Real-time Optimal Control of Multi-wheeled Combat Vehicles – using Artificial Neural Network and Potential Fields

A. Mohamed[1*], M. EI-Gindy[1], X. Huang[2], J. Ren[1] and H. Lang[1]

[1] University of Ontario Institute of Technology, Oshawa, Canada
[2] Istuary Innovation Group, Markham, Canada

Abstract:

This paper presents a real‑time path planning algorithm for autonomous multi‑wheeled combat vehicles using Artificial Neural Network (ANN), Artificial Potential Fields (APFs) and optimal control theory. Real‑time navigation of autonomous vehicles is a very complex problem and it is crucial for many military operations. This paper proposes an optimal control and ANN approach for a dynamic model of the multi‑wheeled combat vehicle to generate the possible optimal paths that cover every part of the workspace. Consequently, the obtained paths are used to train the proposed ANN model. The trained ANN has the capability to control the movement of combat vehicle in real time from any starting point to the desired goal position within the area of interest. The vehicle path is autonomously generated from the previous vehicle location parameter in terms of lateral velocity, heading angle and yaw rate of the vehicle. APF is proposed for preventing the vehicle from colliding with obstacles in border destination. The effectiveness and efficiency of the proposed approach are demonstrated in the simulation results, which show that the proposed ANN model is capable of navigating the multi‑wheeled combat vehicle in real time.

Keywords:

real‑time, optimal control theory, path planning, autonomous multi‑wheeled combat vehicle, Artificial Neural Network, artificial potential fields.

---

* *Corresponding author: Faculty of Engineering and Applied Science, University of Ontario Institute of Technology (UOIT), 2000 Simcoe Street North, Oshawa, ON L1H 7K4, Canada. E-mail: Amr.Mohamed@uoit.net*

1. Introduction

Nowadays, autonomous multi-wheeled combat vehicles are commonly utilized in military applications and are of great relevance. Building and controlling such vehicles started to draw increasing interests while remaining one of the boosting research topics in both mobile robotics and automotive engineering [1, 2]. Such vehicles are considered one of the most difficult types of the vehicles in applying autonomy due to large dimensions, heavy weight, the complex geometry and dynamics.

On the other hand, the expected outcomes of the autonomous combat vehicles are set to increase the overall performance on the battlefield and keep soldier safety. Autonomous multi-wheeled combat vehicles offer numerous advantages on modern battlefields as unmanned companions to ground armed forces. These kinds of autonomous vehicles can operate day and night for dangerous mission compared with manned ground vehicles. Research efforts have been continuously devoted to improving the performance of combat vehicles to be navigated autonomously. For example, [3] discussed the unmanned ground combat vehicle (Black Knight's) with various control modes which include path tracking, guarded teleoperation, and fully autonomous movement. These vehicles have the ability to switch between path tracking via GPS waypoint, remote teleoperation, autonomous navigation, which contained all sensors required for perception such as cameras and LIDAR. [4] developed a multi-wheeled combat vehicle-bicycle model to obtain the reference yaw rate and lateral acceleration for the designed controller. Two PID controllers are developed, the upper controller to corrective yaw moment which is the input to the lower controller to manage torque distribution among the driving wheel.

Path planning is considered one of the enormous challenge in designing autonomous vehicles in order to navigate autonomously. Planning of the optimal paths is still a critical issue in an autonomous vehicles navigation system, especially for multi-wheeled combat vehicles. Autonomous combat vehicles need to find a feasible, optimal path from the start to the target point. Many approaches have been proposed to deal with the path planning problem for autonomous vehicles [5-7]. For example, [8] obtained an optimal path for UAV in 3-D environments based on the combination of flight space partitioning, Dijkstra algorithm, and the potential field method. [9] developed a self-learning rapidly-exploring random tree for a wheeled mobile robot to improve the planning efficiency by taking advantage of the prior knowledge accumulation and cost estimation.

Real-time path planning for autonomous using ANN has many distinct advantages. ANNs have previously been applied in autonomous driving [10-12]. For example, [13] used Bioinspired neural network to perform Real-time path planning for autonomous underwater vehicle in a three-dimensional (3D) unknown environment. Recently, [14] developed a convolutional neural network to learn planning from imitation for path planning in complex traffic situations from both simulated and real-world data. In [15], the author introduced a neural network model via integer linear program formulation for obtaining the shortest path with the optimal feature. [16-19] were also proposed for solving shortest path problems.

The main objective of this work is to develop a real-time path planning framework for autonomous multi-wheeled combat vehicle based on the artificial neural network. The proposed architecture enables the vehicle to navigate autonomously from the initial location to the destination location in real time while avoiding the target location border. First, the vehicle optimal paths between two points were generated

using the developed optimal path planning algorithm. Various optimal paths were obtained using different starting points. Then, the obtained optimal paths are used for training the proposed ANN model considering the vehicle parameters in terms of yaw rate, lateral velocity, heading angle and steering angle. The developed ANN is a multi-layer network, which consists of three layers; an input, a hidden, and an output layers. The network is trained in offline mode using back-propagation supervised learning algorithm. The main contribution of this paper is to combine neural network-based approach with optimal path planning in real time for an autonomous multi-wheeled combat vehicle. The simulation results demonstrated that the proposed ANN model has the capability to provide the optimal solutions by solving the vehicle path planning problem from different starting points to the target position.

