2013 IEEE International Conference on Control Applications (CCA)
Part of 2013 IEEE Multi-Conference on Systems and Control
Hyderabad, India, August 28-30, 2013

On–Line Path Planning for an Articulated Vehicle based on Model Predictive Control

Thaker Nayl, George Nikolakopoulos and Thomas Gustafsson

Abstract— **In this article, a novel on–line path planning algorithm for an articulated vehicle, moving in a partially known and sensory based reconstructed environment, and relying on Model Predictive Control will be presented. The proposed algorithm belongs to the family of bug like path planning algorithms and has the capability to take under consideration the real dynamics of the articulated vehicle. Based on: a) an a priori knowledge of the current and the goal points, and b) a partial sensory based awareness of the surrounding environment, the algorithm is able to tune online the articulated steering angle in order to drive the front and the rear parts of the vehicle from avoiding collision with obstacles, while converging to the goal point. The proposed path planning algorithm is able to produce on–line the next reference way–point, solving the local and sub-optimal problem, while in the sequel a Model Predictive Controller is being utilized for creating the proper control signal, the rate of the articulated angle based on an error dynamics kinematic model of the vehicle. Multiple simulation results are being presented that prove the efficiency of the suggested scheme.**

I. Introduction

Automated articulated vehicles in the past and currently have received an increased scientific attention mainly for their utilization in the mining industry [1]. In these research efforts the aim has been the overall increase of the production rate, while making the working conditions safer for the human operators [2]. In most of the applications, these vehicles are remotely operated, while there is a continuous trend for increasing the autonomy level, especially in the area of path planning and obstacle avoidance as it is required form the vehicle: a) to have awareness of the changing surrounding environment, based on the onboard sensory systems, and b) autonomously plan their route towards the final goal point.

For the classical task of path planning, with an obstacle detection and avoidance capability, the simplest technique to solve the problem is the altering of the vehicle's orientation, while predicting a non collision path, based on: a) the vehicle's kinematic model, b) the sensing range, c) the safety range, and d) the a priori information of the surrounding environment. In this approach an optimal finite sequence of control inputs is being generated according to the initial vehicle position and the desired goal point. In the case where the path planning algorithm needs to be performed on–line, since the vehicle has a limited and in most of the time bounded sensory based awareness of the nearest surrounding environment, e.g. an exploration mission, local path planning which is a sub–optimal solution, needs to be generated and immediately applied to the vehicle [3].

Characteristic examples in the field of global path planning are the Road–Map algorithm [4], the Cell Decomposition [5], the Voronoi diagrams [6], the Occupancy Grinds [7] and the Potential Fields techniques [8], while in most of the cases, a final step of smoothing the produced path curvatures, by the utilization of Bezier curves is being utilized [9]. These algorithms optimize the overall traveled distance but may become computationally complex and maybe proven problematic when the global world model is inaccurate.

For the second case of a partially known and online reconstructed environment, the Bug family algorithms are well known mobile vehicle navigation methods for local path planning based on a minimum set of sensors and with a decreased complexity for online implementation [10]. Two of the most commonly utilized path planning algorithms in this category are the Bug1 and Bug2 [11]. Bug1 algorithm exhibits two behaviors; motion to goal with boundary following and a corresponding hit point and leave point, while Bug2 algorithm presents similar behaviors like the Bug1 algorithm, except from the fact that it tries to follow the fixed line from a start point to the goal, during obstacle avoidance. Other Bug algorithms that also incorporate range sensors are TangentBug [12], DistBug [13] and VisBug [14]. Tangent Bug algorithm is an improvement of the Bug2 algorithm since it is able to determine the shorter path to the goal using a range sensor with a 360^o infinite orientation resolution. DistBug has a guaranteed convergence and will find a path if one exists, while it requires the perception of its own position, the goal position and the range sensory data [15]. The VisBug algorithm, needs global information to update the value of the minimum distance to the goal point, during the boundary following and for determining the completion of a loop during the convergence to the goal.

The main disadvantage of all these algorithms is the fact that in the presented approaches the dynamics of the vehicle are considered to be the ones of as a single point, without any constraints in the movement, while the actual kinematics, which are important especially in the case of non–holonomic vehicles are being neglected.

