IARRC 2018: Wild Bobcats

Letian Lin, Yingnan Zhang, Yichao Li. Yang Liu, Yuanyan Chen, Miguel Sempertegui Sosa, Stuart Randle, Hong Zhang, Matchima Buddhanoy, Krerkkiat Chusap, Jonathan Waters, David Wisniewski, Logan Wilkovich, Dylan Denner, Archie Scott, Mitchel Brightman
Faculty Advisor: J. Jim Zhu (zhuj@ohio.edu)

Ohio University Professsional Autonomous Vehicle Engineering Team (OU-PAVE)
Russ College of Engineering and Technology, Stocker Center 155, Athens Ohio 45701

I. INTROFUCTION

The Ohio University Professional Autonomous Vehicle Engineers (OU_PAVE) team undertook the challenges in developing the technology for high-speed autonomous vehicles and robots, and entered the IARRC completion. In particular, the competition requirements call for:

- An autonomous car that has the racing ability under the realistic race conditions as well as guarantees safety for both vehicles and competitors
- High-speed vehicle localization
- High-speed vehicle control (acceleration and braking) on different surfaces
- Start/Stop light and roadway detection
- Collision avoidance with static objects along boundaries of course
- Collision avoidance with other competing robots.

The team followed the profession engineering design practices, started by formulating the design rules into engineering requirements, and went through many formal design and testing reviews. The completion of the project also helped the students to develop many useful skills in teamwork, communications, ethics, and project management. This report summarizes the main technical design and testing results of our entry: The Wild Bobcat autonomous vehicle.

II. VEHICLE SUBSYSTEM DESIGN

Based on the vehicle size, weight and speed requirements, our Wild Bobcat was modified from a commercial RC model monster truck, the Traxxax X-MAXX, as shown in Figure 1. In order to satisfy the length requirement of no more than 0.75 m, the wheelie wheel in the back of the vehicle was removed. Three cameras, one 2D Lidar and two sonar sensors were also installed on the body shell, and an instrument panel was installed inside on the chassis. A custom made motor shaft encode was installed on the motor pinion gear. The total weight was controlled to be under 12 kg. The finished vehicle is shown in Section VIII.

Figure 1. Traxxas X-MAXX

III. ELECTRICAL AND ELECTRONICS DESIGN

The electrical and electronics design includes power distribution and signal communications. Figure 2 shows the computing platforms, sensors, actuators, wireless communication antennas along with the power lines and signal communication protocols. It is noted that the wireless E-Stop transmitter and receiver are custom made. Also, the onboard e-stop system includes a regenerative brake using a 3-phase rectifier and a bank of 12V/50W lightbulbs to quickly dissipate the vehicle's kinetic energy when braking at high speed.

IV. SYSTEM ARCHITECTURE

The autonomous vehicle control system comprises three subsystems: (i) navigation (localization), (ii) cognitive control and (iii) motion control, as shown in Figure 3. Each subsystem will be described in the following sections.

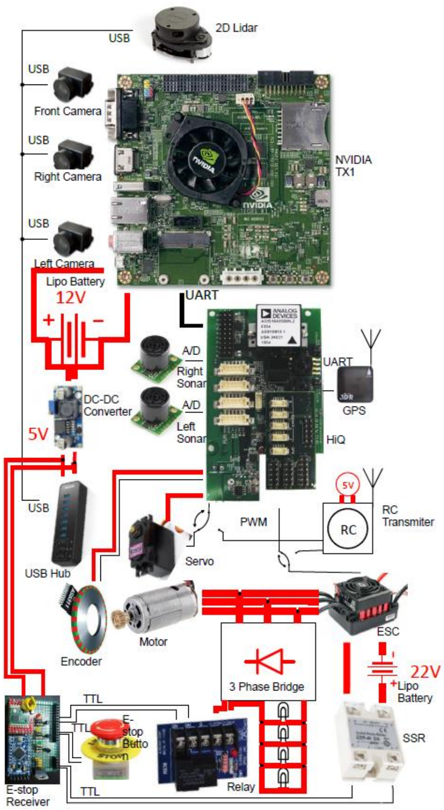

Figure 2. Electrical and Electronics Subsystem Design

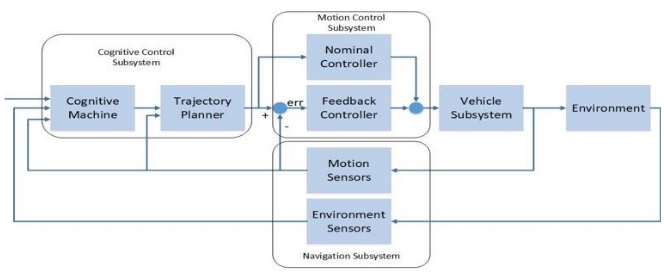

Figure 3. Autonomous Vehicle Control System Architecture

V. NAVIGATION (LOCALIZATION) SUBSYSTEM DESIGN

The navigation subsystem consists of motion sensors and environment sensors. High-speed localization relies on the motion sensors, while autonomous driving operations, such as lane following, start/stop signaling and moving obstacle avoidance reply on the environment sensors.

