

ERDCMP-20-1

Engineer Research and
Development Center



**US Army Corps
of Engineers®**
Engineer Research and
Development Center



Artificial Intelligence and Machine Learning for Autonomous Military Vehicles

Sergey Vecherin, Jacob Desmond, Taylor Hodgdon, Jordan Bates,
Michael Parker, James Lever, Garrett Hoch, Mark Bodie,
and Sally Shoop

August 2020

The U.S. Army Engineer Research and Development Center (ERDC) solves the nation's toughest engineering and environmental challenges. ERDC develops innovative solutions in civil and military engineering, geospatial sciences, water resources, and environmental sciences for the Army, the Department of Defense, civilian agencies, and our nation's public good. Find out more at www.erdclibrary.on.worldcat.org/discovery.

To search for other technical reports published by ERDC, visit the ERDC online library at <https://erdclibrary.on.worldcat.org/discovery>.

Artificial Intelligence and Machine Learning for Autonomous Military Vehicles

Sergey Vecherin, Taylor Hodgdon, Jordan Bates, Michael Parker,
James Lever, Garrett Hoch, and Sally Shoop

*Cold Regions Research Laboratory
U.S. Army Engineer Research and Development Center
72 Lyme Road
Hanover, NH 03755*

Mark Bodie

*Information Technology Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180*

Jacob Desmond

*Department of Mathematics
Purdue University
150 N University Street
West Lafayette, IN 47907*

Final report

Approved for public release; distribution is unlimited.

Prepared for Assistant Secretary for Army Acquisitions, Logistics & Technology
Washington, DC 20314

Under Project 471941, "Remote assessment of Snow Mechanical Properties" and
"Mobility in Peat and Northern Soils"

Preface

This study was conducted for the Assistant Secretary of the Army, Acquisition, Logistics and Technology under Program Element 62784, Project Number AT40, and Task Number 48. The technical monitor was John Rushing.

The work was performed by the Terrestrial and Cryospheric Sciences Branch and by the Force Projection and Sustainment Branch of the Research and Engineering Division, U.S. Army Engineer Research and Development Center, Cold Regions Research and Engineering Laboratory (ERDC-CRREL) and the Computational Analysis Branch of the Science and Engineering Division, Information Technology Laboratory (ERDC-ITL).

At the time of publication, Dr. John Weatherly was Chief, Terrestrial and Cryospheric Sciences Branch; Mr. Justin Putnam was Acting Chief, Force Projection and Sustainment Branch; Mr. Jimmy Horne was Division Chief and Dr. Bert Davis, was the Technical Director for Geospatial Research and Engineering/Military Engineering. The Deputy Director of ERDC-CRREL was Mr. David B. Ringelberg, and the Director was Dr. Joseph L. Corriveau.

Dr. Jeffrey Hensley was Chief, Computational Analysis Branch and Dr. Jerry Ballard was Chief, Computational Science and Engineering Division. The Deputy Director of ERDC-ITL was Ms. Patti S. Duett and the Director was Dr. David Horne.

This paper was originally published as a proceeding of the ISTVS 15th European-African Regional Conference, Prague, Czech Republic, September 9-11, 2019 and funded under the Entry and Sustainment in Complex Contested Environments Project (Mobility in Peat and Northern Soils project under the 6.2 T40 ASTMIS).

The Commander of ERDC was COL Teresa A. Schlosser and the Director was Dr. David W. Pittman.

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR AUTONOMOUS MILITARY VEHICLES

Abstract

Autonomous vehicles are becoming reality for civilian applications. In the form of intelligent driving assistance, the vehicle autonomy of the third level (smart cruise control, pedestrian recognition, automatic braking, blind zone sensors, rare cross-traffic alerts, collision avoidance, etc.) has been available for commercial and private vehicles for number of years. The autonomy of the fourth and fifth level (supervised autonomy and full unsupervised autonomy) are currently in trials. Despite a substantial progress in this area in civilian applications, autonomy for military vehicles is still quite a challenging task. The main distinctions of military autonomous vehicles are: off road operation, unknown terrain for operation, and a possibility of complete re-routing in the open space. This environment requires different algorithms and environmental awareness for intelligent autonomy controls than those used for civilian applications in the industry. Specifically, the tasks of advanced and current terrain awareness, detection of impassible routes, determination of passible alternative routes and vehicle re-routing in the open space, and optimal vehicle control for a given terrain condition and vehicle need to be solved. The presented work describes recent progress in solving some of such challenges. The results indicate that some of the challenges can be successfully solved by machine learning and artificial intelligence algorithms, thus, providing a substantial aid in manual driving of military vehicles.

1. Introduction

Military vehicles continue to evolve from the un-armored jeeps and tanks of World War 1 to armored HMMWV's, Mine-Resistant Ambush Protected (MRAP) vehicles and heavily armored M1A1 tanks used in the Iraq War and the War in Afghanistan. Current military vehicles are more purpose built for specific fighting styles and conditions, which now include Unmanned Air Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs). This requires more vehicles to be optionally manned, tele-operated, or completely autonomous. These modes of operation are all included as options for the Next Generation Combat Vehicle, specifically Robotic Combat Vehicles (RCVs) (Fig. 1).



Fig. 1. Current robotic platforms available for military use. Left: Milrem robot. Right: Packbot.

