Sigma J Eng & Nat Sci 35 (3), 2017, 389-403

Sigma Journal of Engineering and Natural Sciences
Sigma Mühendislik ve Fen Bilimleri Dergisi

Extended Conference Paper

POINT STABILIZATION TECHNIQUE USING MODEL PREDICTIVE CONTROL AND EXACT EUCLIDIAN DISTANCE TRANSFORM METHODS

Suat KARAKAYA[1]*, Gürkan KÜÇÜKYILDIZ[2], Hasan OCAK[3]

[1]*Kocaeli University Mechatronics Engineering, Umuttepe-KOCAELI;* ORCID:0000-0002-3082-0304
[2]*Kocaeli University Mechatronics Engineering, Umuttepe-KOCAELI;* ORCID:0000-0003-2744-0666
[3]*Kocaeli University Mechatronics Engineering, Umuttepe-KOCAELI;* ORCID:0000-0002-9917-4398

Received: 23.11.2016 Revised: 15.03.2017 Accepted: 07.04.2017

ABSTRACT

This paper presents a point stabilization method which is a hybrid form of Exact Euclidian Distance Transform (EEDT) and Model Predictive Control (MPC) theory. An optimal control problem is evaluated using a wheeled mobile robot's mathematical model. The model is represented in polar coordinates considering non-holonomic constraints. The optimal input parameters are obtained by solving a sequential quadratic cost function. The EEDT algorithm provides via-points starting from the initial position to the desired final coordinates. The line segments connected by these points form the shortest obstacle free static trajectory. Assuming that all of the occupied regions in the map are fully known, states of the mobile robot are constrained by avoiding the intersection of the obstacles and the planned path. Point stabilization phase is performed in a partial manner by applying MPC algorithm on consecutive via points. The overall stabilized trajectory between the start and goal is obtained by connecting the stabilized sub-trajectories. This approach provides a way to stabilize a trajectory without determining convex sub-regions on a non-convex map. The results on a simulated differentially driven mobile robot prove that the proposed algorithm gives satisfactory results for point stabilization problem in presence of static obstacles on a non-convex map.
Keywords: Model predictive control, point stabilization, static path planning, mobile robot.

1. INTRODUCTION

The mobile robot control has a rising interest in past decades. Kinematic model description caters simplicity for modelling a wheeled mobile robot (WMR). However, stabilizing the control laws based on the kinematic model is challenging. In addition to the non-holonomic constraints, physical limitations resulting from the actuators of the WMR should be taken into account. These requirements lead some difficulties to obtain smooth feedback control laws [1], [2]. In real-time applications, achieving the input and state limitations is not a trivial work, as the input control action tends to arise continuously. To overcome such problem, plenty of solutions dealing with adaptive and robust control of WMR motion planning task is presented in [3] and [4]. Even though time-variant control approaches offer considerable solutions to point stabilization problem of WMRs, real-time application is not feasible due to the requirement of online optimal-control requirements.

* Corresponding Author/Sorumlu Yazar: e-mail/e-ileti: suat.karakaya@kocaeli.edu.tr, tel: (262) 303 32 54

Model predictive control (MPC) is proposed as a solution for point stabilization problem of WMRs in [5], [6]. MPC is not a new method but collaboration with motion control of WMRs is a recent topic. WMR motion planning and stabilization problem contains both state limitations resulting from the obstacles and input restrictions dealing with the mechanical limits. These constraints should be satisfied in real-time; therefore, the input and state limitations have to be considered simultaneously. MPC generates the input control signal based on a finite horizon strategy. The limitations on the input can be handled in a straightforward way while generating the control action. Besides, the state restrictions depend on the configuration of the convex regions on the map [7]. Limitations of both the state and input control signals, which respect the actuators limit, are the main issues of MPC. This feature makes MPC an appropriate scheme for complex and constrained environments in real-time [8].

MPC consists of three main components: system model, objective function and finite horizon. Composing these stages, MPC can be utilized to solve a finite optimal control problem in a receding horizon policy. In this paper, a non-holonomic WMR model is formulated as the plant model and the input is constrained with certain limits. The objective function of the constrained regulation problem is used to determine a collision free trajectory through the desired goal while taking into account of the kinematic constraints.

For stationary environments, many collision-free path-planning methods are studied in the literature [9], [10], [11], [12], [13], [14], [15]. One of the recent static path planning approaches is Exact Euclidian Distance Transform (EEDT) [14]. This method encompasses the static obstacle avoidance by establishing a visibility protocol between the free cells in the configuration space. This approach provides the shortest obstacle-free trajectory between two coordinates. The collision-free trajectory is generated by EEDT before executing MPC. Thus, the state restriction is not necessary for collision avoidance. EEDT algorithm provides the via-points through which the WMR should pass. The configuration space of the WMR is processed by utilizing these points between the start and goal coordinates.

Executing MPC algorithm on consecutive via-points provides an effective way to compute partially stabilized obstacle-free trajectories. The total path is evaluated by connecting these stabilized trajectory parts. This method makes the objective function less complex than state-constrained quadratic form.