The novelty of this article stems from the proposal of a new online and sub–optimal bug like path planning algorithm for the case of an articulated vehicle, which is able to consider: a) the real articulated kinematic equations formulated in a novel error dynamics model, b) physical constraints and limitations during the vehicle's movements, c) the initial and the goal configuration, d) the minimum and

The Authors are with Faculty of Control Engineering Group, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden, SE-97187.
Corresponding Author Email: thanay@ltu.se

978-1-4799-1559-0/13/$31.00 ©2013 IEEE

the total travel distance between, e) the safety and sensing distances, and f) proper obstacle detection and avoidance for producing on–line future reference way points. In the sequel a Model Predictive Control scheme (MPC) is being utilized that transforms the reference way points to the proper control action for the articulated vehicle, the rate of the articulated angle, based on an error dynamics kinematic model. The utilization of the MPC scheme is able to produce a smooth path tracking behavior with a relative low computational cost. In the related literature of path planning and control of non–holonomic vehicles, the MPC scheme has been utilized before in various implementations and different problem formulations [16]–[18], but to the authors best knowledge this is the first time that the MPC scheme has been combined with an online path planner based on the real articulated vehicle dynamics.

The rest of the article is organized as it follows. In Section II, the articulated robot kinematic and error dynamics model are being derived. In Section III, the proposed on–line path planning is being proposed in combination to the utilized MPC scheme, while in Section IV, multiple simulation results, subjected to various conditions are being presented that prove the efficacy of the proposed scheme in different arenas. Finally, the concluding remarks are provided in Section V.

II. Articulated Vehicle Error Dynamics Modeling

Articulated vehicles consist of two parts, a front and a rear with a width of w and l_1 and l_2 lengths correspondingly, linked with a rigid free joint, while each body has a single axle and the wheels are all non-steerable. The centers of gravity for these parts are being denoted as $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ respectively. The steering action is being performed on the joint, by changing the corresponding articulated angle γ, between the front and the rear of the vehicle as it is being indicated in Figure 1.

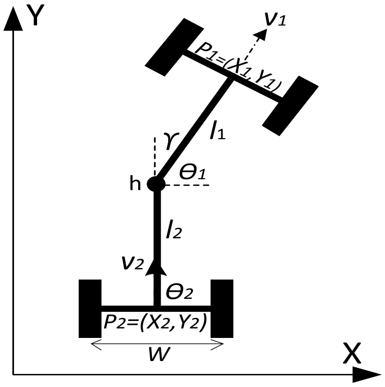


Fig. 1. Articulated robot's geometry and graphical representation of error dynamics transformation

For deriving the vehicle's kinematic equations, it is being assumed that: a) the steering angle γ remains constant under small displacement, b) the dynamical effects due to low speed (like, tire characteristic, friction, load and breaking force) are being neglected, c) each axle is being composed of two wheels that can be replaced by a unique wheel, and d) the robot moves on a plane without slipping effects, the robot's velocity is bounded within the maximum allowed velocity, which prevents the robot from slipping. From the geometrical characteristics of the vehicle it can be easily derived that the equations describing the translation of the articulated front part are,

$$\dot{x}_1 = v_1 \cos \theta_1 \tag{1}$$

$$\dot{y}_1 = v_1 \sin \theta_1 \tag{2}$$

The velocities v_1 and v_2 are considered to have the same changing with respect to the velocity of the rigid free joint of the robot and thus the relative velocity vector equations can be defined as,

$$v_1 = v_2 \cos \gamma + \dot{\theta}_2 l_2 \sin\gamma \tag{3}$$

$$v_2 \sin\gamma = \dot{\theta}_1 l_1 + \dot{\theta}_2 l_2 \cos \gamma \tag{4}$$

where $\dot{\theta}_1$ and $\dot{\theta}_2$ are the angular velocities of the front and rear parts respectively. By combining equations (3) and (4) it yields,

$$\dot{\theta}_1 = \frac{v_1 \sin\gamma + l_2 \dot{\gamma}}{l_1 \cos\gamma + l_2} \tag{5}$$