Commercial companies like Google, Tesla, Ford and others are currently working to develop optionally manned and fully autonomous vehicles for on-road use with some having entered into advanced road testing phases. These modes of operation require the vehicle to perform two main tasks: 1) drive while staying on the road, avoiding obstacles, and obeying traffic laws, and 2) adapt to changing road surface conditions (rain, snow, ice, mud, etc.). The first task requires sensors like vehicle mounted radar, lidar, cameras, ultrasonic sensors, GPS and others to keep the vehicle on the road and for obstacle avoidance, including static or dynamic objects like parked cars, passing vehicles, or people walking through the streets. The second task requires development of algorithms for advanced traction control systems (TCS), anti-lock braking systems (ABS), and electronic stability control (ESC). While all of these systems are on current passenger and some commercial vehicles, they have been designed and highly optimized for on-road driving conditions and lack the ability to adequately control vehicles in off-road driving conditions and continue to experience limitations when operating in harsh winter road conditions (Fig. 2).



Fig. 2. Vehicles operating in harsh winter conditions.

Even with the current systems on passenger and commercial vehicles and the advancements in optionally manned vehicle technology, the auto industry still has a long way to go before they are ready to release this type of vehicle to the public. Older military vehicles do not have the advanced TCS, ABS, or ESC systems installed and need adequate mobility in off-road terrain conditions with deep snow, mud, ice and other harsh conditions. This requires specific algorithms that are optimized for these conditions where existing systems are inadequate.

Research laboratories within the U. S. Army Corps of Engineers Engineer Research and Development Center (ERDC) are currently developing simulation tools to assist with the development of optionally manned, tele-operated, and fully autonomous vehicles, with emphasis on terrain-vehicle interaction, especially, in the winter terrain conditions. Other ERDC laboratories are focused on assured position, timing, and navigation in conjunction with the Ground Vehicle Systems Command (GVSC), which is conducting research to develop optionally manned and autonomous platforms, focusing largely on the hardware and software within the vehicle with little emphasis on externally mounted terrain sensors or winter operating environments. The U. S. Army Cold Regions Research and Engineering Laboratory (CRREL) conducts vehicle mobility research in winter and extreme environments, which is needed in both the simulation and development of optionally manned and autonomous vehicles. The scope of this work is investigating applicability of artificial intelligence and machine learning for military vehicles operating in winter conditions. This paper describes initial efforts to fulfill this objective.

2. Approach

The overwhelming majority of literature on the subject of autonomous vehicles is devoted to driving in urban conditions. Off-road vehicles do not have roads guiding their trajectory, nor a consistent driving surface, and must also consider their three-dimensional orientation reflecting uneven 3D terrain. This advocates for using more sophisticated artificial intelligence methods, such as the PilotNet convolutional neural network, which recently was used to teach a vehicle to steer itself by recording 72 hours of successful driving in different urban conditions on camera and using these data as a training set (Bojarski et al., 2017). On the other hand, a significant body of knowledge is collected on traditional automatic vehicle control, without using neural networks. For example, the winner of DARPA 2005 challenge team did not use neural networks but relied on more traditional automatic control algorithms for automatic control of their robot Stanley (Thrun et al., 2006). To take advantage of this knowledge, and yet be relevant to challenging requirements for off-road operations, we propose to implement a hybrid approach that combines both artificial intelligence and classic control methods.

Specifically, we propose to use neural networks to continuously determine and update the terrain type on which a vehicle is moving, and a vehicle “critical values,” which are mobility constraints the vehicle must obey along its route, such as the maximum allowed velocity, maximal rate of acceleration and deceleration, and range and maximal rate of steering of the vehicle. Two artificial intelligence algorithms will be used. One for automatic terrain classification, and the other one for predicting critical control values for the terrain type identified by the first algorithm. By using a neural network to predict the critical values, the on-vehicle autonomous control system does not need to specifically account for all terrain types and orientations, but instead be suitably tailored to adjust in real time to the current driving conditions. Figure 3 outlines the model architecture. The current terrain estimate, surface type and condition, desired trajectory, and vehicle state will be used to make predictions of the critical constraints for the velocity, maximal rate of acceleration/deceleration, and steering. These values will serve as target values for the traditional Proportional Integral Derivative (PID) controller for brake/throttle, and Model Predictive Control (MPC) controller for steering. Then, the actual vehicle state will be assessed and the terrain, critical values, and route will be updated accordingly until the vehicle reaches the desired destination.