Once the static via points are eliminated by EEDT, the only constraint is the goal orientation of via-points and kinematic limits of the WMR. The orientation of the WMR on a via-point is calculated by averaging the tangents of the line segments connected by the current point. This operation smoothens the transitions between the sub-trajectories along with the global target. Safety of the trajectory is guaranteed by EEDT; therefore, states do not have to be regarded in the objective function. The map on which the point stabilization is processed must satisfy convex conditions. To overcome this issue, the study in [5] offers splitting the map into convex half planes. The methods presented in [16],[17],[18],[19] have significantly high computational cost. Besides, these methods require large number of decision variables. In this study, we offer a new approach to obtain stabilized trajectories in both convex and non-convex situations. This method shortens the path and computational cost with respect to previous MPC based point stabilization schemes.

The paper is organized as follows: In section 2, the technical basis of the proposed scheme is explained. The experimental results are presented in section 3 and the results are concluded in section 4.

2. TECHNICAL BACKGROUND

A. Kinematic Model of the Wheeled Mobile Robot

The mathematical model of the wheeled mobile robot (WMR) given in Fig 1 is expressed in (1).

$$\dot{x} = v\cos\theta$$
$$\dot{y} = v\sin\theta \tag{1}$$
$$\dot{\theta} = w$$

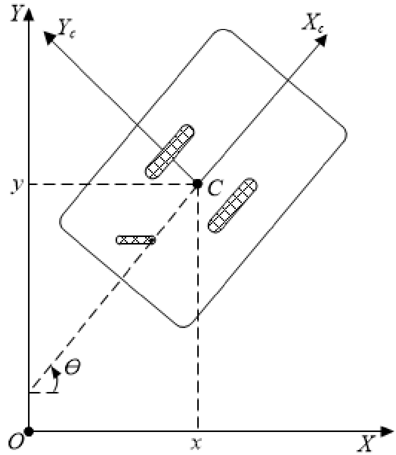

Figure 1. Cartesian coordinate system of the WMR

The mobile robot expressed within (1) is assumed to be a rigid platform with non-slipping wheels. The contact between the ground and the wheels is pure rolling. C and $\{X_c, Y_c\}$ denotes the center point of the wheels of the WMR and the local coordinate system of the WMR respectively. $\{x, y, \theta\}$ denotes the configuration of the WMR with respect to the global frame OXY. The input parameters are linear and angular velocities, where v is linear velocity and w is angular velocity of the WMR. The kinematic model of the WMR can be described in a brief form as given in (2).

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{2}$$

where **u** is the input control vector. The kinematic model given in (2) should be discretized to be able to study in discrete time conditions. The discrete-time model of the WMR is given in (3),

$$\mathbf{x}(k+1) = f_{\text{dis}}(\mathbf{x}(k), \mathbf{u}(k)) \tag{3}$$

where k is the sampling step.

The mathematical model of the WMR can be defined in polar coordinates by transforming the states $\{x, y, \theta\}$ to the discrete-time set $\{e, \psi, \alpha\}$. MPC based point stabilization scheme described in polar coordinates provides smoother trajectories for the WMR plants as the x and y states are decoupled from the common input v unlike the models described in Cartesian space. This coupling leads steady-state errors in x and y states [5]. The transformed parameters are given in (4),

$$
\begin{aligned}
e &= \sqrt{x^2 + y^2} \\
\psi &= \arctan(y, x) \\
\alpha &= \theta - \psi \\
\mathbf{x}_\mathrm{p} &= [e \quad \psi \quad \alpha]^\mathrm{T}
\end{aligned}
\tag{4}
$$

When the robot is forced to converge to origin, the first two state variables e and ψ both converge to 0 even if the final heading angle is non-zero. Therefore, making the two state variables converge to zero does not guarantee that the heading angle θ will also converge to zero. The third state variable α is included in the state vector so that, when $\mathbf{x}_\mathrm{p}$ is forced to converge zero, not only x and y converge to zero but the orientation angle θ also converges to zero.

B. Model Predictive Control (MPC)

MPC is an optimal control scheme, which utilizes the plant model to generate an optimal control structure by minimizing an objective function. In a prediction horizon, MPC uses the plant model to estimate the system behavior. An optimal input sequence is obtained and the first element of this sequence is feed backed to each recursive loop. The current optimal input is given to the plant to precede the control algorithm to the next step. This procedure is repeated over a prediction horizon recursively. The time window is shifted to the next state at each control loop and the first element of the obtained control sequence is assumed to be the optimal solution of the inputs: linear velocity (v) and angular velocity (w). The objective function is a quadratic function of the states and inputs as given in (5),

$$
\Phi_\mathrm{p}(k) = \sum_{i=1}^{N} \mathbf{x}_\mathrm{p}^\mathrm{T}(k+i \mid k)\mathbf{Q}\mathbf{x}_\mathrm{p}(k+i \mid k) + \mathbf{u}^\mathrm{T}(k+i-1 \mid k)\mathbf{R}\mathbf{u}(k+1-i \mid k)) \tag{5}
$$

where N is the prediction horizon of the MPC; **Q** and **R** are penalty matrices for weighting the states and the inputs respectively. The state vector $\mathbf{x}_\mathrm{p}$ denotes the polar states $[e \;\psi \;\alpha]^\mathrm{T}$. The optimization problem can be stated as to find $\mathbf{u}^*$ which minimizes the objection function. This issue is expressed in (6).

$$
\bar{\mathbf{u}}^* = arg\, \underset{\bar{\mathbf{u}}}{min} \{\Phi_\mathrm{p}(k)\} \tag{6}
$$