For the case that there is a steering limitation for driving the rear part according to the coordinates of the point $P_2 = (x_2, y_2)$, the relationship between P_1 and P_2 is provided by,

$$x_2 = x_1 - l_1 \cos\theta_1 - l_2 \cos\theta_2 \tag{6}$$

$$y_2 = y_1 - l_1 \sin\theta_1 - l_2 \sin\theta_2 \tag{7}$$

In the presented approach, the manipulated variable is the rate of the articulated angle, while the vehicle's speed is considered to be constant, while the kinematic model can be presented in a general state space form as,

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \tag{8}$$

Based on [19], it is being assumed that the vehicle develops a steady-state motion turning, where the rates of the orientation change are being provided by the following equations and R is the curvature of the reference path,

$$r_1 = \frac{v_1}{\dot{\theta}_1}, \quad r_2 = \frac{v_2}{\dot{\theta}_2} \tag{9}$$

As it has been extensively presented in [20], the system representation can be formulated as a set of error dynamics, described by the following parameters: curvature e_c, heading e_h and displacement e_d errors, also depicted in Figure 2. The curvature error is the difference between the curvature of the desired and the actual paths, heading error is the angle between the centers of the desired and actual paths with the front center of the robot and the displacement error is the difference between the two paths, as it has been defined in Figure 2. By defining an error state vector as

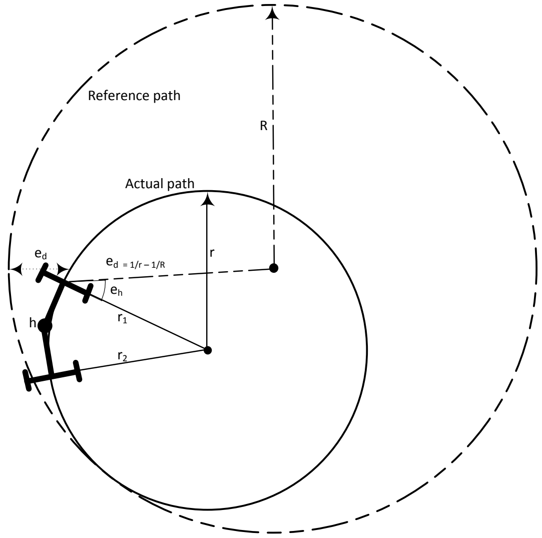


Fig. 2. A graphical representation of the error dynamics transformation

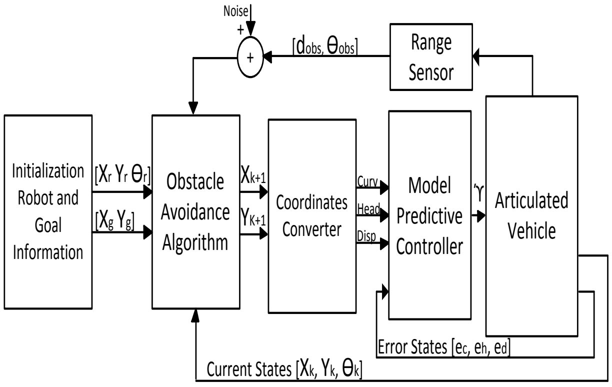
$\dot{\mathbf{x}}_e = [\dot{e}_c \quad \dot{e}_d \quad \dot{e}_h]^T$, the error dynamics description of the vehicle can be presented as it follows,

$$\begin{aligned} \dot{e}_c &= k_1 \; \dot{\gamma} \\ \dot{e}_h &= v \; e_c + k_2 \; \dot{\gamma} \\ \dot{e}_d &= v \; e_h \end{aligned} \tag{10}$$

where, $k_1 = \frac{1}{l_1 + l_2}$ and $k_2 = \frac{l_2}{l_1 + l_2}$ are constants. In this formulation it has been assumed that the three state variables are measured and that direct measurements for both the velocity and the position of the robot can be performed based on the on–board sensory system and thus no estimation action should be applied to the system. As it will be presented in Section IV the proposed MPC is utilizing this state space representation for generating the proper rate of articulation angle for controlling the articulated vehicle.