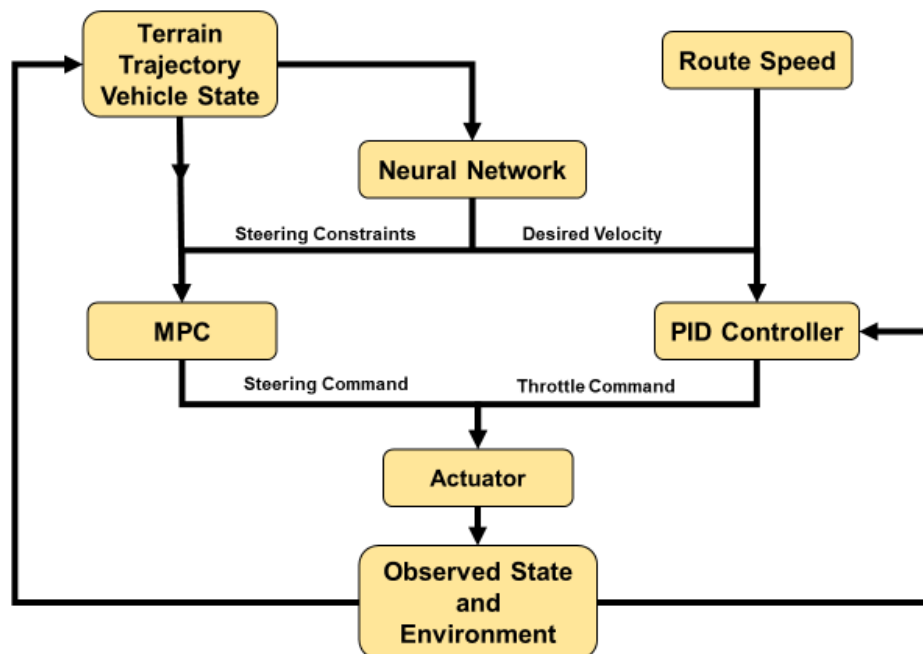


Fig. 3. Architecture of the proposed hybrid autonomous control approach: critical values predicted for given terrain type, vehicle orientation, topography, and surface conditions using neural network, and set as targets for adaptive MPC or PID controllers.

2.1 Automatic Terrain Classification

One of the most common machine learning applications is automatic classification with the goal to determining which “class” a data point belongs to, based on one or more independent attributes. In this study, we attempt to classify different types of terrain to provide look ahead information for autonomous ground vehicles. To accomplish this task, we employed a common classification algorithm, referred to as a Support Vector Machines (SVM). Generally speaking, the SVM method is based on the concept of decision planes, or “hyperplanes”, that separate data points belonging to different classes. These hyperplanes are constructed by determining the greatest distance between points of separate classes from the training datasets. A comprehensive description of this method can be found in Wang (2005).

In addition to the SVM method being suitable for a broad spectrum of machine learning classification problems, there are several other factors that make it well suited for real-time classification problems, which is of great importance for this work. First, SVMs typically use simple decision boundaries, thus, the risk of overfitting is lessened (Wang, 2005). Second, once the SVM has been trained and the hyperplane constructed, this method is computationally inexpensive, which is necessary for real-time classification.

To demonstrate applicability of this method to the considered problem, we elected to classify two different types of terrain, ice and pavement, although this method could be adapted to incorporate several other classes. The input data used for the training dataset include infrared and visual imagery collected using a FLIR camera. Images were collected at different points throughout the winter 2019 season to account for effects of varied solar loading and ambient air temperature. Approximately 150 infrared images and 150 visual images were collected of each class through the data acquisition season. Images of pure ice and pure pavement were collected to make the classification of the training datasets simpler. Several images were taken to capture a contact between ice and pavement to be used as the test data for the analysis (Fig. 4).

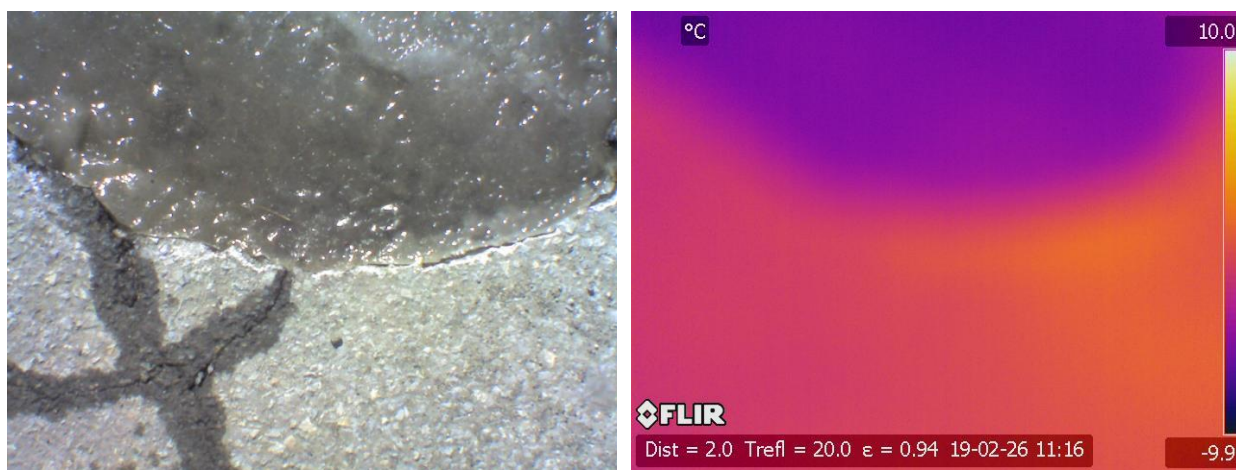


Fig. 4. Example image for testing asphalt/ice classification using SVM. Left: visual image. Right: thermal image.

In this analysis, the Scikit-learn Python® library (Pedregosa et al., 2011) was used as it provides readily available SVM functionality that is well tested. A linear kernel was used along with a default penalty parameter of 1.0. All other optional parameters were set to their default values. The results of this analysis applied to the test image in Fig. 4 can be seen plotted in Fig. 5 (left) in the axes of visual image intensity and surface temperature and Fig. 5 (right) in the original spatial axes. Overall, the classification method worked well given the input data used for the analysis. The placement of the hyperplane seems to be dominated by the thermal signature of the different classes, which, given the temperature differences of ice and pavement, fits with one’s intuition. While this analysis provides apposite initial outlook for the viability of this method, further testing will be required to determine if it will be a robust method for a multi-class classification terrain problem.