The expression $n(m|t)$ corresponds the value of n at the immediate time m predicted at the instant t. Thus, the control sequence obtained at t^{th} step deals with the future predictions from the step t to t+m. The control sequence is appeared as in (7).

$$
\bar{\mathbf{u}} \cong \{\mathbf{u}(i \mid i), \mathbf{u}(i+1 \mid i), \mathbf{u}(i+2 \mid i), \ldots, \mathbf{u}(i+N-1 \mid i)\} \tag{7}
$$

Solving (6) for each step along with the prediction horizon at the instant time i, optimal control sequence of the inputs is eliminated as given in (8).

$$
\bar{\mathbf{u}}^* \cong \{\mathbf{u}^*(i \mid i), \mathbf{u}^*(i+1 \mid i), \mathbf{u}^*(i+2 \mid i), \ldots, \mathbf{u}^*(i+N-1 \mid i)\} \tag{8}
$$

Considering the limitations of the actuators, input signal is restricted with an upper and a lower bound.

The constrained optimization problem in (6) is solved within a prediction horizon N and $\mathbf{u}^*(k|k)$ is selected as the instant solution. While minimizing the objective function, the state vector tends to zero, but if the state vector is desired to set another configuration, a series of rotations and translations are applied to the state vector. Point stabilization for a non-zero reference point is expressed in (9) and (10). Starting from the point $\{x_{init}, y_{init}, \theta_{init}\}$, assume that the target configuration is $\{x_{tar}, y_{tar}, \theta_{tar}\}$. The target point is translated to the origin and the translated initial point is rotated around the origin. The translated and rotated point is assumed the temporary initial pose while the origin {0, 0, 0} is the temporary target configuration.

$$\{x_{\text{init}}, y_{\text{init}}, \theta_{\text{init}}\} \xrightarrow{MPC} \{x_{\text{tar}}, y_{\text{tar}}, \theta_{\text{tar}}\} \tag{9}$$

$$\begin{bmatrix} x^*_{init} \\ y^*_{init} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{\text{tar}}) & -\sin(\theta_{\text{tar}}) \\ \sin(\theta_{\text{tar}}) & \cos(\theta_{\text{tar}}) \end{bmatrix} \times \begin{bmatrix} x_{\text{init}} - x_{\text{tar}} \\ y_{\text{init}} - y_{\text{tar}} \end{bmatrix} \tag{10}$$

After MPC is executed between $\{x^*_{\text{init}}, y^*_{\text{init}}, \theta^*_{\text{init}}\}$ and {0, 0, 0} the stabilized trajectory is obtained by the inverse transformation of the ones given in (10), where $\theta^*_{\text{init}} = \theta_{\text{init}} - \theta_{\text{tar}}$. Assuming the stabilized path coordinates between $\{x^*_{\text{init}}, y^*_{\text{init}}, \theta^*_{\text{init}}\}$ and {0, 0, 0} are denoted with $\mathbf{x}_0$ and $\mathbf{y}_0$, the inverse transformation is given in below.

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \cos(-\theta_{\text{tar}}) & -\sin(-\theta_{\text{tar}}) \\ \sin(-\theta_{\text{tar}}) & \cos(-\theta_{\text{tar}}) \end{bmatrix} \times \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{y}_0 \end{bmatrix} \tag{11}$$

As given in (11), the stabilized trajectory coordinates between custom points are determined as the vectors **x** and **y** in {XY} plane.

Given a sequence of via-points, the stabilization procedure given above can be applied on the point pairs consecutively, and the total trajectory can be created by connecting the partial trajectories. EEDT algorithm will be used to generate the via-points, which will be stabilized. A Matlab routine ‘quadprog’ is used for solving all quadratic equations within this study.

C. Exact Euclidian Distance Transform (EEDT)

Workspace of a WMR can be populated with obstacles; therefore, obstacle avoidance and safe trajectory planning modules are inevitable for motion planning of WMRs.