III. On Line MPC based Path Planning for Articulated Vehicle

The proposed online path planning algorithm can be applied for the objective of moving a vehicle from a starting point to the goal point, while detecting and avoiding identified obstacles based on the real vehicle's dynamic equations of motion and a Model Predictive Controller. As a common property of the Bug like algorithms, the proposed scheme initially faces the vehicle towards the assumed constantly known goal point. In the presented algorithm it has also been assumed that the vehicle is able to online sense the surrounding environment, included in a sensing radius of r_{obs}. During translation, the path planner is on-line generating future way points that act as a reference path for the utilized MPC. Before feeding the MPC with the reference coordinates, a proper conversion is taking place from the cartesian space to the curvature, heading and distance space, the states of the vehicle's error dynamic states [20]. The overall proposed concept of path planning and MPC control is depicted in Figure 3.



Noise
+
+
$[d_{obs}, \Theta_{obs}]$
Range Sensor
Initialization Robot and Goal Information
$[X_r\ Y_r\ \Theta_r]$
$[X_g\ Y_g]$
Obstacle Avoidance Algorithm
X_{k+1}
Y_{K+1}
Coordinates Converter
Curv
Head
Disp
Model Predictive Controller
Υ
Articulated Vehicle
Error States $[e_c, e_h, e_d]$
Current States $[X_k, Y_k, \Theta_k]$

Fig. 3. The combined on–line path planning and MPC for an articulated vehicle

As it can be observed from this diagram the algorithm starts by defining the current position and orientation of the vehicle, denoted by $[X_r, \; Y_r, \; \theta_r]$ and the final goal position denoted by $[X_g, \; Y_g, \; \theta_g]$. Based on the onboard sensory system, the vehicle identifies the surrounding environment and obstacles and generates on–line the way points for reaching the goal destination by solving the local and by definition sub–optimal path planning problem. In the sequel the way points are been translated to references for the error dynamics based MP controller. The control scheme generates the rate of articulated angle, which is the control signal for the vehicle. In the presented architecture it is assumed that accurate and continuous position and orientation measurements as well as corresponding updates are being provided, without any loss of generality.

The assumed sensory system is able to provided a partially bounded information on the surrounding environment due to the assumption of a limited sensing range $r_{obs} \in \Re$, while these measurements are being transformed to relative distances and angles from the articulated robot to the identified surroundings or obstacles, defined as $d_{obs} \in \Re$ and $\theta_{obs} \in \Re$ correspondingly. It should be noted that in the presented approach all the obstacles and the surrounding environment are being considered as point clouds in a 2–dimensional space, while overlapping obstacles are being merged and represented by a single and unified obstacle. In the path derivation a safety distance has been also considered, denoted as $r_{min} \in \Re$, for protecting the vehicle from approaching close to the obstacles. In the consideration of this safety distance, the algorithm calculates the related distance and angle from the identified obstacle, denoted as $d_{min} \in \Re$ and $\theta_{min} \in \Re$ and tunes the path planning accordingly. The notations utilized and the overall concept of the proposed path planning algorithm are depicted in Figure 4.

The operation and the sequential execution of the proposed path planner can be summarized as it follows:

[Step 1: Initialization] Sample the current position and orientation of the vehicle $[X_k \; Y_k \; \theta_k]$ and define the goal $[X_g \; Y_g]$ points, the articulated vehicle's specific parameters $v, \; d_{min}, \; \theta_{min}, \; d_{obs}, \; \theta_{obs}$, and the vehicle's mechanical and

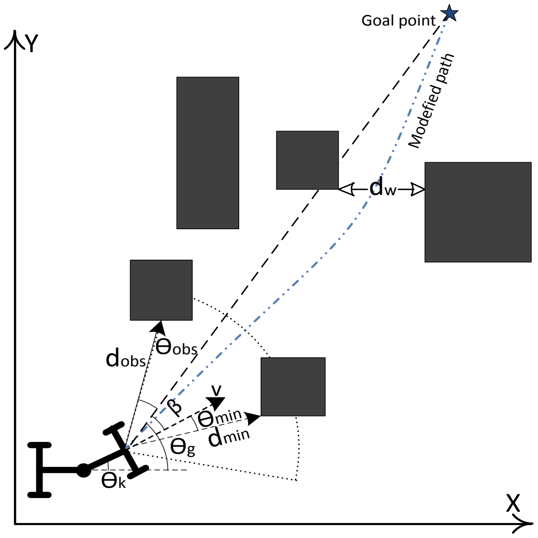


Fig. 4. Notations and overall concept of the proposed path planning algorithm.

physical constraints that need to be taken under consideration.