The Motion Sensor System includes GPS/INS (Inertial Navigation System and velocity encoder data processing. In practice, the onboard inertial sensors, like gyroscopes and accelerometers will have some bias. If these signals are used directly to calculate the orientation and position by integration, the results would drift because of the bias. Comparing to inertial sensors, GPS is much more accurate over time, but the data update frequency is lower than inertial sensors. The quadrature motor shaft encoder can be erroneous when vehicle is in dynamic motion due to wheel slipping and skidding, but provides relatively accurate velocity data when the vehicle is running at constant velocity.

All of the on-board sensors data, like gyroscopes, accelerometers and encoder data update at 100 Hz, which ensure the high-speed localization. In this project, a 3DR GPS module with Ublox LEA-6H GPS chip is introduced. The accuracy is up to ± 2 meters at 15 satellites fix and 1.0 HDOP. The update rate is 5 Hz. A bias removal function is applied to the inertial sensors, and a Kalman filter algorithm is employed for inertial and shaft encoder sensor data fusion. For outdoor operations, the GPS and magnetometer data are also fused in with the inertial and encoder data using the Kalman filter.

The vision system is specifically designed for the IARRC competition. It consists of two parts: the lane detection algorithm that is primarily based on OpenCV and the object detection algorithm primarily that is based on TensorFlow Object Detection API.

The lane detection algorithm is different from the standard Hough Transformation, which runs in cubic time complexity and is not applicable in real time system [1] [2]. Three lookout range lines that mark 1m, 2m, 3m distance from the car are set for each frame. At each lookout range we detect the white color (left line), yellow color (right line), and magenta color (end line). Points are fitted into a linear function using scipy.stats.linregress for further reducing the noise and determining the left turn and right turn moves. The vision sensor data are first passed to an event generator. The output events then serve as the input to the cognitive state machine. Table 1 shows all the data detected in the vision system.

The object detection algorithm is based on the Single Shot Detector Lite – MobileNet V2 model [3]. Pretrained SSDLite-MobileNetV2 using the COCO dataset was downloaded from the TensorFlow Model Zoo. Such model was retrained to detect red/green traffic light and cones.

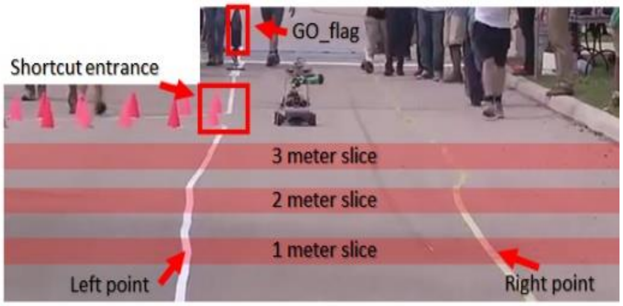

Figure 4. Example of detected lanes and objects.

Table 1. Vision data and event generation

Source	Data	Data Format	Trigged event
Front camera	Left_line_point Right_line_point	Float	Only left line, Only right line, Both lines
	Finish_line_distance		Finish line (1m, 2m, 3m)
	H_left_line		Horizontal left line (1m, 2m, 3m)
	H_right_line		Horizontal right line (1m, 2m, 3m)
	Left_cone		Shortcut entrance (1m, 2m, 3m), Shortcut exit (1m, 2m, 3m)
	Right_cone		Shortcut entrance (1m, 2m, 3m) , Shortcut exit (1m, 2m, 3m)
	GreenLight	Boolean	Green Light
Left camera	Left_line_distance	Float	Left line close
Right camera	Right_line_distance		Right line close

VI. Cognitive Subsystem Design

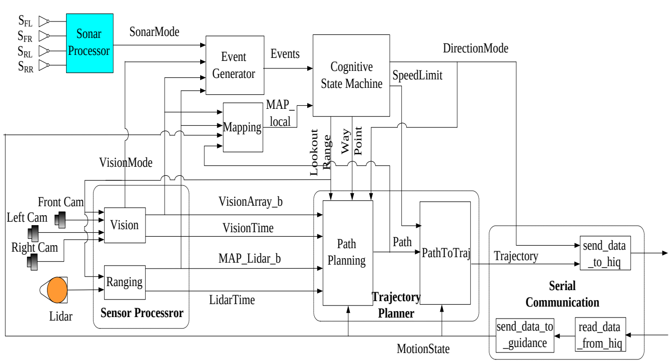

Figure 5. Cognitive Control Subsystem.