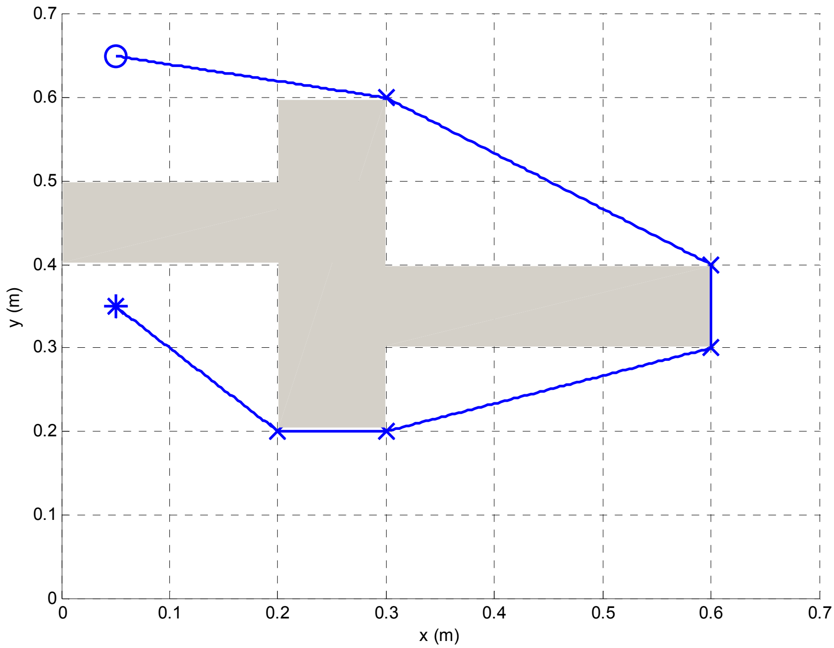

Figure 2. Unprocessed paths generated by EEDT

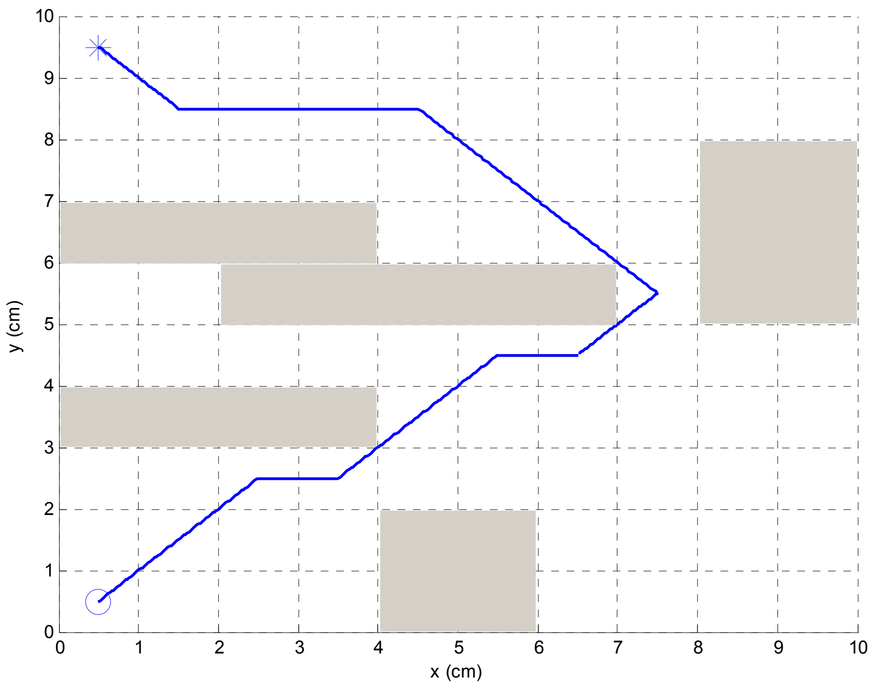

Figure 3. Euclidian distance transform (Jarvis)

Jarvis distance transform algorithm [14] and EEDT are similar as both of the methods are 8-neigbourhood based approaches. However, EEDT outperforms the travelled distance cost by following a visibility-based approach. Fig 2 depicts the path generated by EEDT where the crosses on the figure depict the via-points. Fig 3 shows the path determined by Jarvis' transform. As clearly seen in Fig 3, Jarvis' method limits the next move with the angle of 45^o. This is not an optimal solution for path planning problem though. The EEDT-based path is in an open polygonal format; therefore, corners of the trajectory are needed to be smoothened to increase the feasibility for real-time applications. For this purpose, each branch of the trajectory is processed by MPC algorithm independently. The goal orientation on a via-point is calculated by averaging the tangents of the branches connected at the via-point. The angle of the first via-point is given to MPC as the target orientation starting from the global start configuration. Then the next via-point is selected as the instant target by shifting the target points. The total point-stabilized trajectory is obtained by connecting the stabilized branches to each other.

A symbolic description of the hybrid point stabilization method is illustrated in Fig 4. Given a start and global target configuration, it is purposed that planning a collision-free trajectory which passes through the via-points determined by EEDT algorithm. The angular positions, on which the mobile robot has to carry at the via-points, are determined by averaging the tangents of consecutive static path branches connected by the current via-point. MPC algorithm is executed between each consecutive point pairs, therefore, the angular orientations that the WMR has to be settled on the via-points, are calculated partially. Starting from the initial configuration, the WMR visits the via-points by satisfying the pre-determined angular orientations respecting the input constraints. As clearly seen in Fig 4, the WMR is initially positioned at an arbitrary location with an orientation of 0^o. The main concern of the algorithm is to settle the WMR on the given target with an orientation angle 90^o. First, EEDT algorithm is executed to obtain the static path lines and the via-points, which the WMR has to visit. Then the orientation conditions of each via-point are calculated by averaging the consecutive path lines. Finally, MPC algorithm is implemented to the consecutive via-points by utilizing the orientations calculated at the previous step. The stabilized trajectory parts are combined to obtain the overall stabilized trajectory. The final obstacle-free trajectory satisfies the goal orientation and kinematic constraints of the WMR.