[Step 2: Path Update] Calculate the desired proper orientation of articulated vehicle's front part, in order to face the goal point during translation with constant speed as,

$$\begin{aligned} \theta_g(k) &= \arctan\frac{Y_g - Y_k}{X_g - X_k} \\ \beta(k) &= \theta_g(k) - \theta_k \end{aligned}$$

where θ_g is the angle between the line that connects the center of gravity of the vehicle's front part to the goal point and the X-axis. The β is the difference angle among the vehicle's orientation angle with the X axis and θ_g.

[Step 3: Desired Movement Update] Calculate the update of the vehicle's desired movement based on the following equations,

$$\begin{aligned} x(k) &= v\cos(\theta(k) + \beta(k)) \\ y(k) &= v\sin(\theta(k) + \beta(k)) \\ \theta(k) &= \beta(k) \end{aligned}$$

[Step 4: Scan Area] Calculate the smallest distance and the angle from the obstacles to the current location, as also the angle between the obstacles and the goal point, denotes by $\theta_{obs,g} \in \Re$, as it follows,

$$\begin{aligned} d_{obs} &= \sqrt{(X_{k+1} - X_{obs})^2 + (Y_{k+1} - Y_{obs})^2} \\ \theta_{obs} &= \arctan\frac{Y_{k+1} - Y_{obs}}{X_{k+1} - X_{obs}} \\ \theta_{obs,g} &= \theta_{obs} - \theta_g \end{aligned}$$

[Step 5: Obstacle Detection] The obstacle avoidance strategy becomes active when the safety conditions d_{min} and θ_{min} are satisfied. This can be evaluated by the following conditions,

$$\begin{gathered} (d_{obs} < d_{min}) \quad AND \quad (\theta_{obs,g} < \theta_{min}) \\ OR \\ (d_{obs} < d_{min}) \quad AND \quad (\theta_{obs,g} < -\theta_{min}) \end{gathered}$$

[Step 6: Obstacle Avoidance] If Step 5 is being satisfied, the planned path should be updated with respect to the following conditions and formulas,

$$\begin{aligned} &\text{if} \quad \theta_{obs,g} > 0: \quad \beta(k+1) = \beta(k) + \Delta\beta(k) \\ &\text{if} \quad \theta_{obs,g} < 0: \quad \beta(k+1) = \beta(k) - \Delta\beta(k) \end{aligned}$$

It should be noted that in this step the desired angle update has been limited to increments or decrements of only 0.5 radians for avoiding sudden updates that could lead to an algorithmic instability or local minima.

[Step 7: Update Reference Path for MPC] The path update for the MPC controller is being calculated by,

$$\begin{aligned} x(k+1) &= x(k) + v\cos(\beta(k+1)) \\ y(k+1) &= y(k) + v\sin(\beta(k+1)) \end{aligned}$$

[Step 8: Algorithm Termination] In case that the vehicle has not reached the final goal, or the following equation is not true,

$$x(k+1) \in X_g \pm T \quad AND \quad y(k+1) > Y_g \pm T \tag{11}$$

with T the tolerance in the positioning accuracy, the algorithm returns to Step 2 and repeats, otherwise the goal has been achieved and the speed is set to $v = 0$.

The utilized MPC controller produces the rate of the articulated angle (control signal for the vehicle) based on the reference way points, generated from the path planner, the system model representation in equations (10) and the following cost function cost, which reflects the reference tracking error and the control effort. In this formulation $\tilde{x} = x - \hat{x}$ reflects the reference tracking error and $\tilde{u} = u - \hat{u}$ reflects the control action as,

$$\begin{aligned} J(\hat{x}(i+1), U_t) = \sum_{k=1}^{N} [&\tilde{x}^T(i+k|i) Q\, \tilde{x}(i+k|i) \\ &+ u^T(i+k|i) R\, u(i+k|i)] \end{aligned} \tag{12}$$

and Q, R are diagonal weighting matrices. During the calculation of the control action at the current time k, an optimal control actions sequence is being generated for the complete finite fixed horizon $[k, k+N]$, where N is the prediction horizon, while only the first prediction is being utilized as the control input and the calculations are then re–iterated at the next time instant, $[k+1, k+1+N]$.

