1}(q)A^T(q)]A(q)M^{-1}(q) + I_{5x5}]^{-1}[MA_q + V(q,\dot{q}) - H(q)\dot{A}(q)\dot{q} + H(q)A(q)M^{-1}(q)V(q,\dot{q})]} \tag{32}$$

3.1.5. OUTER CONTROL LOOP

The error of the system can be define as the outer control loop as shown in (33)

$$q_e = q_r - q \tag{33}$$

Where q_e is the error coordinate which convert to 0 or $\lim_{t \to \infty} q_e = 0$, and q_r is the reference coordinate and q is the robot coordinate.

From the feed forward and PID- controller, we can get linearization of $\ddot{q}$ as shown (34)

$$aq = \ddot{q}d + k_p q_e + k_d \dot{q}_e \tag{34}$$

4. RESULTS AND DISCUSSION

The section presents the path trajectory along the way points based on the B-spline curvature. The results are shown and discussed in detail. However, the AMR robot studied in this paper is shown by the virtual AMR trajectory motion to investigate dynamic control parameters such as velocity, acceleration, and positions of the robot located in the SLAM mapping. This simulation can be linked to the real AMR control by feedback control using robotic operation system (ROS) and odometry system connecting by TCP/IP protocol whereas the SLAM can be generated by the scanning lidar system. From the Simulink

dynamic model, the way point is sent like a pattern or random by dynamic route path which is located on the 2 planes of the *X* and *Y* as shown in the Table 1.

Table 1. The way points coordination of the robot path

Position/Point	*P1*	*P2*	*P3*	*P4*	*P5*	*P6*	*P7*	*P8*
X (cm)	0	0	30	50	60	80	60	60
Y (cm)	0	70	70	70	40	20	60	80

The Figure 3 (a) shows the desired path planning motion compared to the robot trajectory motion. The desired path planning is based on the waypoint assignment. From the simulation, the simulation shows that the desired path planning is very close to the robot motion. The graph shows the robot moving positions along the trajectory. Figure 3 (b) shows the AMR speed of velocity and time on the *X* and *Y* planes. Critical speed is shown in the *P3* and *P6* which represent the speed of the curvature. The speed is down when the curve is small. It is also found that the curve impacts the wheels. The counter clockwise (CCW) curve shows that the robot speed is faster than the desired speed whereas the clockwise (CW) curve enforces that the robot speed move slower. Figure 3 (c) shows the right wheel velocity. It is shown that the robot's speed is opposite to the left wheel both CW and CCW, because of the impact of the inner and outer wheels of the robot. Figure 3 (d) shows the linear velocity, which shows that the speed is steeply down from P1 to P3 and stable until speed down again at the *P6* before stepping up to the end point. Figure 3 (e) shows the angular speed which starts at with 0 rad/sec, and speed down to –2 rad/sec meaning that the robot wheel turns right at *P3*. Then, the robot moves in the opposite direction with left turn. The angular speed is highest at the *P6* and decrease to 0 rad/sec at the end. Figure 3 (f) shows acceleration starting from 0.5 m/sec^2 and then decreasing to 0 m/sec^2 at *P3*, fluctuating and accelerating again at the *P6*.
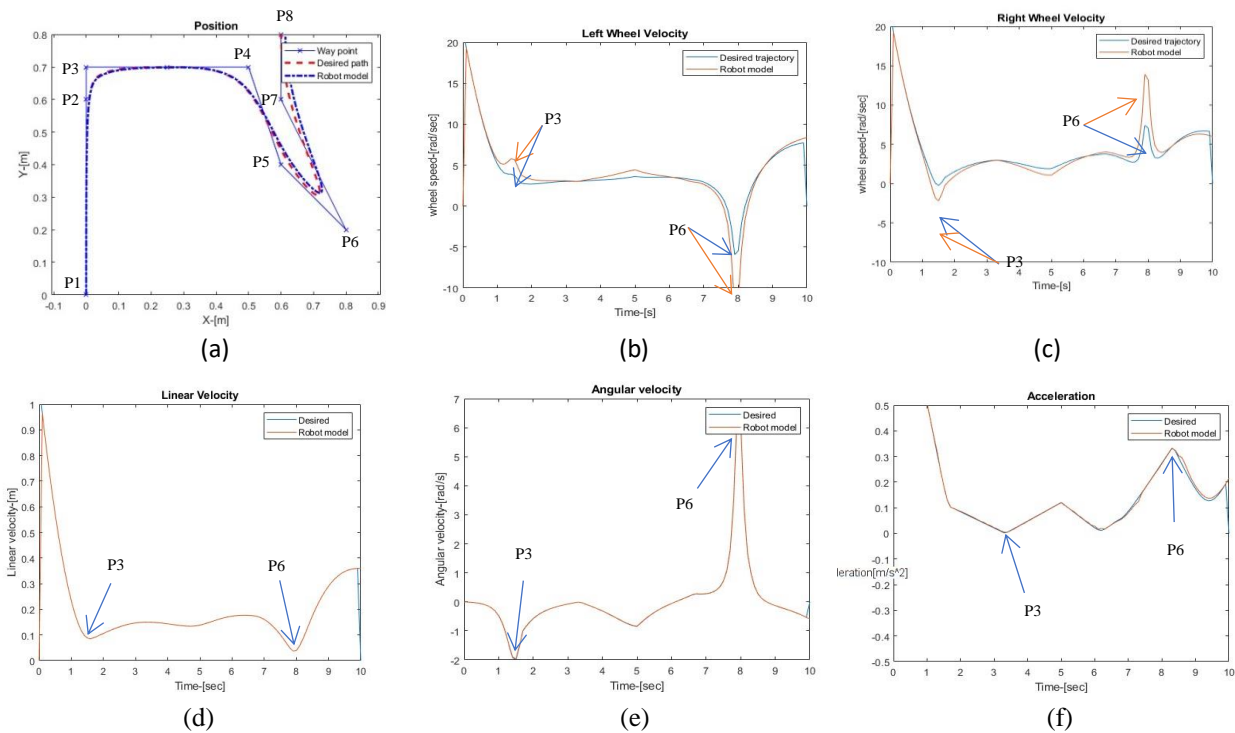