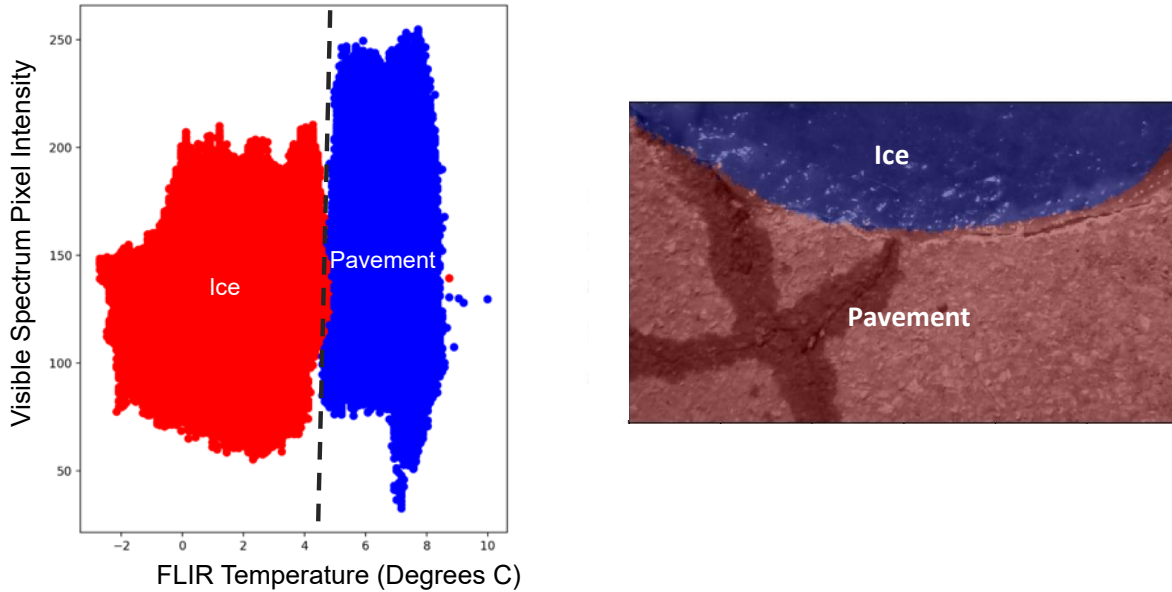


Fig. 5. Results of SVM classification algorithm applied to the test images in Fig. 4. Left: intensity-temperature space. Right: mapping identified classes on the original image in spatial axes.

2.2 Throttle/Brake and Steering controllers

PID controllers are a well-known classical method for velocity control, which predates autonomous vehicles, responsible for vehicle's cruise control setting. The PID controllers have been implemented in vehicles where the desired speed varies to satisfy changing needs, such as the 2005 DARPA Grand Challenge winner Stanley (Thrun et al., 2006). In order to operate under an increasing number of requirements, PID controllers have been modified so that their gains are adjusted via gain scheduling or adaptive control (Ioannou et al., 1993, Ioannou and Sun, 2012, Nie et al., 2018).

Steering control algorithms have seen an even wider array of control methods. The simplest control calculates a desired steering angle based on the car's orientation and distance relative to a desired trajectory, as used by Stanley (Thrun et al., 2006). A widely used alternative is the PID controller, notably the adaptive PID controller, where the gains are tuned offline for various states and updated accordingly or adjusted in real time using neural networks, genetic algorithms, or fuzzy logic (Puntunan and Parnichkun, 2006, Scott et al., 1992, Wai, 2003, Ghanbari and Noorani, 2011). The last dominant control method used in the industry is Model Predictive Control (MPC), which uses a model of the steering input effect on the vehicle trajectory and attempts to find an optimal path (Lenain et al., 2005, Borrelli et al., 2005, Falcone et al., 2007a, Falcone et al., 2007b). A more detailed account of the different methodologies can be found in Zhao et al. (2012).

In our implementation of autonomous vehicle control, a user is able to choose a desired driving mode from three possible options: safe, normal, and aggressive. These modes will be tailored to meet the needs of the user depending on the situation. For example, if the vehicle is carrying an injured soldier and minimizing travel time is essential, then the aggressive driving mode would be selected. The three modes alter the target vehicle speed, acceleration/deceleration, and steering by multiplying the maximal possible values by 0.3, 0.5, or 0.9, which correspond to a driving mode of safe, normal, or aggressive, respectively. Then, the PID and MPC controllers will account for the selected driving mode. The logic of the autonomous control is described below.

We combined the throttle and the brake controls into a single variable whose range is the interval $[-100,100]$. A positive number indicates the percentage of the throttle engaged, a negative number is the percentage of the brake applied, and 0 indicates that neither the throttle nor the break are engaged. Recall that a classic PID controller takes the current measured state of the vehicle as input, which in our case is its current velocity, and a desired state which we will refer to as the desired velocity. The controller output $u(t)$ represents the change in throttle percentage and is given by $u(t) = K_p \varepsilon(t) + K_d \varepsilon'(t) + K_i \int \varepsilon(t) dt$. The error term $\varepsilon(t)$ is the difference between the desired velocity

and the measured velocity, and K_p , K_d , and K_i are referred to as the proportional, derivative, and integral gains, respectively. We have chosen for $u(t)$ to represent the change in throttle percentage instead of the absolute percentage in order to avoid adding a constant term that is the amount of throttle required to maintain a constant speed. Since this amount changes based on the terrain (e.g. going uphill or over snow requires more throttle for a set speed), we ignore the difficulty of calculating or measuring this term and simply let the controller adjust the throttle depending on whether the desired speed is attained.