IV. Simulation Results

For simulating the efficacy of the proposed path planner, the following articulated vehicle's characteristics have been considered: $l_1 = l_2 = 0.6m$, $w = 0.58m$, while the vehicle's speed is constant and equal to $1m$sec. Moreover, the constraints imposed on the articulated angle γ have been defined as ± 0.523 *rad* and random measurement Gaussian noise with a fixed variance was added to all measurements of the range sensor to simulate the real life measurement distortion. The parameters of the MPC have been tuned as: a) prediction horizon $M = 10$, b) control horizon $N = 5$, and c) a time sampling of 0.2 *sec*. The constraints imposed on the input and control action weighings matrices have been tuned according to the current active state space model as $Q = R = 0.3I_{3\times 3}$ for suspending aggressive maneuvers.

In Figure 5, a typical example of an articulated vehicle's motion behavior in an open loop configuration is being presented, where the safety and the sensing radius have been also indicated with blue and red color respectively.

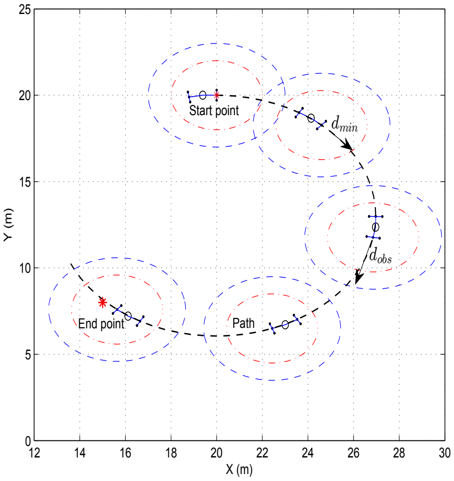


Fig. 5. Motion behavior of the articulated robot starting at $[20\ 20\ 0\ -10^o]$ with $v = 1m/s$, $\dot{\gamma} = 0$, $l_1 = l_2 = 0.61m$ $w = 0.58m$, $r_{obs} = 3m$, and $r_{min} = 2$.

From this movement it is straight forward that in the case that the vehicle's dynamics are being taken under consideration and due to the non–holonomic constraints, it is impossible for the vehicle to reach the final destination in a straight line as it would happen in the case that single point dynamics have been considered.

For evaluating the efficacy of the proposed online path planning and control scheme for an articulated vehicle two simulation scenarios have been considered with: a) random distribution of obstacles in an open arena and b) path planning within a multiple bounding surrounding environment, which mimics the operation in a mining environment or a construction site.

In Figure 6, the case of movement from the start point to two different goal points, in an arena with different shape obstacles, is being considered. In the examined case, the articulated vehicle initiates from $[0\ 0\ 45^o\ 7.5^o]$ and the multiple distinct goals have been set as [10, 35] and [20, 25], with $r_{obs} = 4m$ and $r_{min} = 3$. From the obtained results it is obvious that the vehicle is converging to the goal points, while avoiding the encountered obstacles. In the depicted closed loop paths, the articulated kinematics of the vehicle are obvious due to the existing curvature of the the vehicle during translation. In Figure 7 the corresponding responses of the error dynamics states as well as the articulated angle and the rate of the angle (control effort) are being depicted for both examined path tracking scenarios.

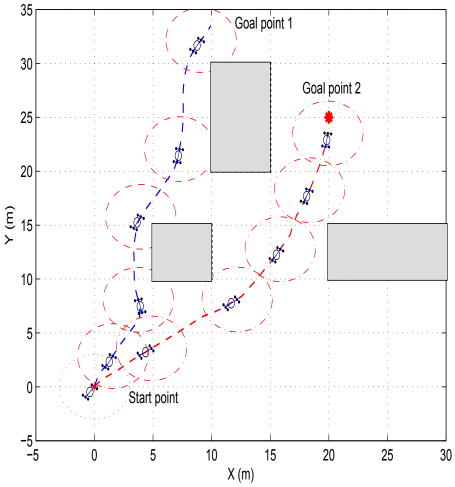


Fig. 6. Path tracking in an open arena with different shape obstacles. The vehicle starts at $[0\ 0\ 45^o\ 7.5^o]$ and the goal points are being located at [10 35] and [20 25] with $r_{obs} = 4m$ and $r_{min} = 3$.