(a) (b) (c)

(d) (e) (f)

Fig. 3. B-splined trajectory simulation for the first test

The research effort is to validate one simulation assigning 7 positions of the *P1* to *P7* as shown in the Fig. 4 (a). It shows the second path planning of the simulation environment. The trajectory motion is compared between the desired path planning and the robot simulation. Figure 4 (b) and Fig. 4 (c) show the left and the right wheel speed, and the trajectories are very close to the desired path planning. The speed of the wheels is shown similarly with the previous simulation of the Fig.3. It is investigated that the B-spline shows that the virtual AMR moving faster on the first curvature whereas it moves slower on the second. It is because of the characteristics of the curve. The right wheel velocity moves in the opposite direction of the left wheel. The first curve shows that the robot moves slower than the desired path planning but it moves faster in the second curve or turn. It can be concluded that the left wheel of the virtual AMR moves faster than the right wheel when making a left turn in an anti-clockwise direction. In contrast, the virtual AMR moves faster with the left than the right wheel. Critically, the left wheel and the right wheel speeds are closely at the wide curvature whereas. Figure 6 (d) shows the linear velocity of the virtual AMR. It starts with high speed at linear trajectory and reduces sharply to the first curvature before increase the speed again. Figure 6 (e) shows the angular velocity. It is very high velocity at the beginning and fluctuates for the medium speed until it moves to the narrow curvature; the angular speed is very low before increasing to the medium speed at the end point. Figure 6 (f) shows the acceleration which decreases at the beginning and remains stable before the end. Then increases dramatically.
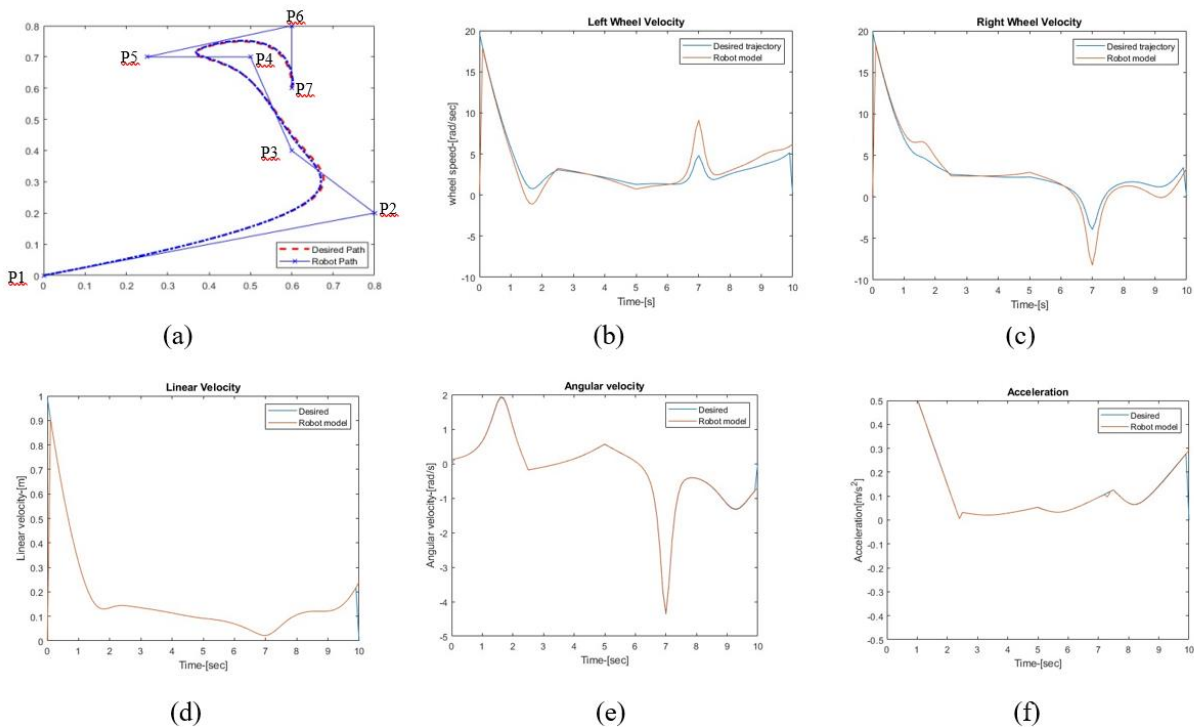

Fig. 4. Positions of *X* and *Y* in the 2D map (localization) related to the AMR movement's time