Through our simulations on a flat terrain, we have determined that the integral gain K_i can be set equal to zero, because it did not affect the performance of the controller. Therefore, our controller is referred to as a PD controller with the integral term omitted. We note that this seems to be an anomaly within the PID controller community, as most controllers are PI or PID controllers. Testing a PD controller on a vehicle may reveal that the integral term is needed. If so, the current implementation of the PD algorithm will need to be altered to include an anti-windup term that does not integrate error terms when the throttle is already saturated.

To regulate the steering commands, we have decided to implement model predictive control (MPC). A brief overview of MPC for steering follows, whereas a detailed description of these controllers can be found in Khaled and Pattel (2018). The goal of MPC is to predict the vehicle's future trajectory over a finite horizon window, the prediction horizon, based off a sequence of steering commands. To do so, a well-designed model of how the input affects the vehicle's trajectory is required. The steering commands are interpreted as an input signal that is a piecewise constant function which can only change values at sample times. For every sample time there is a reference trajectory the vehicle is attempting to follow. The algorithm generates an input signal and uses the model to predict what the vehicle's trajectory would be over the prediction horizon. This trajectory is compared to the reference via a cost function and the algorithm intelligently updates the steering commands to both minimize the cost of the trajectory and to ensure that all constraints of the input and output are satisfied. After a number of iterations, the algorithm finds an optimal trajectory. The program then relays the input signal value at the first sample time to the actuator. After this sample time has passed, the entire process is repeated. Note that only the first part of the input signal obtained from the optimization is implemented; the rest is discarded.

Our reasoning for deciding to use MPC for the vehicle's steering is as follows. Since vehicle steering is well-understood, an accurate model of the vehicle's trajectory should be straightforward to design. One major strength of MPC lies in its ability to optimize trajectories that obey desired constraints on the input and output. For example, once the critical values of the vehicle's state are determined, one can require that a maximum steering range not be exceeded in order to prevent skidding or loss of vehicle control. In addition, to managing constraints, the adaptability of the cost function allows us to optimize trajectories that reflect the selected driving mode. This includes how the vehicle avoids objects in its path and the perceived smoothness of the steering for passenger comfort.

3. Simulation

For simulation, we consider a test problem on the racetrack route depicted in Fig. 6. The track consists of two parallel segments connected by the semi-circles at the ends, and is located on flat asphalt. The racetrack's straight sections are 500 meters long and the semicircles on both ends have a radius of 75 meters. The vehicle drives one lap around the racetrack in the aggressive mode starting from rest and comes to a complete stop at the end of the course. Steering control using MPC is not yet implemented and so the model only regulates the vehicle's velocity.

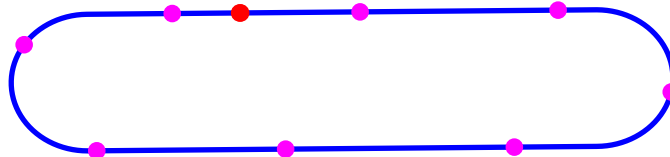


Fig. 6. A test route used in the simulation.

Before driving is initialized, the maximum speeds along the route must be determined. We assume that the route is specified by a set of waypoints with, generally, a non-uniform spacing. Different algorithms exist for finding optimal routes at complex terrain conditions, and this task is not discussed here. The waypoints are passed into an algorithm

that interpolates a curve and outputs uniformly spaced points along the curve with any desired resolution. These replace the initial waypoints. Next, the maximum velocity the vehicle may travel without slipping is estimated at every waypoint. For a flat uniform terrain, this can be implemented quite efficiently by assuming that the maximal radial acceleration allowed is the same, and, thus, the maximal speed depends only on the instantaneous radius of the path curvature. In a general case of non-flat heterogeneous terrains, we assign this task to the second neural network. For the flat terrain, however, three points on the surface uniquely determine a circle, and, thus, we use three consecutive waypoints to calculate the instantaneous radius. This radius represents the local curvature of the route at a particular waypoint and is used with the maximum allowed centripetal acceleration to calculate the maximum velocity the vehicle can safely drive. Note that such a procedure requires only measurements of the maximal speed at a single known radius (e.g. from a circle test breakout). For a straight line segments of the route (infinite radius), we limit the maximal speed of the vehicle to that which is allowed for given terrain.

Once the maximal waypoint speeds have been determined, the target driving speed for the PD controller is the minimum of the following two speeds. One is the speed recommended by the neural network. The other is the minimum speed among all the waypoints in front of the vehicle whose distance along the route from the vehicle current location is less than twice the vehicle current braking distance. So, the target speed to the PD controller might be different (smaller) from the maximal speed associated with every waypoint. Clearly the minimum look-ahead distance must be the braking distance of the vehicle in case it must stop. We have also experimented with altering the speeds associated with the waypoints. Instead of assigning the maximum allowed speed to a given waypoint, we apply an algorithm that works backwards along the route, calculating the fastest the vehicle may travel in order to safely decelerate and meet any upcoming waypoint speed requirements, avoiding rapid braking. An example showing differences in the target speeds along the route obtained with these two methods is shown in Fig. 7.