The paper is structured as follows. The multi-wheeled combat vehicle dynamics model is presented in section 2. Section 3 describes obstacle modelling for the goal border. Section 4 describes motion planning algorithm. Section 5 introduces the proposed artificial neural network model. The simulation results obtained will be presented in section 6. The conclusion is given in section 7.

2. Dynamics Model of the Multi-Wheeled Combat Vehicle

A simplified two-degree-of-freedom bicycle model is derived to represent the vehicle motion and generate the optimal path. The simplified model describes the lateral and yaw motion of multi-wheeled combat vehicle with neglect of roll dynamics is shown in Fig. 1. In this model the left and right sides of the vehicle have been combined and the centre of gravity (CG) height is assumed zero. Vehicle forward speed U, lateral speed V, and yaw rate r, are shown acting at the CG. Tires slip angles and wheels steer angles are shown in blue and red respectively. The actual vehicle configuration and the simulated full vehicle model using TruckSim are shown in Fig. 2. The vehicle is equipped with four axles and the front two axles are steerable.
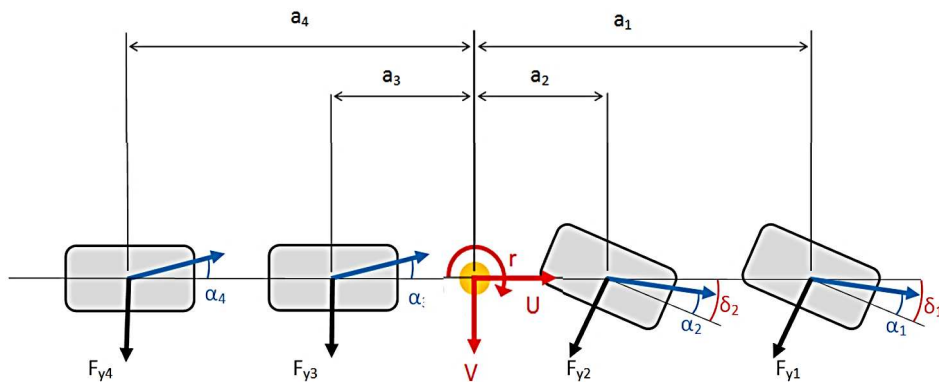

Fig. 1 Two-degree-of-freedom Bicycle Vehicle Model

In the simplified model, Fig. 1, the forward velocity U is assumed constant and $\delta1$, $\delta2$ represents the average steering angles of the first and second axle's wheels, respectively. Also, the steering angles are assumed small, i.e. $\cos \delta = 1$. The governing equations of motion, for the vehicle body lateral and yaw dynamics, are given in Eqs (1), (2) below, respectively:

Fig. 2 (left) Actual vehicle configuration [25] *and (right) the simulation model*

$$m\left(\dot{V} + rU\right) = F_{y1} + F_{y2} + F_{y3} + F_{y4}\,, \tag{1}$$

$$I_{zz}\dot{r} = a_1 F_{y1} + a_2 F_{y2} - a_3 F_{y3} - a_4 F_{y4}\,. \tag{2}$$

Tire cornering force and slip angle equations for each axle are as follows, where $i = 1,\dots, 4$ and $j = 1, 2$

$$F_{yi} = \alpha_i C_{\alpha i}\,, \tag{3}$$

$$\alpha_j = \delta_j - \left(\frac{V + a_j r}{U}\right), \tag{4}$$

where $C_{\alpha i}$ represents the axle cornering stiffness including the left and the right tire.

The lateral sideslip acceleration and the yaw acceleration can be derived as follows:

Lateral Motion Equation

$$\dot{V} = -\frac{C_{\alpha 1} + C_{\alpha 2} + C_{\alpha 3} + C_{\alpha 4}}{mU} V +$$
$$+ \left(\frac{-a_1 C_{\alpha 1} - a_2 C_{\alpha 2} + a_3 C_{\alpha 3} + a_4 C_{\alpha 4}}{mU} - U\right) r + \frac{C_{\alpha 1}\delta_1 + C_{\alpha 2}\delta_2}{m}\,, \tag{5}$$

Yaw Motion Equation

$$\dot{r} = \frac{-a_1 C_{\alpha 1} - a_2 C_{\alpha 2} + a_3 C_{\alpha 3} + a_4 C_{\alpha 4}}{I_{zz}U} V +$$
$$+ \frac{-a_1^2 C_{\alpha 1} - a_2^2 C_{\alpha 2} + a_3^2 C_{\alpha 3} + a_4^2 C_{\alpha 4}}{I_{zz}U} r + \frac{a_1 C_{\alpha 1}\delta_1 + a_2 C_{\alpha 2}\delta_2}{I_{zz}}\,, \tag{6}$$

where
U, V represents the longitudinal and lateral velocities of vehicle centre of gravity respectively;
m is the vehicle mass;
I_{zz} is the yaw moment of inertia;
r, Ψ are the vehicle yaw rate and yaw angle, respectively;
$C_{\alpha 1}$, $C_{\alpha 2}$, $C_{\alpha 3}$, $C_{\alpha 4}$ denote the cornering stiffness's of tires.