The results of the trajectory modified generation for smooth motion of the virtual AMR using the B-spline method is presented and discussed. The dynamic model was created and tested in the Simulink. Two different path planning environments were assigned and test for

satisfaction. The result can be applied to the nonholonomic AMR with a two wheels differential drive. The next step is to develop the real AMR robot and communication by ROS2 using the Linux operating system on the Ubuntu platform. The odometry system can be done by tracking data from a motor encoder. SLAM is created by connecting ROS2 to a scanning lidar sensor. In addition, artificial intelligence and machine learning can be applied.

5. CONCLUSION

The paper presents the concept of a B-spline trajectory modified generation model in order to create the virtual nonholonomic AMR two wheels with different drive based on path ways assignment that is a point-to- point. The path planning assigns the direction of the robot's motion from the starting point to the next one incrementally until to the end point. The trajectory is the robot's motion as a time series. The B-spline is autonomous generated by the dynamic Simulink model. Algorithms and code are developed within each block in the Simulink flow diagram. The modified trajectory is the AMR motion in the form B-spline with curve referred to as the control points (three points). The optimization of velocity control based on the curvature-modified path is generated by the dynamic model. The outputs are position, velocity both linear and angular velocity, acceleration, velocity of the right and left wheel motors. After the dynamic model was tested and compared with the desired and the robot motion. It was found that they are very close that can infer that the model is acceptable and high accuracy. Therefore, the model can be applied and linked with the actual AMR robot with sensor and actuator connection by internet TCP/IP.

REFERENCES

[1]　KATONA K., NEAMAH H.A., KORONDI P., 2024, *Obstacle Avoidance and Path Planning Methods for Autonomous Navigation of Mobile Robot*, Sensors, 24/11, https://doi.org/10.3390/s24113573.