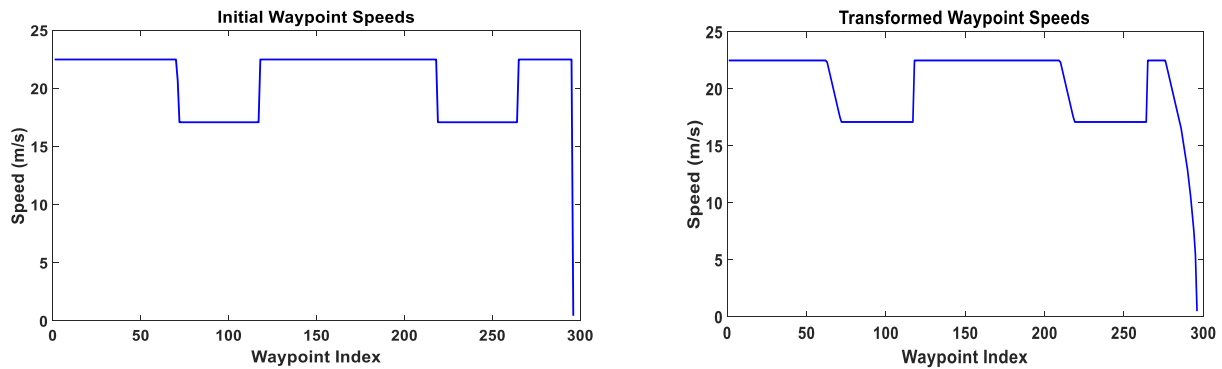


Fig. 7. Example of the different desired speeds along the route. Left: the minimal allowed speed at two braking distances ahead of the vehicle. Right: the minimal allowed speed obtained by backwards calculations assuming smooth braking. This makes autonomous acceleration and deceleration smoother, especially, when the vehicle needs to be brought to a full stop.

In Fig. 7, the flat segments of the target speeds at the 23 m/s correspond to the straight route segments. Then, as the vehicle approaches the turning point, it detects that the maximal allowed speed should be lower, 17 m/s along the curves. The simple look-ahead strategy prescribes this value, based on the curved section, and, as a result, one can see sharp target speed decrease for the curved segments (Fig. 7, left). The PD controller may want to prescribe aggressive braking at this point. In the backwards calculations, however, with the requirement of smooth deceleration, the prescribed lower target speeds are entered into the PD controller earlier, which results in more gradual deceleration and a smoother target speed profile (Fig. 7, right). To simulate the observed speed as a response of the PD controller output, we use standard dynamics to approximate the forces the car observes while in motion, including drag from wind and the tires.

Figure 8 shows the results of the simulation. The waypoints along the route are numbered to see the correspondence between location on the track (left) and vehicle speed (right). The red target speeds in Fig. 8, right, correspond to the simple look ahead target speed determination, while the green target speeds correspond to the backwards calculations that were given to the PD controller. One can see that the resulting simulated “actual” vehicle speed closely follows the target speed (the green curve) and is always below the maximal allowed (the red curve), which is a desired outcome.

Figure 9 illustrates the difference between which waypoint speeds are set as target values to the PD controller. Setting the initial maximal waypoint speeds leads to a more aggressive vehicle behavior (faster

acceleration/deceleration) because of the larger error differences between the current and target speeds. When the vehicle is transitioning from the straight track to the curve, only the speed along the curve is set as a target into the PD controller, and the proportional gain causes the PD controller to have a larger response compared to the transformed target speeds, which were calculated using backward technique. Not only is the error typically lower when the PD uses the transformed speeds, but the vehicle observes an oncoming speed reduction sooner and begins to decelerate earlier.

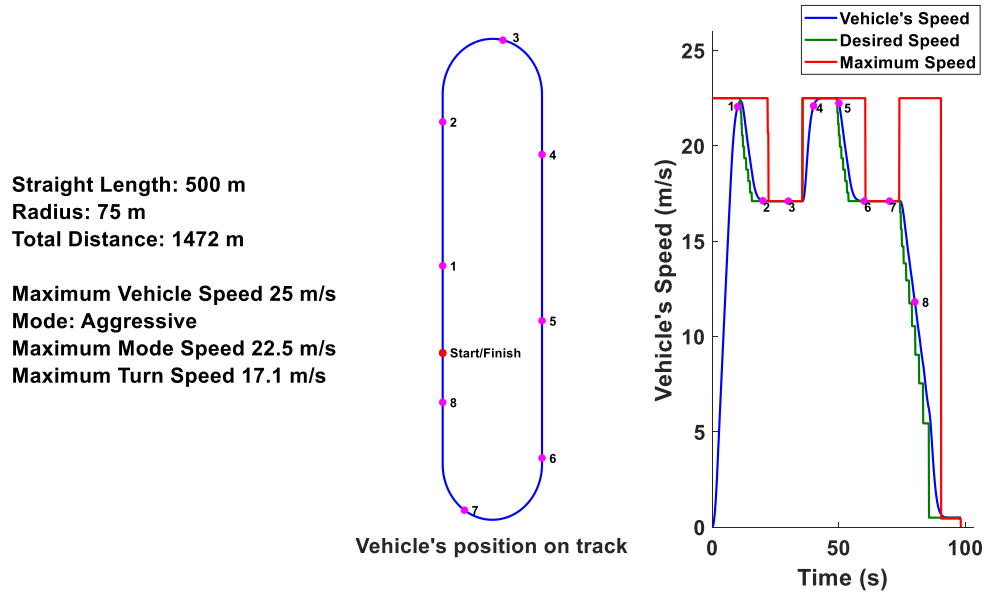


Fig. 8. Simulation results. Left: the test route. Right: target and actual vehicle speeds along the track. The waypoints are numbered to see the correspondence of the vehicle location on the route (left) and its speed (right).