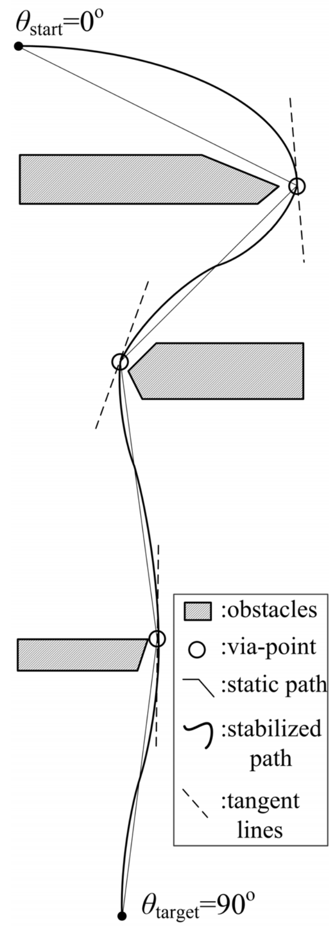

Figure 4. Obstacle free stabilized path by EEDT and MPC

3. RESULTS

In this section, simulation results of the algorithm are presented. The simulation platform is Matlab software. The developed scheme is compared with another MPC based point stabilization method in terms of trajectory length and processing burden. Stabilized trajectories generated by these methods are plotted on common two-dimensional figures. The gray regions in 2D figures correspond the static obstacles. The path-planning algorithm inflates the obstacles to improve safety. The hybrid method outperforms the path-planning effort and trajectory length, as the other scheme needs to divide the non-convex map into convex partitions. This process is also an optimization problem and increases the computational burden. In addition to map splitting, the conventional approach determines such via-points that make length of the trajectory unnecessarily long. The hybrid method handles this issue by utilizing the via-points determined by EEDT algorithm rather than convex map splitter. The hybrid method is executed on a three-dimensional mobile robot toolbox, which is explained in [20]. A simulated differential drive mobile robot is defined and the input controls are restricted regarding the simulated WMR's kinematic capabilities. An example solution is given in Fig 5. Considering $\mathbf{u}_{min}$={-0.5 m/s, -1.4 rad/s} and $\mathbf{u}_{max}$={0.5 m/s, 1.4 rad/s}, initial position and orientation of the WMR is {0 m, 6 m, 90^o}, where the final configuration is {0 m, 0 m, 0^o}. The planned trajectory shows the trajectory starts from the initial pose and settles to the final configuration successfully. The linear velocity is negative why the simulated WMR moves in reverse direction during its travel. This scenario illustrates a simple map without any obstacles. In fact, this operation is performed for each branch on unprocessed static path as given in Fig 6 and Fig 7.

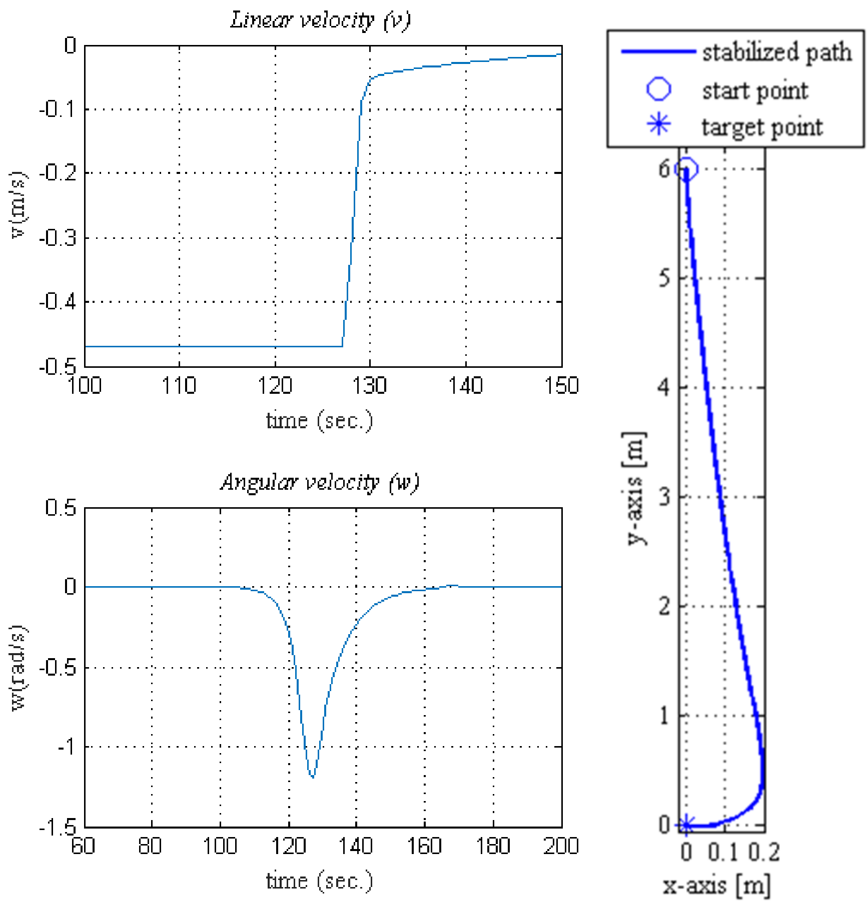

Figure 5. Input control signals and the stabilized path

The common use of MPC in point stabilization on existence of stationary obstacles, depends on the bounding the states as well as the input variables [5]. For such a task, it is necessary to split the configuration space in multi-convex sub-regions. In addition, upper and lower bounds of the states for each convex region have to be defined. The goal configurations of the states are also to be determined in every single convex region. This situation increases the length of the planned trajectory with increasing complexity of the configuration space. The proposed method decreases the length of the trajectory by avoiding the map splitting.

In Fig 6, a comparison between the conventional method [5], [6] and the proposed scheme is given. Convex regions for the conventional MPC based point stabilization scheme are given in (12).