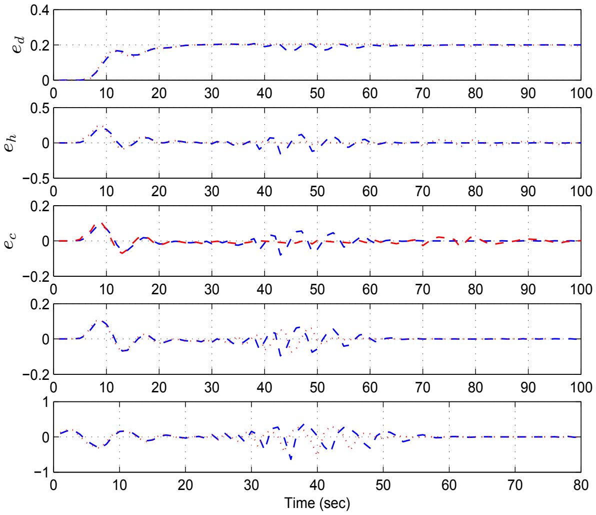


Fig. 7. Vehicle's error dynamics response and corresponding articulated angle γ and MPC's control effort $\dot{\gamma}$ for the trajectories presented in Figure 6 (blue-dashed lines for goal point 1 and red-dashed lines for goal point 2.

In the sequel the case of movement within a multiple bounding surrounding environment is being considered as it has been presented in Figure 8. In the examined case the articulated vehicle starts at $[0\ 0\ 15^o\ 7.5^o]$ with the goal points (1,2,3) at [10 15], $[10\ -15]$ and [40 10] respectively, with $r_{obs} = 4m$ and $r_{min} = 3$. As in the previously examined case, the proposed scheme manages to provides a smooth path

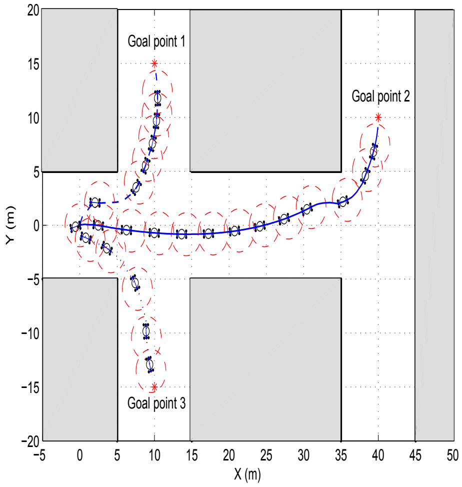


Fig. 8. The arena having same boundaries on both sides of the road, the robot starting at $[0\ 0\ 15^o\ 7.5^o]$ and the goal points (1,2,3) at $[10\ 15]$, $[10\ -15]$ and $[40\ 10]$ respectively with $r_{obs} = 4m$ and $r_{min} = 3$.

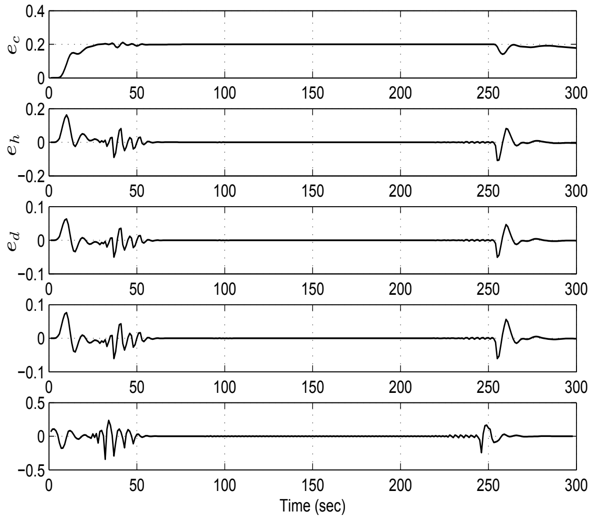


Fig. 9. The systems states error dynamics performance in meter, with respect the lines style of the three goal points in Figure(8).

generation and tracking for the vehicle. The existence of a non–holonomic behavior due to the articulated dynamics are again obvious due to the curvature of the obtained path. Finally, in Figure 9, the corresponding responses of the error dynamics states as well as the articulated angle and the rate of the angle (control effort) are being depicted for only the case of path generation and tracking to the second goal point.