The coordinates of the vehicle body motion (x, y) in global coordinates can be derived as following equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} U & -(V + a_1 r) \\ (V + a_1 r) & U \end{bmatrix} \begin{bmatrix} \cos \Psi \\ \sin \Psi \end{bmatrix},$$ (7)

Setting the state vector as $X = [V\ r\ \Psi\ x\ y\ \delta_f\ ]^T$ and using Eqs (5)–(7), the state space equations for the bicycle model can be written in the following vector form:

$$\dot{X} = f(X) + \mathbf{B}u,$$ (8)

where

$$f(X) = \begin{bmatrix} -a_{11}V + a_{12}r + b_{11}\delta_f \\ a_{21}V + a_{22}r + b_{21}\delta_f \\ r \\ U\cos\Psi - (V + a_1 r)\sin\Psi \\ U\sin\Psi + (V + a_1 r)\cos\Psi \\ 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad u = \dot{\delta}_f$$

where

$$a_{11} = \frac{C_{\alpha 1} + C_{\alpha 2} + C_{\alpha 3} + C_{\alpha 4}}{mU},$$

$$a_{12} = \frac{-a_1 C_{\alpha 1} - a_2 C_{\alpha 2} + a_3 C_{\alpha 3} + a_4 C_{\alpha 4}}{mU} - U,$$

$$a_{21} = \frac{-a_1 C_{\alpha 1} - a_2 C_{\alpha 2} + a_3 C_{\alpha 3} + a_4 C_{\alpha 4}}{I_{zz}U},$$

$$a_{22} = \frac{-a_1^2 C_{\alpha 1} - a_2^2 C_{\alpha 2} + a_3^2 C_{\alpha 3} + a_4^2 C_{\alpha 4}}{I_{zz}U},$$

$$b_{11} = \frac{C_{\alpha 1} + C_{\alpha 2}Ks}{m},$$

$$b_{21} = \frac{a_1 C_{\alpha 1} + a_2 C_{\alpha 2}Ks}{I_{zz}}.$$

3. Obstacle Modelling Based Artificial Potential Field

In this section, the obstacles are modeled based on the artificial potential field using the sigmoid function. APF is employed to define the destination location borders such as parking spot border. The principle of this approach is specified in using imaginary forces acting on the vehicle direction. Obstacles have repulsive fields around them and the goals have attractive fields. The synthesis of all the forces determines the moving direction of the vehicle in the field [21]. The implementation of the artificial potential field in the vehicle model requires the calculation of the vehicle body, surrounding obstacles and target location coordinates.

If we associate a vector $p = \{x, y\}$ with each point in space, the potential field is based on sigmoid function which can be described as following:

$$f(\boldsymbol{p}) = \sum_{i=0}^{M} \prod_{j=0}^{N} f_{\text{sig}} \left[\varphi_{ij}(\boldsymbol{p}) \right], \tag{9}$$

$$f_{\text{sig}}(s) = \frac{1}{1 + e^{-\gamma f(s)}}, \tag{10}$$

where

$\boldsymbol{p}$ is the position vector of a point in space;

$s = \varphi(\boldsymbol{p})$ is a surface function representing the distance from the object surface and the value of γ is application dependent.

Consequently, using large γ due to the navigating through narrow paths is suitable to highly localize the effects of obstacles at the target location and avoid collision. The APF forces of the obstacles at the destination location are obtained as shown in Fig. 3. The represented obstacles as a goal border are defined by the following three points $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, and $P_3(x_3, y_3)$, where x_i, y_i represent the coordinate of the obstacle. Then, the potential field at the goal location is defined as a convolution function.

$$f_{\text{O}}(x, y) = f_{\text{sig}} \left[\varphi_1(x, y) \right] * f_{\text{sig}} \left[\varphi_2(x, y) \right] * f_{\text{sig}} \left[\varphi_3(x, y) \right], \tag{11}$$

The visualization of the generated potential field using Eq. (11) is shown in Fig. 3. This figure shows 3D repulsor potential field representing the goal location border.

4. Motion Planning Algorithm

The proposed motion planning algorithm is based on integration between artificial potential filed and optimal control theory. The optimal trajectory for the autonomous multi-wheeled combat vehicle is obtained via minimizing the defined cost function in Eq. (12) to enable the multi-wheeled combat vehicle to navigate autonomously along the generated path from its current position to the goal location. The Pontryagin's Principle [22-24] is introduced to solve the optimal control problem for generating the optimal path.

$$\min u(t) J = \int_{t_o}^{t_f} L \left[X(t), u(t), t \right] \mathrm{d}t, \tag{12}$$

where

$X(t)$ is the states vector $X = [V \, r \, \Psi \, x \, y \, \delta_f]^{\text{T}}$;

$u(t)$ represent the control action for front wheel steering angle.

The Lagrange function L(x,u) is introduced to minimise the generated path according to the potential field's coordinates of the obstacle as follows:

$$\min u(t) J = X^T Q X + u^T R u + W_{po} \sum_{i} P_{oi}(X_c), \tag{13}$$

where

X, X^{T} are the state vector and its transpose respectively,

u is the control input,
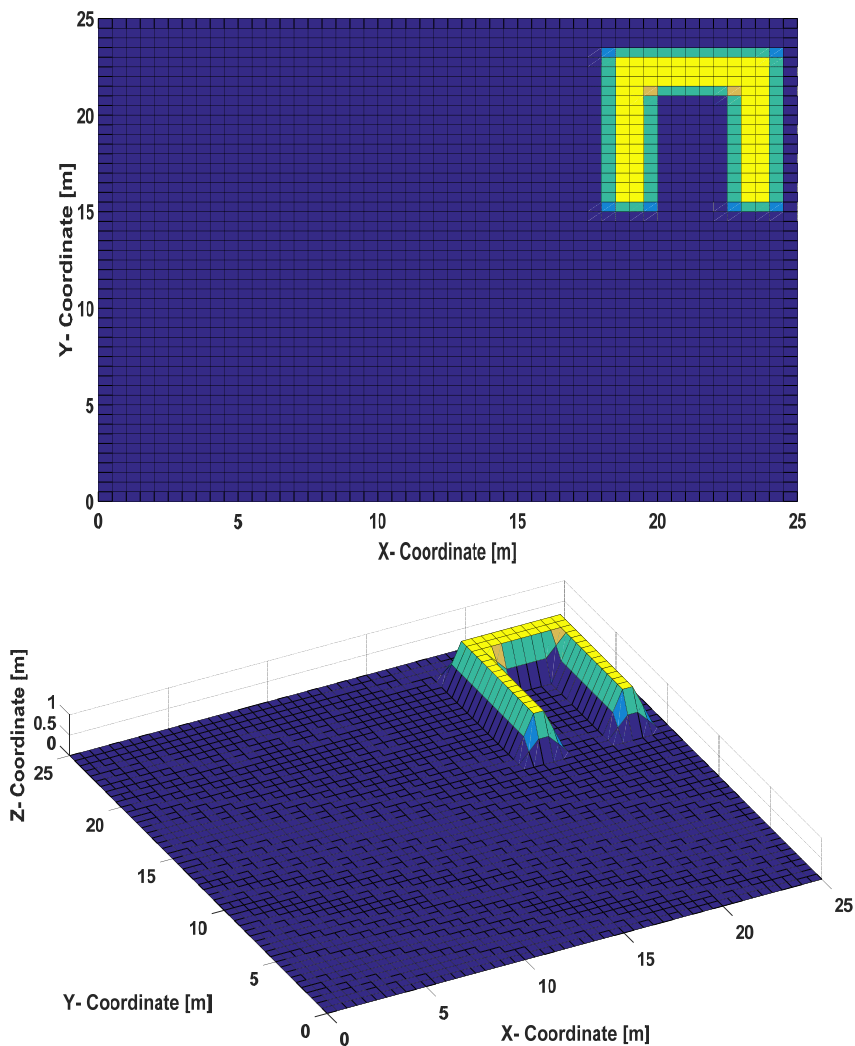
(Q, R) are positive weighting matrices.

Fig. 3. Visualization of the workspace using the artificial potential field in 2D & 3D based on sigmoid function.

Consequently, by considering the initial and final condition of the vehicle states as follows:

$$\left. \begin{array}{l} X\left(t_f\right) = \begin{bmatrix} 0 & 0 & \Psi_0 & d_{xo} & d_{yo} & 0 \end{bmatrix} \\ X\left(t_o\right) = \begin{bmatrix} 0 & 0 & \Psi_f & d_{xf} & d_{yf} & 0 \end{bmatrix} \end{array} \right\},$$

(14)

where
(d_{x0}, d_{y0}), (d_{yf}, d_{yf}) are the initial vehicle positions in longitudinal and lateral directions respectively;
(Ψ_0, Ψ_f) are the initial and final heading angles of the vehicle.

Based on the MATLAB Software, the optimal problem was solved and the state variables within time interval from t_0 to t_f to and consequently were calculated, the optimal path for the vehicle was generated. The obtained optimal paths will be used in the next section for training the developed ANN.

5. Neural Network Model

The proposed ANN is a multilayer perceptron (MLP), consists of three layers, an input layer, a hidden layer, and an output layer. The input layer has five input source nodes which are physically fed by the multi-wheeled combat vehicle parameters (position in x, y and vehicle heading angle θ, vehicle yaw rate r and lateral velocity V). The hidden layer has 20 neurons which receive input data from the input layer and multiply them by the weights denoted by W_{ij} and bias then forwards the resulted values to the output layer. The output layer has one neuron that directly represents the front wheel steering angle which controls the vehicle path.

W_{ij} represent the weights values the interconnection between the different neurons of the network. Additionally, two biases are employed in the hidden and the output layers to regulate and limit the output of the network.

Formally, the proposed neural network shown in Fig. 4 can be defined as follows: ANN = {I, O, T, W} where I represents the set of input nodes, O represents the output node, T represents the topology of the network including the number of layers and the number of their neurons, W represents the set of weights values.

$I = \{V, r, \theta, x, y\}$
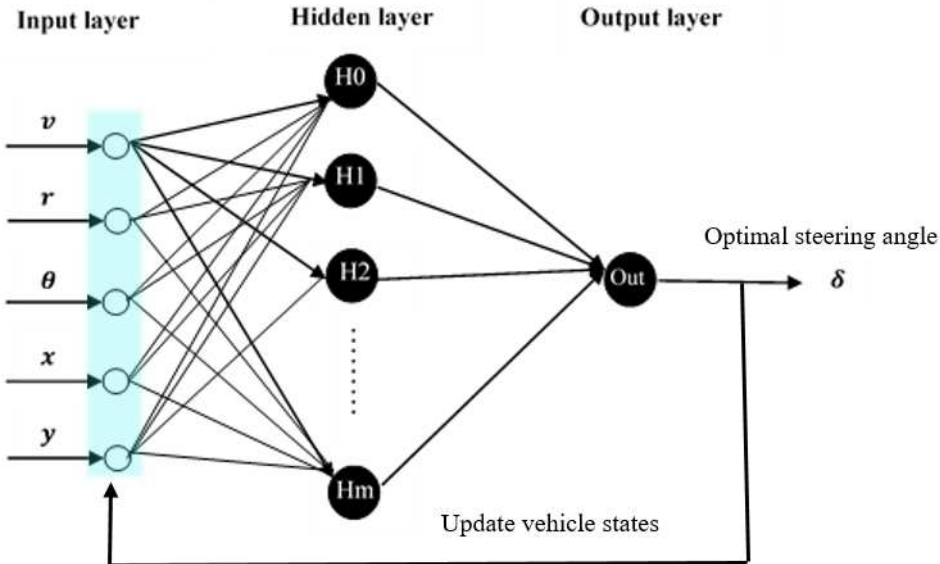$T = \{Lin\text{-}5, Lh0\text{-}20, Lout\text{-}1\}$