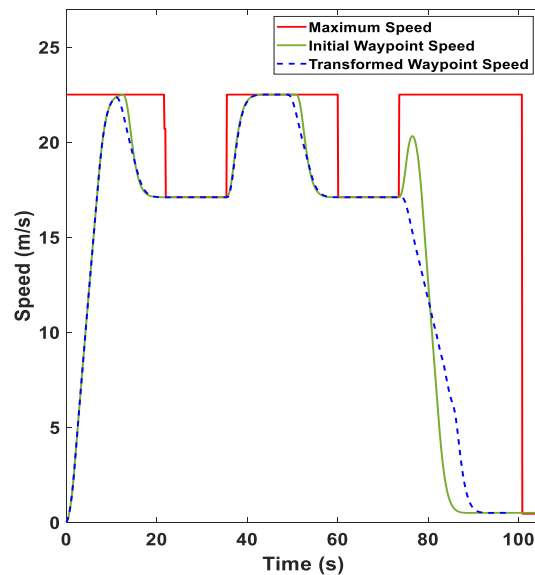


Fig. 9. Difference in vehicle performance caused by different choices of the target speeds. Without backwards calculations, vehicles is being driven in more aggressive mode.

The difference is clearly seen at the end of the lap, 80 to 100 seconds on the time axis. With the original look-ahead strategy (green line), the vehicle still accelerates right before the finishing location, while the backwards calculated target speeds result in more cautious driving (dashed blue line).

One issue we encountered was how quickly the braking distance decreased as a function of velocity at braking versus how quickly the PD controller could decelerate the vehicle to the target speed. This issue was observed when the PD target speed was zero, that is, when the vehicle needs to be brought to full stop. If the PD controller decelerates the vehicle too quickly, the look-ahead distance decreases fast enough so that, at some instance, the target speed for the PD controller, which is sought at two braking distances ahead, is no longer zero, because the braking distance became too short. This, in turn, causes the vehicle to increase its speed until its look-ahead distance is great enough to observe the waypoint with zero-speed waypoint. We have observed that this behavior can be reduced or avoided when using the transformed waypoint speeds, which is what the current implementation of the PD controller uses. This issue can be corrected by prohibiting the look-ahead distance from decreasing below a fixed target speed value. Alternatively, the contribution from a trained neural network can prevent this issue from occurring.

4. Conclusion

The paper describes initial efforts in the implementation of the hybrid approach for autonomous control for vehicles operating in off-road and harsh conditions, like ice, snow, mud, etc. The hybrid approach combines two different technical approaches: artificial intelligence (neural networks and related methods) for predicting the type of terrain the vehicle is on and critical values for vehicle controls, like maximal acceleration/deceleration, maximal steering, maximal speed, maximal rate of acceleration/deceleration, etc. Then, these limits set as hard limitations or target values for tradition control mechanisms (PID, MPC) to navigate the vehicle. The route of the vehicle, as well as the terrain and condition information in front of the vehicle are updated regularly as the vehicle is moving. So, the changing of information in time inherently incorporated in the proposed approach. The approach also allows to choose the style for autonomous control, such as safe, normal, and aggressive. It is anticipated that such feature will be very relevant to military operations. The initial results of numerical simulations are promising, but real testing is needed to judge the effectiveness and viability of the approach, which we plan to conduct in the future.

Acknowledgements

The work was supported by the ERDC-CRREL vehicle mobility business area from the projects Vehicle-Born IED Entry Control Point Screening (work item funding 0162GF), Machine Learning Analytics for Cold Mobility (work item funding 261L1H), and Extreme Terrain Research - Mobility Business Development Initiative (work item funding 6J1D1H). This research was supported in part by an appointment with the National Science Foundation (NSF) Mathematical Sciences Graduate Internship (MSGI) Program sponsored by the NSF Division of Mathematical Sciences. This program is administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and NSF. ORISE is managed for DOE by ORAU. All opinions expressed in this paper are the author's and do not necessarily reflect the policies and views of NSF, ORAU/ORISE, CRREL, or DOE. Permission for publishing was granted by the Director, Cold Regions Research and Engineering Laboratory.