V. Conclusions

In this article, a novel on–line path planning algorithm for an articulated vehicle has been proposed. The algorithm belongs to the bug like path planning algorithms and it is able to: a) consider the real articulated vehicle dynamics, b) avoid online encountered obstacles, and c) converge to the goal point based on a MPC scheme. For executing the algorithm it is assumed that the vehicle is aware of the starting and the goal point, while during translation partial sensory based information of the surrounding environment is being provided. Multiple simulation results have been presented that prove the efficiency of the suggested scheme.

References

[1] J. Larsson, "Unmanned operation of load-haul-dump vehicles in mining environments," Ph.D. dissertation, Norwegian University of Science and Technology, 2011.

[2] S. Scheding, G. Dissanayake, E. M. Nebot, and H. Durrant-Whyte, "An experiment in autonomous navigation of an underground mining vehicle," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 1, pp. 85–95, 1999.

[3] S. M. LaValle, "Motion planning," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 2, pp. 108–118, 2011.

[4] N. Nilsson, "A mobile automaton: An application of artificial intelligence techniques," DTIC Document, Tech. Rep., 1969.

[5] B. Chazelle, "Approximation and decomposition of shapes," *Advances in Robotics*, vol. 1, pp. 145–185, 1987.

[6] E. Guechi, J. Lauber, and M. Dambrine, "On-line moving-obstacle avoidance using piecewise bezier curves with unknown obstacle trajectory," in *Control and Automation, 2008 16th Mediterranean Conference on*. IEEE, 2008, pp. 505–510.

[7] K. Usher, "Obstacle avoidance for a non-holonomic vehicle using occupancy grids," in *2006 Australasian Conference on Robotics and Automation*, 2006.

[8] S. Ge and Y. Cui, "New potential functions for mobile robot path planning," *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 5, pp. 615–620, 2000.

[9] I. Škrjanc and G. Klančar, "Optimal cooperative collision avoidance between multiple robots based on bernstein–bézier curves," *Robotics and Autonomous systems*, vol. 58, no. 1, pp. 1–9, 2010.

[10] J. Ng and T. Bräunl, "Performance comparison of bug navigation algorithms," *Journal of Intelligent & Robotic Systems*, vol. 50, no. 1, pp. 73–84, 2007.

[11] V. Lumelsky and A. Stepanov, "Dynamic path planning for a mobile automaton with limited information on the environment," *Automatic Control, IEEE Transactions on*, vol. 31, no. 11, pp. 1058–1063, 1986.

[12] I. Kamon, E. Rimon, and E. Rivlin, "Tangentbug: A range-sensor-based navigation algorithm," *The International Journal of Robotics Research*, vol. 17, no. 9, pp. 934–953, 1998.

[13] I. Kamon and E. Rivlin, "Sensory-based motion planning with global proofs," *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 6, pp. 814–822, 1997.

[14] V. Lumelsky and T. Skewis, "Incorporating range sensing in the robot navigation function," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 20, no. 5, pp. 1058–1069, 1990.

[15] N. Buniyamin, W. Wan Ngah, N. Sariff, and Z. Mohamad, "A simple local path planning algorithm for autonomous mobile robots," *International journal of systems applications, Engineering & development*, vol. 5, no. 2, pp. 151–159, 2011.

[16] K. Kanjanawanishkul and A. Zell, "Path following for an omnidirectional mobile robot based on model predictive control," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 3341–3346.

[17] A. Ismail, A. Sheta, and M. Al-Weshah, "A mobile robot path planning using genetic algorithm in static environment," *Journal of Computer Science*, vol. 4, no. 4, pp. 341–344, 2008.

[18] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robotics and Autonomous Systems*, vol. 55, no. 6, pp. 460–469, 2007.

[19] P. Ridley and P. Corke, "Load haul dump vehicle kinematics and control," *Journal of dynamic systems, measurement, and control*, vol. 125, p. 54, 2003.

[20] T. Nayl, G. Nikolakopouls, and T. Gustafsson, "Switching model predictive control for an articulated vehicle under varying slip angle," in *Control and Automation, 2012 20th Mediterranean Conference on*. IEEE, 2012, pp. 884–889.