Fig. 4. Architecture of the proposed ANN

5.1. The Proposed Approach

The proposed neural network model shown in Fig. 4 is trained through a supervised learning approach using the back-propagation algorithm [25]. The obtained optimal paths from the MATLAB model are used for the training purpose. Various optimal paths are obtained for the training purpose using different starting points. This ANN model is an iterative type as follows; first, it will be initialized by using the initial input data of the vehicle states. Then the corresponding steering wheel angle is calculated. The obtained steering wheel angle is used to update the vehicle states. The updated states are used to update the ANN model in each iteration to estimates the new steering wheel angle of the moving vehicle for the next navigation step. The developed motion planning algorithm can be summarized as follows:

1. Specify the start configurations for the multi-wheeled combat vehicle (V_i, r_i, θ_i x_i, y_i).
2. Initialize the artificial neural network model using the initial input data which returns the front steering wheel angle of the next motion step.
3. Use the obtained steering wheel angle from NN model to update the vehicle parameters, then input the updated parameter again to the NN model to return the new front steering wheel angle.
4. Generate the optimal path for the multi-wheeled combat vehicle by repeating step 3 till it reaches the target position.

The proposed artificial neural network model was implemented using OpenNN library. OpenNN is an open source class library written in C++ programming language which implements neural networks, the main area of machine learning research. The main advantage of OpenNN is its high performance. It is developed in C++ for better memory management and higher processing speed. The NN was compiled under the MS Visual Studio. It incorporates a training engine that can be fed with various static and dynamic parameters including initial weights, input data, learning rate, and the number of epochs to execute.

6. Simulation and Result

In this section, the developed motion planning algorithm using MATLAB has been tested for generating the optimal path of the vehicle from different starting points. Fig. 5 shows the generated optimal path which validates the ability of the proposed methodology to generate an optimal path for the vehicle from a given starting point and destination point avoiding hitting the parking borders. Then, various optimal paths are generated using different starting points which are used to training the ANN model.

The developed ANN is tested and validated by generating the vehicle optimal path through the following three scenarios. The first scenario shows the vehicle optimal path between two points using ANN compared with the training optimal path for testing. Then it is validated using a different point in between the training data. The second scenario examines the vehicles optimal path from different starting points compared with the generated optimal path for the same condition using the Matlab code. The third scenario shows the effects of the vehicle longitudinal speed on the obtained paths.
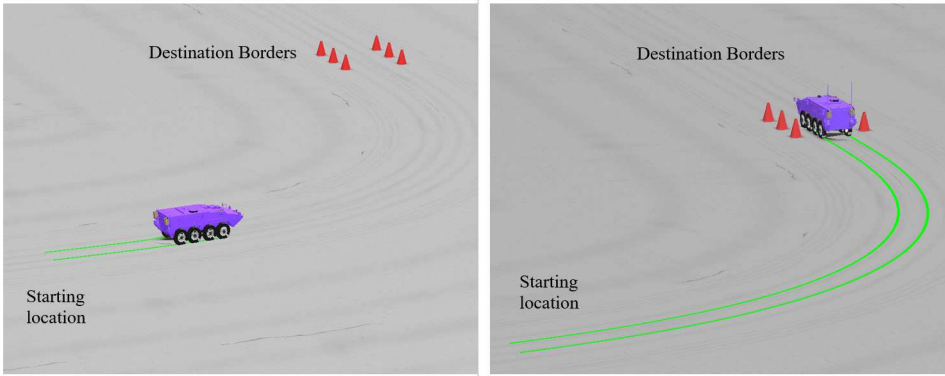
*Fig. 5. Simulation for the obtained optimal path between two points
with destination border*

6.1. First Scenario: NN Optimal Path Planning (Testing)

The main objective of this scenario is to check the capability of the proposed ANN model to provide the optimal path for the vehicle from a given starting point to the destination point safely without hitting the destination border. Fig. 6 shows the obtained optimal path from both the optimal control theory (training data) represented in blue and the NN model represented in dashed red for a starting point (0.14). The results showed that a proposed ANN model was able to generate the optimal path in real time for the vehicle to reach its goal location through the safe and correct trajectory compared with the original optimal path without hitting the border.