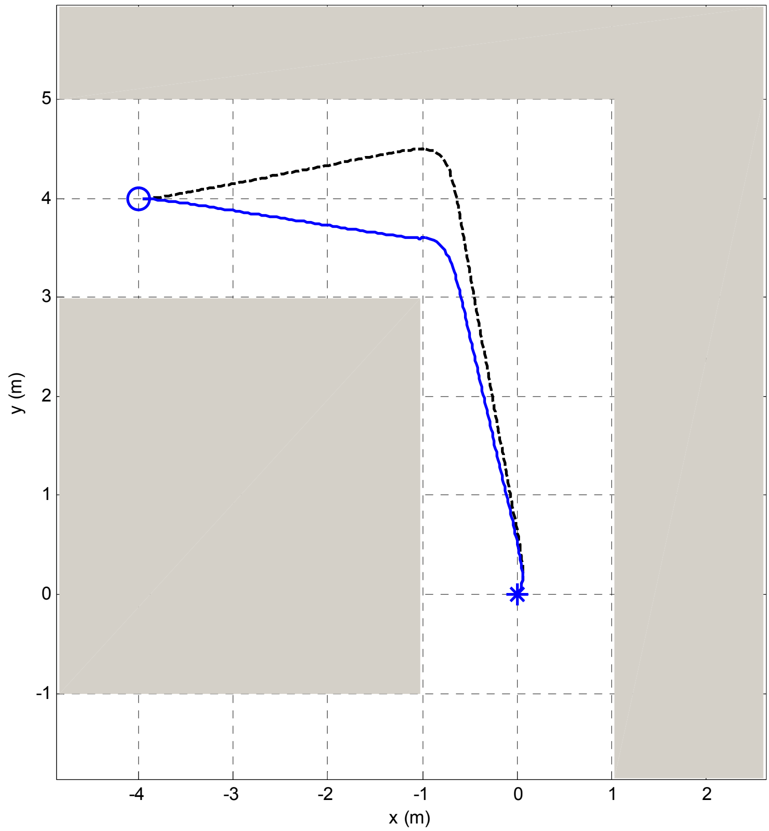

Figure 6. Planned trajectories with state-bounding and MPC-EEDT approaches

$$
\begin{aligned}
x(k\,|\,k) < -1 &: \begin{cases} x(k+j\,|\,k) \le 1, \\ 3 \le y(k+j\,|\,k) \le 5, \\ \mathbf{x}_r = [0\,4\,\frac{4\pi}{3}]^{\mathrm{T}} \end{cases} \\
x(k\,|\,k) \ge -1 &: \begin{cases} -1 \le x(k+j\,|\,k) \le 1, \\ y(k+j\,|\,k) \le 5, \\ \mathbf{x}_r = [0\,0\,\frac{4\pi}{3}]^{\mathrm{T}} \end{cases}
\end{aligned}
\qquad (12)
$$

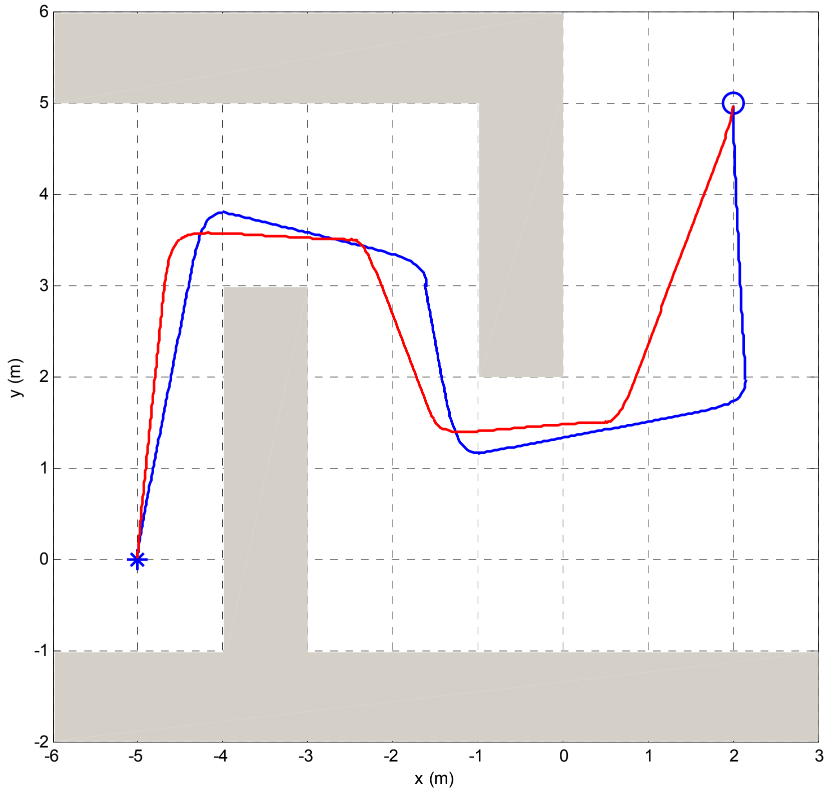

Figure 7. Planned trajectories with state-bounding and MPC-EEDT approaches

Fig 6 shows the trajectories of the robot in the XY-plane where occupied polygons demonstrates the stationary walls, dashed-line depicts the path generated by conventional MPC (state bounding) and continuous line depicts the path generated by the proposed scheme. The starting and goal coordinates are (-4,4) and (0,0) points respectively. The conventional method divides the configuration space in two convex parts where it is already an independent optimization problem. For this scenario the line, x = 1, defines the limit between two convex planes. For each half plane, different reference state vectors ($\mathbf{x}_r$) and state constraints are considered (12). The length of path generated by the proposed method is 15% shorter than the one produced by the conventional approach.