[2]　KECZAN L., et al., 2024, *Technical Limitations of Organic Human-Robot Interaction (O-HRI) for Mobile Robots Moving Amongst Humans*, 2024 IEEE 21st International Power Electronics and Motion Control Conference (PEMC), IEEE, 1–6, https://ieeexplore.ieee.org/abstract/document/10726327/.

[3]　NEAMAH H.A., MAYORGA O.A., 2024, *Optimized TD3 Algorithm for Robust Autonomous Navigation in Crowded and Dynamic Human-Interaction Environments*, Results Eng., 24, 102874, https://doi.org/10.1016/j.rineng.2024.102874.

[4]　KOCH P.E., WANG K., 1988, *Introduction of B-Splines to Trajectory Planning for Robot Manipulators*, Model. Identif. Control Nor. Res. Bull., 9/2, 69–80, 1988, https://doi.org/10.4173/mic.1988.2.2.

[5]　RIBOLI M., CORRADINI F., SILVESTRI M., AIMI A., 2023, *A New Framework for Joint Trajectory Planning Based on Time-Parameterized B-Splines*, Comput.-Aided Des., 154, 103421, https://doi.org/10.1016/j.cad.2022.103421.

[6]　ALMUSAWI H., AL-JABALI M., KHALED A., KORONDI P., HUSI G., 2022, *Self-Driving Robotic Car Utilizing Image Processing and Machine Learning*, IOP Conf. Ser. Mater. Sci. Eng., 1256, 012024, https://doi.org/10.1088/1757-899X/1256/1/012024.

[7]　NEAMAH H.A., KHUDHAIR M.A., DHAIBAN M.S., 2024, *Navigating Attention-Centric: a Machine Learning Approach to EMG-Based Hand Gesture Recognition for Interactive RC Car*, 2024 IEEE 21st International Power Electronics and Motion Control Conference (PEMC), 1–6, https://doi.org/10.1109/PEMC61721.2024.10726353.

[8]　KANO H., FUJIOKA H., 2018, *B-Spline Trajectory Planning with Curvature Constraint*, Annual American Control Conference (ACC), Milwaukee, WI: IEEE, Jun., 1963–1968. https://doi.org/10.23919/ACC.2018.8431703.

[9] BERGLUND T., BRODNIK A., JONSSON H., STAFFANSON M., SODERKVIST I., 2010, *Planning Smooth and Obstacle-Avoiding B-Spline Paths for Autonomous Mining Vehicles*, IEEE Trans. Autom. Sci. Eng., 7/1, 167–172, https://doi.org/10.1109/TASE.2009.2015886.

[10] CHANG S.R., HUH U.Y., 2015, *Curvature-Continuous 3D Path-Planning Using QPMI Method*, Int. J. Adv. Robot. Syst., 12/6, 76, https://doi.org/10.5772/60718.

[11] CHARLEBOIS M., GUPTA K., PAYANDEH S., 1999, *Shape Description of Curved Surfaces from Contact Sensing Using Surface Normals*, Int. J. Robot. Res., 18/8, 779–787, https://doi.org/10.1177/02783649922066556.

[12] BECERRA H.M., COLUNGA J.A., ROMERO J.G., 2018, *Simultaneous Convergence of Position and Orientation of Wheeled Mobile Robots Using Trajectory Planning and Robust Controllers*, Int. J. Adv. Robot. Syst., 15/1, 172988141875457, https://doi.org/10.1177/1729881418754574.

[13] IBRAHIM F., ABOUELSOUD A.A., FATH ELBAB A.M.R., OGATA T., 2019, *Path Following Algorithm for Skid-Steering Mobile Robot Based on Adaptive Discontinuous Posture Control*, Adv. Robot., 33/9, 439–453, https://doi.org/10.1080/01691864.2019.1597764.

[14] NETO W.A., PINTO M.F., MARCATO A.L.M., DA SILVA I.C., FERNANDES D.D.A., 2019, *Mobile Robot Localization Based on the Novel Leader-Based Bat Algorithm*, J. Control Autom. Electr. Syst., 30/3, 337–346, https://doi.org/10.1007/s40313-019-00453-2.