Additionally, the time-history of the vehicle parameter in terms of lateral velocity, heading and steering angles and yaw rate during this manoeuvre is shown in Fig. 7 compared with the optimal control theory model.
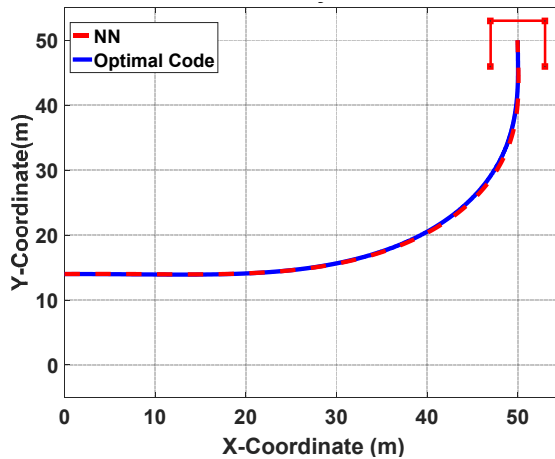

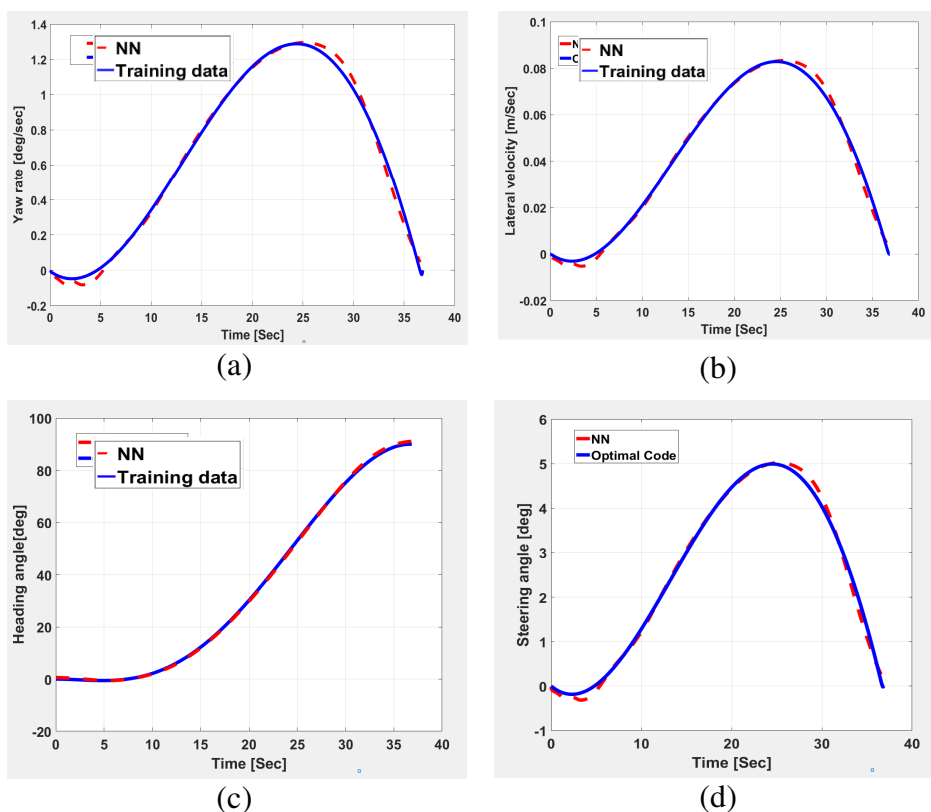*Fig. 6. Obtained path for both NN model and optimal control for the same
starting point*

Fig. 7. Time-history comparison a) Yaw Rate [deg·s^{-1}], *b) Lateral Velocity* [m·s^{-1}],
c) Heading Angle [deg], *d) Steering Angle* [deg].

The ANN is validated with the training data by generating various optimal paths in between the training data as shown in Fig. 8. It has been noticed that the introduced ANN model has the capability to generate the vehicle optimal paths that behave in the same way as the training paths even choosing new starting points.

6.2. Second Scenario: Optimal Path Planning

In this scenario, the capability of the proposed ANN model is examined to navigate the vehicle using various starting point in the workspace as shown in Fig. 9. The obtained result shows that the vehicle is able to autonomously navigate from any point in the workspace to reach the goal point following the optimal path in red line compared with optimal control theory result (training data) in the blue line.

6.3. Optimal Control and NN Path Planning at Different Speed

This scenario is aimed to show the effect of the vehicle longitudinal speed on the obtained paths. For this purpose, the vehicle optimal paths using the MATLAB model are generated at speeds 2, 4, 8, and 10 m·s^{-1} to show the speed effect on the generated paths. Then, compare the obtained results with the NN path at speed 2 m·s^{-1} as shown

in Fig. 10. The obtained result shows that there is not much deviation in the generated optimal paths by increasing the speed. It has been noticed that the obtained path using ANN model is very close to the generated paths from the optimal path algorithm at different speeds.
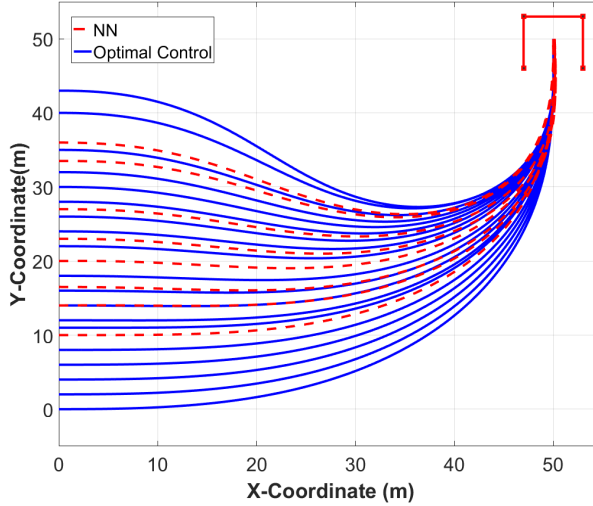