Cognitive state machine. The overall cognitive control system is shown in Fig. 5, where "left/right" refer to the driver's left/right. The event-driven cognitive state machine which is a Mealy state machine conducts the high level decision. The events which are the output of the environment sensor system serve as the input to the cognitive state machine. The events together with the current state determine the state transition. The output logic sets the appropriate value to the outputs according to the current state and input.

The events that passed to the cognitive state machine are labeled with numbers as shown in the following tables.

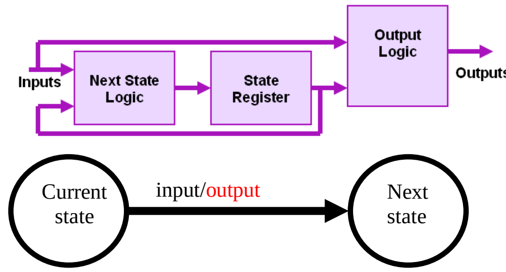

Figure 6. Principle of a Mealy machine

The state of the system is represented by a 5-digit decimal number of the form (Traveling mode, Turn, Car avoidance, Car chasing, Course selection). Each digit represents the value of a specific state variable.

Event class	Event (Input)
Light	0 - Green Light
Side line	1 - Only left line
	2 - Only right line
	3 - Left line close
	4 - Right line close
Shortcut	5 - Shortcut entrance
	6 - Shortcut exit
Horizontal line	7 - Horizontal left line (1m, 2m, 3m)
	8 - Horizontal right line (1m, 2m, 3m)
	9 - Finish line (1m, 2m, 3m)
Other cars in the course	10 - Left car
	11 - Right car
	12 - Front car

Table 2. Cognitive events

State Variable	Value
Traveling mode	0 - Ready
	1 - Go
	2 - Finish
Turn	0 - No turn
	1 - Left turn
	2 - Right turn
Car avoidance	0 – No car
	1 - Left car avoidance
	2 - Right car avoidance
	3 - Front car avoidance
Car chasing	0 - No chasing
	1 - Chasing
Course selection	0 - Normal
	1 - Go shortcut
Lap number counter	n - Up to N laps

Table 3. Cognitive states

The outputs of the state machine are listed as following. The output is represented by a 4-digit decimal number of the form (Direction mode, Speed limit level, Lookout range level, Waypoint selection scenario).

An example for the state transition of the cognitive state machine is shown in Fig. 6, where the car starts to run when the traffic light is switched to green, then it takes a sharp right turn when a horizontal left line is detected. After the car goes back to the straight course, it passes a front car. After the car runs two laps, the car stops at the finish line. In the Fig 7, the capital E, S and O represents the event, the state and the output, respectively. The resulted state transition is: S00000 → S10000 → S12000 → S10000 → S00300 → S10000 → n+1 → S10000 → S12000 → S10000 → S20000.

Output variable	Value
Direction Mode	0 - Forward
	1 - Stop
	2 - Reverse
Speed Limit level	0 - Slow
	1 - Medium
	2 - Fast
Lookout Range level	0 - Near
	1 - Medium
	2 - Far
Waypoint selection scenario	0 - Both lines
	1 - Only left line
	2 - Only right line
	3 - Horizontal left line
	4 - Horizontal right line
	5 - Left car
	6 - Right car
	7 - Front car
	8 - Between cones
	9 – No way out

Table 4. Cognitive outputs

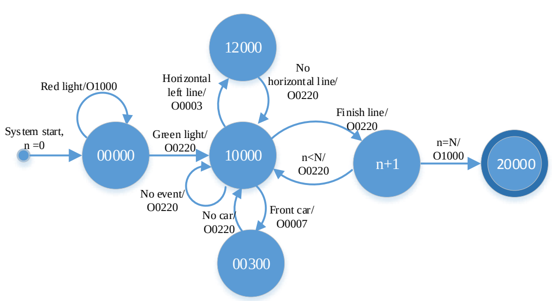

Figure 7. The state transition diagram for the example

Path Planning. The path planner is responsible for generating a feasible, collision free path that leads the vehicle to the target. In our design, the path planner produces the waypoints based on the output of the cognitive state machine. As an example, consider the following two different driving scenarios: driving without other cars in the course and driving with other cars in the course, which are shown in the following Fig 8. Scenario 1 consists of 2 sub-scenarios: the case when two boundary lines can be seen by the camera and the case when only one line can be seen.