References

- Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., Muller, U., 2017. Explaining how a deep neural network trained with end-to-end learning steers a car. arXiv preprint arXiv:1704.07911.
- Borrelli, F., Falcone, P., Keviczky, T., Asgari, J., Hrovat, D., 2005. MPC-based approach to active steering for autonomous vehicle systems. *Int. J. Veh. Auton. Syst.* 3(2), 265–291.
- Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., Hrovat, D., 2007a. A model predictive control approach for combined braking and steering in autonomous vehicles, in 2007 Mediterranean Conf. Control Autom., IEEE, 1–6.
- Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., Hrovat, D., 2007b. Predictive active steering control for autonomous vehicle systems. *IEEE Trans. Control Syst. Technol.* 15(3), 566–580.
- Ghanbari, A., Noorani, S., 2011. Optimal trajectory planning for design of a crawling gait in a robot using genetic algorithm. *Int. J. Adv. Robot. Syst.* 8(1), 29–36.
- Ioannou, P., Xu, Z., Eckert, S., Clemons, D., Sieja, T., 1993. Intelligent cruise control: theory and experiment, in Proc. 32nd IEEE Conf. Decis. Control, 1885–1890.

- Ioannou, P. A., Sun, J., 2012. Robust adaptive control, second ed. Dover, Meneola, New York.
- Khaled, N., Pattel, B., 2018. Practical design and application of model predictive control: MPC for MATLAB® and Simulink® users. Butterworth-Heinemann, Cambridge.
- Lenain, R., Thuilot, B., Cariou, C., Martinet, P., 2005. Model predictive control for vehicle guidance in presence of sliding: application to farm vehicles path tracking, in Proc. 2005 IEEE Int. Conf. Robot. Autom., 885–890.
- Nie, L., Guan, J., Lu, C., Zheng, H. & Yin, Z., 2018. Longitudinal speed control of autonomous vehicle based on a self-adaptive PID of radial basis function neural network. IET Intell. Trans. Syst. 12(6), 485–494.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12(10), 2825–2830.
- Puntunan, S., Parnichkun, M., 2006. Online self-tuning precompensation for a PID heading control of a flying robot. Int. J. Adv. Robot. Syst. 3(4), 323–330.
- Scott, G. M., Shavlik, J. W., Ray, W. H., 1992. Refining PID controllers using neural networks. Adv. Neural Inform. Proces. Syst., 555–562.
- Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., 2006. Stanley: The robot that won the DARPA Grand Challenge. J. Field Robot. 23(9), 661–692.
- Wai, R. J., 2003. Tracking control based on neural network strategy for robot manipulator. Neurocomputing 51, 425–445.
- Wang, L. (Ed.), 2005. Support vector machines: theory and applications. Springer, Berlin, Heidelberg, New York.
- Zhao, P., Chen, J., Song, Y., Tao, X., Xu, T., Mei, T., 2012. Design of a control system for an autonomous vehicle based on adaptive PID. Int. J. Adv. Robot. Syst. 9(2), 44–54.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE August 2020			2. REPORT TYPE Final			3. DATES COVERED (From - To)			
4. TITLE AND SUBTITLE Artificial Intelligence and Machine Learning for Autonomous Military Vehicles						5a. CONTRACT NUMBER			
						5b. GRANT NUMBER			
						5c. PROGRAM ELEMENT NUMBER			
6. AUTHOR(S) Sergey Vecherin, Jacob Desmond, Taylor Hodgdon, Jordan Bates, Michael Parker, James Lever, Garrett Hoch, Mark Bodie, and Sally Shoop						5d. PROJECT NUMBER 471941			
						5e. TASK NUMBER 48			
						5f. WORK UNIT NUMBER			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Engineer Research and Development Center Cold Regions Research Laboratory Information Technology Laboratory 72 Lyme Road 3909 Halls Ferry Road Hanover, NH 03755 Vicksburg, MS 39180						8. PERFORMING ORGANIZATION REPORT NUMBER ERDC MP-20-1			
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Assistant Secretary of the Army/Acquisitions, Logistics & Technology 103 Pentagon Washington, DC 20314-1000						10. SPONSOR/MONITOR'S ACRONYM(S)			
						11. SPONSOR/MONITOR'S REPORT NUMBER(S)			
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.									
13. SUPPLEMENTARY NOTES Originally published in <i>Proceedings of the ISTVS 15th European-African Regional Conference</i> , Prague, Czech Republic, September 9-11, 2019.									
14. ABSTRACT Autonomous vehicles are becoming reality for civilian applications. In the form of intelligent driving assistance, the vehicle autonomy of the third level (smart cruise control, pedestrian recognition, automatic braking, blind zone sensors, rare cross-traffic alerts, collision avoidance, etc.) has been available for commercial and private vehicles for number of years. The autonomy of the fourth and fifth level (supervised autonomy and full unsupervised autonomy) are currently in trials. Despite a substantial progress in this area in civilian applications, autonomy for military vehicles is still quite a challenging task. The main distinctions of military autonomous vehicles are: off road operation, unknown terrain for operation, and a possibility of complete re-routing in the open space. This environment requires different algorithms and environmental awareness for intelligent autonomy controls than those used for civilian applications in the industry. Specifically, the tasks of advanced and current terrain awareness, detection of impassible routes, determination of passible alternative routes and vehicle re-routing in the open space, and optimal vehicle control for a given terrain condition and vehicle need to be solved. The presented work describes recent progress in solving some of such challenges. The results indicate that some of the challenges can be successfully solved by machine learning and artificial intelligence algorithms, thus, providing a substantial aid in manual driving of military vehicles.									
15. SUBJECT TERMS Artificial Intelligence, Autonomous Vehicles, Machine Learning, Military Vehicles, Terrain Mobility									
16. SECURITY CLASSIFICATION OF:						17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area code)				