[15] NEAMAH H.A., AL ARDI D., KORONDI P., 2024, *Enhancing Autonomous Robot Perception for Precision Positioning and Localization*, 2024 IEEE/SICE International Symposium on System Integration (SII), IEEE, 1058–1063, https://ieeexplore.ieee.org/abstract/document/10417351/.

[16] OFTADEH R., AREF M.M., GHABCHELOO R., MATTILA J., 2013, *Mechatronic Design of a four Wheel Steering Mobile Robot with Fault-Tolerant Odometry Feedback*, IFAC Proc., 46/5, 663–669, https://doi.org/10.3182/20130410-3-CN-2034.00092.

[17] MUDENG V., et al., 2020, *Design and Simulation of Two-Wheeled Balancing Mobile Robot with PID Controller*, Int. J. Sustain. Transp. Technol., 3/1, 12–19, https://doi.org/10.31427/IJSTT.2020.3.1.3.

[18] YEKINNI L.A., DAN-ISA A., 2019, *Fuzzy Logic Control of Goal-Seeking 2-Wheel Differential Mobile Robot Using Unicycle Approach*, IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Selangor, Malaysia: IEEE, 300–304, https://doi.org/10.1109/I2CACIS.2019.8825082.

[19] DENG M., INOUE A., SEKIGUCHI K., JIANG L., 2010, *Two-Wheeled Mobile Robot Motion Control in Dynamic Environments*, Robot. Comput.-Integr. Manuf., 26/3, 268–272, https://doi.org/10.1016/j.rcim.2009.11.005.

[20] CHEN C.Y., LI T.H.S., YEH Y.C., CHANG C.C., 2008, *Design and Implementation of an Adaptive Sliding-Mode Dynamic Controller for Wheeled Mobile Robots*, Mechatronics, 19/2, 156–166, https://doi.org/10.1016/j.mechatronics.2008.09.004.

[21] MARTINS F.N., CELESTE W.C., CARELLI R., SARCINELLI-FILHO M., BASTOS-FILHO T.F., 2008, *An Adaptive Dynamic Controller for Autonomous Mobile Robot Trajectory Tracking*, Control Eng. Pract., 16/11, 1354–1363, https://doi.org/10.1016/j.conengprac.2008.03.004.

[22] NEAMAH H.A., BUTDEE R., 2024, *Optimization Modeling Parameters for Industrial AMR Slippage Using ANFIS System in Dynamic Environment*, Advanced in Creative Technology- added Value Innovations in Engineering, Materials and Manufacturing.

[23] ANTONINI P., IPPOLITI G., LONGHI S., 2006, *Learning Control of Mobile Robots Using a Multiprocessor System*, Control Eng. Pract., 14/11, 1279–1295, https://doi.org/10.1016/j.conengprac.2005.06.012.

[24] NEMEC D., ŠIMÁK V., JANOTA A., HRUBOŠ M., BUBENÍKOVÁ E., 2019, *Precise Localization of the Mobile Wheeled Robot Using Sensor Fusion of Odometry, Visual Artificial Landmarks and Inertial Sensors*, Robot. Auton. Syst., 112, 168–177, https://doi.org/10.1016/j.robot.2018.11.019.

[25] LINDGREN F., RUE H., LINDSTRÖM J., 2011, *An Explicit Link Between Gaussian Fields and Gaussian Markov Random Fields: The Stochastic Partial Differential Equation Approach*, J. R. Stat. Soc. Ser. B Stat. Methodol., 73/4, 423–498, https://doi.org/10.1111/j.1467-9868.2011.00777.x.

[26] SARKKA S., SOLIN A., HARTIKAINEN J., 2013, *Spatiotemporal Learning Via Infinite-Dimensional Bayesian Filtering and Smoothing: A Look at Gaussian Process Regression Through Kalman Filtering*, IEEE Signal Process. Mag., 30/4, 51–61, https://doi.org/10.1109/MSP.2013.2246292.