Another comparison between two methods is given in Fig 7 while the blue and red trajectories depict the conventional method and the proposed scheme results respectively. Convex splitting conditions are given in (13). In (12) and (13) $\mathbf{x}_r$ denotes the local target of the WMR in current convex region. The convex splitting is another optimization problem, which requires extra computational effort. Path length and cost are reduced approximately 15% and 30% respectively for the scenario given in Fig 7.

$$\begin{aligned}
&\begin{matrix} x(k\,|\,k) \geq -1 \\ y(k\,|\,k) \geq 3 \end{matrix} : \begin{cases} x(k+j\,|\,k) > 0, \\ y(k+j\,|\,k) > -1, \\ \mathbf{x}_r = [2\ 1\ \pi]^{\mathrm{T}} \end{cases} \\
&\begin{matrix} x(k\,|\,k) < -1 \\ y(k\,|\,k) < 3 \end{matrix} : \begin{cases} x(k+j\,|\,k) > -3, \\ -1 < y(k+j\,|\,k) < 2, \\ \mathbf{x}_r = [-2\ 1\ \frac{\pi}{2}]^{\mathrm{T}} \end{cases} \\
&\begin{matrix} -3 < x(k\,|\,k) < -1 \\ -1 < y(k\,|\,k) < 3 \end{matrix} : \begin{cases} -3 < x(k+j\,|\,k) < -1, \\ -1 < y(k+j\,|\,k) < 5, \\ \mathbf{x}_r = [-2\ 4\ \pi]^{\mathrm{T}} \end{cases} \\
&\begin{matrix} -4 < x(k\,|\,k) < -1 \\ 3 < y(k\,|\,k) < 5 \end{matrix} : \begin{cases} -4 < x(k+j\,|\,k) < -1, \\ 3 < y(k+j\,|\,k) < 5, \\ \mathbf{x}_r = [-5\ 4\ \frac{3\pi}{2}]^{\mathrm{T}} \end{cases} \\
&x(k\,|\,k) < -4 : \begin{cases} x(k+j\,|\,k) < -4, \\ -1 < y(k+j\,|\,k) < 5, \\ \mathbf{x}_r = [-5\ 0\ \frac{3\pi}{2}]^{\mathrm{T}} \end{cases}
\end{aligned} \tag{13}$$

The presented method is also verified on a three-dimensional Matlab simulation toolbox recently presented in [20]. This toolbox provides a visual effect for the analysis performed in 2D plots. The point stabilization in an obstacle-free map is illustrated in Fig 8. The simulated WMR is initialized at 60^{o} intervals and navigated through the origin of the map.

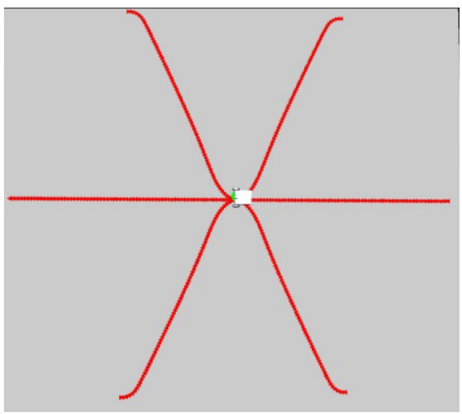

Figure 8. Point stabilization scheme for obstacle-free environments

Fig 9 illustrates that the simulated WMR starts from the initial configuration {45 m, 5 m, 10°} to the target {15 m, 90 m, 95°}. The average positioning error of 0.1 m and the angular error of 0.09° is achieved during this simulation.

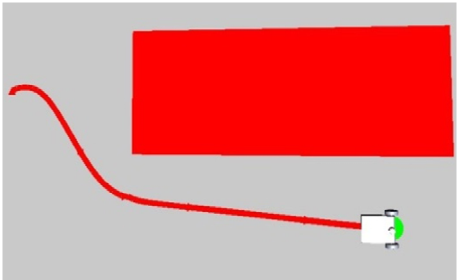

Figure 9. Point stabilization scheme for occupied environments-1

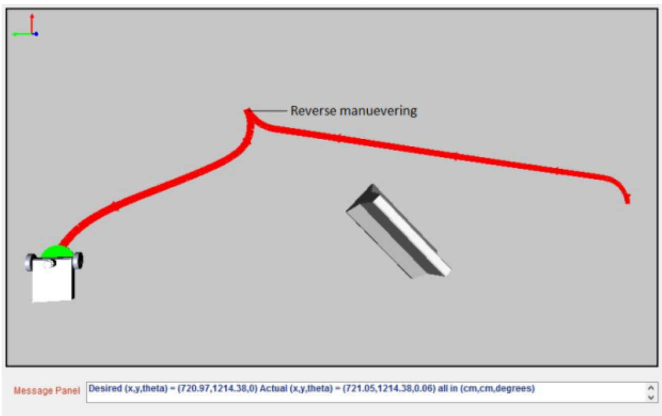

Figure 10. Point stabilization scheme for occupied environments-2

Another result is given in Fig 10. The goal configuration of the WMR is {72.09 m, 121.43 m, 0°} where the simulated WMR settles at the final configuration {72.10 m, 121.43 m, 0.06°}.The positioning error of the point stabilization scheme for this scenario is {0.01 m, 0 m, 0.06°}. As clearly seen in Fig 10, the simulated WMR makes a reverse maneuvering to achieve the final pose. The results prove the efficacy of the proposed point stabilization scheme in existence of obstacles around a WMR. The positioning and angular errors of the goal configurations are satisfactory for many applications.