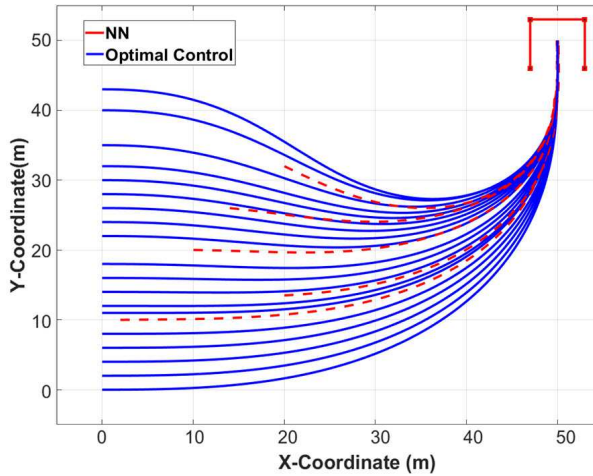Fig. 8. Optimal Path between Two Points using ANN and optimal control

Fig. 9. Generating Optimal Path from different starting point in the space using ANN and optimal control (trained data)

7. Conclusion

This paper presented a motion planning algorithm for the Multi-wheeled combat vehicle in real time based on optimal control theory and artificial potential filed. First, the optimal paths were obtained for the vehicle from different starting points. The generated paths based on optimal control theory will be used for training the proposed ANN
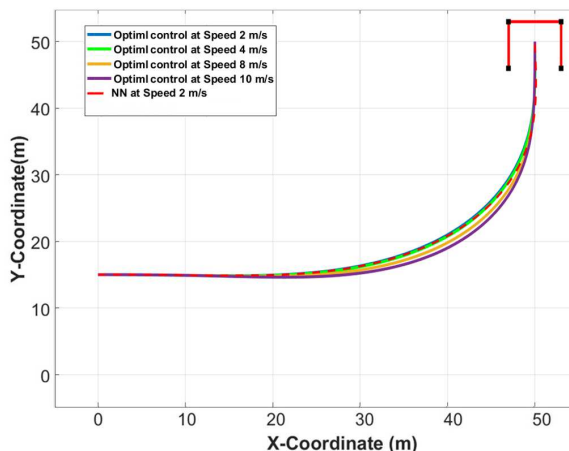
*Fig. 10. Generating Optimal Path at different longitudinal speed ANN
and optimal control*

model. The ANN is a multilayer network and can be trained offline using back-propagation learning algorithm. After the training, the developed ANN model is able to provide the optimal steering angle in real time to navigate the vehicle autonomously between two points. The proposed ANN model is a significant improvement in the field of autonomous combat vehicle navigation and control because it provided an optimal and real-time solution for this application.

Various numerical simulations are performed from different starting points to validate the performance of the introduced algorithm. For each simulation, the results demonstrate that the proposed neural network method is able to optimally generate the required path in real time to reach the goal location while maintaining the vehicle parameters, including lateral velocity, heading and steering angles, and yaw rate, identical with the vehicle parameters obtained from the data used for training.

Acknowledgement

References

[1] FISHER, A. Inside Google's Quest to Popularize Self-Driving Cars. *Popular Science*, 2013, vol. 18, p. 2013-09.

[2] KESSLER, A.M. Elon Musk Says Self-driving Tesla Cars Will Be in the US by Summer. *The New York Times*, 2015-03-19.

[3] VALOIS, J., HERMAN, H., BARES, J. and RICE, D.P. Remote Operation of the Black Knight Unmanned Ground Combat Vehicle. In *Proceedings of SPIE – Unmanned Systems Technology X*, 2008. ISBN 978-0-8194-7153-6.

[4] RAGHEB, H., EL-GINDY, M. and KISHAWY, H.A. On the Multi-wheeled off-road Vehicle Performance and Control. *International Journal of Vehicle Systems Modelling and Testing*, 2013, vol. 8, no. 3, p. 260-281. ISSN 1745-6436.

[5]   FAROUKI, R.T., GIANNELLI, C., MUGNAINI, D. and SESTINI, A. Path Planning with Pythagorean-hodograph Curves for Unmanned or Autonomous Vehicles. In *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, London: Sage, 2017. DOI 10.1177/09544100 17690550.

[6]   PERUMAL, D.G. et al.. MILP Based Autonomous Vehicle Path-planning Controller for Unknown Environments with Dynamic Obstacles. *International Journal of Heavy Vehicle Systems*, 2016, vol. 23, no. 4, p. 350-369. DOI 10.1504/ IJHVS.2016.079272.

[7]   QU, Y., ZHANG, Y. and ZHANG, Y. Optimal Flight Path Planning for UAVs in 3-D Threat Environment. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, Orlando: IEEE, p. 149-155. DOI 10.1109/ ICUAS.2014.6842250.

[8]   YANG, S., WANG, Z. and ZHANG, H., Kinematic Model-based Real-time Path Planning Method with Guide Line for Autonomous Vehicle. In *36th Chinese Control Conference (CCC)*, Dalian: IEEE, 2017, p. 990-994. DOI 10.23919/Chi CC.2017.8027474.