[27] HARTIKAINEN J., SARKKA S., 2010, *Kalman Filtering and Smoothing Solutions to Temporal Gaussian Process Regression Models*, 2010 IEEE International Workshop on Machine Learning for Signal Processing, Kittila, Finland: IEEE, 379–384, https://doi.org/10.1109/MLSP.2010.5589113.

[28] CHONG T.J., TANG X.J., LENG C.H., YOGESWARAN M., NG O.E., CHONG Y.Z., 2015, *Sensor Technologies and Simultaneous Localization and Mapping (SLAM)*, Procedia Comput. Sci., 76, 174–179, https://doi.org/10.1016/j.procs.2015.12.336.

[29] LENAC K., KITANOV A., CUPEC R., PETROVIĆ I., 2017, *Fast Planar Surface 3D SLAM Using LIDAR*, Robot. Auton. Syst., 92, 197–220, https://doi.org/10.1016/j.robot.2017.03.013.

[30] CADENA C., et al., 2016, *Past, Present, And Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age*, IEEE Trans. Robot., 32/6, 1309–1332, https://doi.org/10.1109/TRO.2016.2624754.

[31] ANDERSON S., DELLAERT F., BARFOOT T.D., 2014, *A Hierarchical Wavelet Decomposition for Continuous-Time SLAM*, 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China: IEEE, 373–380, https://doi.org/10.1109/ICRA.2014.6906884.

[32] HAN M.W., KOLEJKA T., 1994, *Artificial Neural Networks for Control of Autonomous Mobile Robots*, IFAC Proc., 27/4, 157–162, https://doi.org/10.1016/S1474-6670(17)46016-2.

[33] CARELLI R., FORTE G., CANALI L., MUT V., ARAGUÁS G., DESTÉFANIS E., 2008, *Autonomous and Teleoperation Control of a Mobile Robot*, Mechatronics, 18/4, 187–194, https://doi.org/10.1016/j.mechatronics. 2008.01.002.

[34] SÁNCHEZ-IBÁÑEZ J.R., PÉREZ-DEL-PULGAR C.J., GARCÍA-CEREZO A., 2021, *Path Planning for Autonomous Mobile Robots: A Review*, Sensors, 21/23, 7898, https://doi.org/10.3390/s21237898.

[35] SHUM A., MORRIS K., KHAJEPOUR A., 2015, *Direction-Dependent Optimal Path Planning for Autonomous Vehicles*, Robot. Auton. Syst., 70, 202–214, https://doi.org/10.1016/j.robot.2015.02.003.

[36] TAKEI R., TSAI R., 2013, *Optimal Trajectories of Curvature Constrained Motion in the Hamilton–Jacobi Formulation*, J. Sci. Comput., 54/2–3, 622–644, https://doi.org/10.1007/s10915-012-9671-y.

[37] BOKANOWSKI O., FORCADEL N., ZIDANI H., 2010, *Reachability and Minimal Times for State Constrained Nonlinear Problems Without Any Controllability Assumption*, SIAM J. Control Optim., 48/7, 4292–4316, https:// doi.org/10.1137/090762075.

[38] LEE J., KIM D.W., 216, *An Effective Initialization Method for Genetic Algorithm-Based Robot Path Planning Using a Directed Acyclic Graph*, Inf. Sci., 332, 1–18, https://doi.org/10.1016/j.ins.2015.11.004.

[39] CHEN Y., LI L., PENG H., XIAO J., YANG Y., SHI Y., 2017, *Particle Swarm Optimizer with Two Differential Mutation*, Appl. Soft Comput., 61, 314–330, https://doi.org/10.1016/j.asoc.2017.07.020.

[40] RAVANKAR A., RAVANKAR A., KOBAYASHI Y., HOSHINO Y., PENG C.C., 2018, *Path Smoothing Techniques in Robot Navigation: State-of-the-Art, Current and Future Challenges*, Sensors, 18/9, 3170, https:/ /doi.org/10.3390/s18093170.

[41] BAKSHI S., FENG T., YAN Z., MA Z., CHEN D., 2021, *Energy-Conscientious Trajectory Planning for an Autonomous Mobile Robot in an Asymmetric Task Space*, J. Intell. Robot. Syst., 101/1, 18, https://doi.org/10.1007/ s10846-020-01288-9.