Table 1. Experimental results

# Experiment	x-error (%)	y-error (%)	θ-error (%)
1	0.2	0.8	0.1
2	0.6	0.8	0.3
3	0.4	0.4	0.2
4	0.6	0.2	0.4

4. CONCLUSIONS

In this paper, a hybrid point stabilization method is proposed. EEDT and MPC algorithms are collaborated to calculate a trajectory consistent with the positioning constraints of a WMR. State limitations are considered by EEDT where the input constraints are considered by MPC. The optimization problem is handled by solving the objective function without convex splitting. This reduces the computational cost when solving the constrained objective function. The method is verified on a 3D Matlab simulation toolbox. For the future work, it is required to be implemented complexity analysis for real-time implementation. In addition, dynamic obstacle avoidance capability is needed to improve safety. The current version of this study provides an input-controlled static point stabilization scheme for researchers.

Acknowledgements

This study is a part of the project supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under the grant TUBITAK-113E777 and BIDEB-2211.

REFERENCES

[1] Brockett R.W., (1982) New directions in applied mathematics, *Springer-Verlag*, New York.

[2] Lages W.F., Hemerly E.F. (1998) Smooth time-invariant control of wheeled mobile robots, *Proc. Int. Congress on Systems Science*, 1-8.

[3] Oya M., Yu C.H., Katoh R., (2003) Robust adaptive motion force tracking control of uncertain non-holonomic mechanical systems, *IEEE Trans. Robotics and Automation*, 19, 175-181.

[4] Dixon W.E., Queiroz M.S., Dawson D.M., Flynn T.J., (2004) Adaptive tracking and regulation of a wheeled mobile robot with controller update law modularity, *IEEE Trans. Control Systems Technology*, 12, 138-147.

[5] Kühne F., Lages W.F., Silva J.M.G., (2005) Point stabilization of mobile robots with nonlinear model predictive control, *Proc. of the IEEE International Conference on Mechatronics and Automation*, 1163-1168.

[6] Alves J.AV., Lages W.F., (2007) Real-time point stabilization of a mobile robot using model predictive control, *Proc. IASTED International Conference of Robotics and Applications*, 58-63.

[7] Hsieh C.H., Liu J.S., (2012) Nonlinear model predictive control for wheeled mobile robot in dynamic environment, *Proc. International Conference on Advanced Intelligent Mechatronics*, 363-368.

[8] Sun S., Cui P., (2004) Path tracking and a practical point stabilization of mobile robot, *Journal of Robotics and Computer-Integrated Manufacturing*, 20, 29-34.

[9] Shokraneh K.M., Masehian E., (2016) Planning robot navigation among movable obstacles (NAMO) through a recursive approach, *Journal of Intelligent & Robotic Systems*, doi: 10.1007/s10846-016-0344-1.

[10] Zhengcai C., Yingtao Z., Qidi W., (2011) Genetic fuzzy + PI path tracking control of a non-holonomic mobile robot, *Chinese Journal of Electronics*, 20, 31-34.

[11] Rongbing L., Jianye L., Zhang, L., Hang Y. (2014) LIDAR/MEMS IMU integrated navigation (SLAM) method for a small UAV in indoor environments, Proc. IEEE Inertial Sensors and Systems Symposium (ISS), 1-15.

[12] Wu Z., Feng L., (2012) Obstacle prediction–based dynamic path planning for a mobile robot, *International Journal of Advancements in Computing Technology*, 4(1), 118-124.

[13] Tzafestas S.G., Deliparaschos K.M., Moustris G.P. (2010) Fuzzy logic path tracking control for autonomous non-holonomic mobile robots: Design of system on a chip, Robotics and Autonomous Systems, 58, 1017-1027.

[14] Carlos E.J., Federico P.E., Gabriel R.J., (2013) The exact Euclidian distance transform: a new algorithm for universal path planning, *International Journal of Advanced Robotic Systems*, 10, 1-10.

[15] Jarvis R.A., (1985) Collision free trajectory planning using distance transform, *Proc. National Conference and Exhibition on Robotics*, 187-191.

[16] Essen H. V., Nijmeijer H., (2001) Nonlinear model predictive control of constrained mobile robots. *Proc. European Control Conference*, 1157–1162.

[17] Mayne D.Q., Rawlings J. B., Rao C.V., Scokaert P.O.M. (2000) Constrained model predictive control: stability and optimality, Automatica, 36(6), 789-814.

[18] Ollero A., Amidi O. (1991) Predictive path tracking of mobile robots: application to the CMU Navlab. *Proc. IEEE International Conference on Advanced Robotics*, 1081–1086.

[19] Normey-Rico J. E., Gomez-Ortega J., Camacho E. F. (1999) A smith-predictor-based generalised predictive controller for mobile robot path-tracking, *Control Engineering Practice*, 7(6), 729–740.

[20] Karakaya S., Ocak H., Kucukyildiz G. (2017) A New Mobile Robot Toolbox for Matlab, *Journal of Intelligent & Robotic Systems*, DOI: 10.1007/s10846-017-0480-2.