[9]   ZHANG, X., LÜTTEKE, F., ZIEGLER, C. and FRANKIE, J. Self-learning RRT Algorithm for Mobile Robot Motion Planning in Complex Environments. In MENEGATTI, E., MICHAEL, N., BERNS, K. and YAMAGUCHI, H. (eds) *Intelligent Autonomous Systems 13. Advances in Intelligent Systems and Computing*, 2016, vol. 302, Cham: Springer, p. 57-69. DOI 10.1007/978-3-319-08338-4_5.

[10] ZHANG, Y., LI, S. and GUO, H. A Type of Biased Consensus-based Distributed Neural Network for Path Planning. *Nonlinear Dynamics*, 2017, vol. 89, no. 3. p. 1803-1815. DOI 10.1007/s11071-017-3553-7.

[11] KODGULE, S.A., DAS, A. and SELVAKUMAR, A.D.A. A Comparative Study between Hopfield Neural Network and A* Path Planning Algorithms for Mobile Robot. In DASH, S., VIJAYAKUMAR, K., PANIGRAHI, B. and DAS, S. (eds) *Artificial Intelligence and Evolutionary Computations in Engineering Systems. Advances in Intelligent Systems and Computing*, vol. 517, Singapore: Springer, 2017, p. 33-48, DOI 10.1007/978-981-10-3174-8_4.

[12] LUO, C., ZHANG, J., JAN, G.E. and MO, H. A New Hopfield-type Neural Network Approach to Multi-goal Vehicle Navigation in Unknown Environments. In *2016 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Beijing: IEEE, 2016, p. 1-6. DOI 10.1109/ICVES.2016.7548165.

[13] NI, J., WU, L., SHI, P. and YANG, S.X. A Dynamic Bioinspired Neural Network Based Real-Time Path Planning Method for Autonomous Underwater Vehicles. *Computational Intelligence and Neuroscience*, 2017. DOI 10.1155/2017/9269742.

[14] REHDER, E., QUEHL, J. and STILLER, C. Driving like a Human: Imitation Learning for Path Planning Using Convolutional Neural Networks. In *International Conference on Robotics and Automation Workshops*, 2017.

[15] NAZEMI, A. and OMIDI, F. An Efficient Dynamic Model for Solving the Shortest Path Problem. *Transportation Research Part C: Emerging Technologies*, 2013, vol. 26, p. 1-19. DOI 10.1016/j.trc.2012.07.005.

[16] ZHANG, Y., WU, L., WEI, G. and WANG, S. A Novel Algorithm for all Pairs Shortest Path Problem Based on Matrix Multiplication and Pulse Coupled Neural Network. *Digital Signal Processing*, 2011, vol. 21, no. 4, p. 517-521. DOI 10. 1016/j.dsp.2011.02.004.

[17] SANG, Y., LV, J., QU, H. and YI, Z. Shortest Path Computation Using Pulse-coupled Neural Networks with Restricted Autowave. *Knowledge-Based Systems*, 2016, vol. 114, p. 1-11. DOI 10.1016/j.knosys.2016.08.027.

[18] LI, X., MA, Y. and FENG, X. Self-adaptive Autowave Pulse-coupled Neural Network for Shortest-path Problem. *Neurocomputing*, 2013, vol. 115, p. 63-71. DOI 10.1016/j.neucom.2012.12.030.

[19] QU, H., YI, Z. and YANG, S.X. Efficient Shortest-path-tree Computation in Network Routing Based on Pulse-coupled Neural Networks. *IEEE Transactions on Cybernetics*, 2013, vol. 43, no. 3, p. 995-1010. DOI 10.1109/TSMCB. 2012.2221695.

[20] *Wikipedia: Stryker* [on line]. [cited 2013-02-10]. Available from: <http://en. wikipedia.org/wiki/Stryker>.

[21] REN, J. et al. A Potential Field Model Using Generalized Sigmoid Functions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2007, vol. 37, no. 2. p. 477-484. DOI 10.1109/TSMCB.2006.883866.

[22] KIEN, B.T., RÖSCH, A. and WACHSMUTH, D. Pontryagin's Principle for Optimal Control Problem Governed by 3D Navier–Stokes Equations. *Journal of Optimization Theory and Applications*, 2017, vol. 173, no. 1, p. 30-55. DOI 10. 1007/s10957-017-1081-8.

[23] ONORI, S., SERRAO, L. and RIZZONI, G., Pontryagin's Minimum Principle. In ONORI, S., SERRAO, L. and RIZZONI, G. (eds) *Hybrid Electric Vehicles – Energy Management Strategies.* London: Springer, 2016, p. 51-63. DOI 10.1007/978-1-4471-6781-5_5.

[24] MOHAMED, A., REN, J., LANG, H. and EL-GINDY, M. Optimal Collision-free Path Planning for an Autonomous Articulated Vehicle with Two Trailers. In *International Conference on Industrial Technology (ICIT)*, Toronto: IEEE, 2017, p. 860-865. DOI 10.1109/ICIT.2017.7915472.

[25] KAUR, T. Implementation of Back Propagation Algorithm: A Neural Network Approach for Pattern Recognition. *International Journal of Engineering Research and Development*, 2012, vol. 1, no. 5, p. 30-37. ISSN 2278-067X.