Consider the scenario that no car is in sight. If both of the boundary lines are in the view of the camera, then the waypoint is simply taken as the middle point of the intersections of the lookout range line and the boundary lines. If only one boundary line is in the view, for instance, the left line, then the waypoint is taken on the lookout range line on the right of the left line with a proper distance. In the scenario that there are other cars in sight, the way point is properly taken between the lines and the other cars to allow the car to go through without collisions. If there is no sufficient space to overtaken the front car, then the speed of the vehicle is reduced and the chasing state is entered.

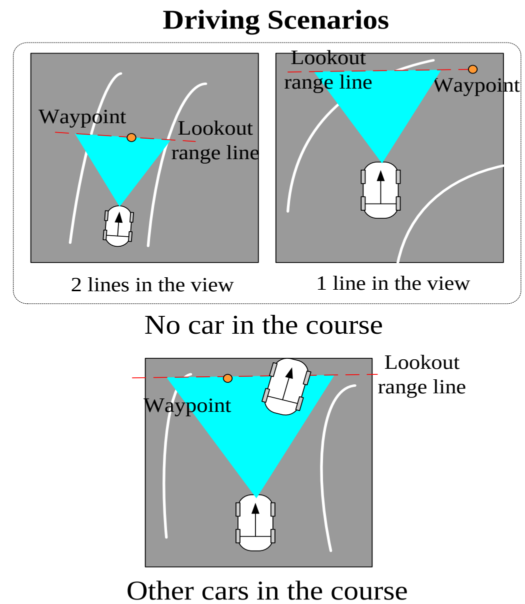

Figure 8. Path planning Scenarios

Path-to-trajectory conversion. The path-to-trajectory converter is used to assign a feasible velocity profile along the path and thus convert it to a nominal trajectory. Out optimality objective for velocity assignment is to minimize the traveling time of the car to reach the goal. Also, the path-to-trajectory conversion must be subject to both the kinematics and dynamics constraints of the car. The traditional approach is formulating the path-to-trajectory conversion as a free-ending-time optimal control problem. However, the corresponding computing time is usually too long to satisfy the real-time requirement.

By the dimension of the Wild Bobcat, the competition course (including the shortcut) is sufficiently wide for the car to drive without needing backups. Therefore, after the path planner determines the waypoints in the course properly, a fast line-of-sight pure pursuit guidance (LOS PPG) [4] is adapted to conduct the trajectory tracking guidance. LOS PPG has been used in aircraft guidance design. By taking into account the nonholonomic constraint, as well as the constraints on the linear and angular velocities, linear and angular accelerations, and the curvature, we modified LOS PPG to generate a feasible trajectory for car-like ground vehicles. The LOS PPG trajectory generator consists of four subsystems, including target seeker, heading guidance, speed guidance and trajectory synthesizer, as shown in Fig 9.

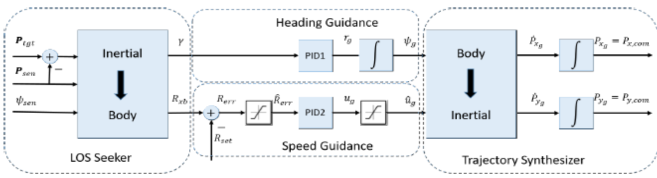

Figure 9. LOS PPG Trajectory Generation Architecture

The simulation results are shown in Section V.

Mapping. By using the data from the environment sensors and the motion states from the motion sensors, the mapping system updates the global map in the memory and retrieves a local map for the cognitive control purpose. The map is internally represented by a grayscale bitmap of range from 0 to 100, where each pixel corresponds to an inertial coordinate. For each type of the objects, the pixels are assigned to a pre-defined grayscale as shown in the following table.

VII. Motion Control Subsystem Design

3 Degree-of-Freedom (DOF) motion may be categorized as path-following and trajectory-tracking. Path-following only requires the vehicle to follow a specified path without time constraints. Thus, path-following controller systems only need to deal with vehicle kinematics. In contrast, trajectory-tracking control systems require the vehicle to traverse a prescribed path with a given velocity. Trajectory-tracking is more challenging than path-following because the vehicle dynamics must be considered in addition to vehicle kinematics. For an under-actuated, nonholonomic car-like ground vehicle trajectory tracking Guidance, Navigation, and Control (GNC), we consider the 3DOF nonlinear vehicle rigid-body dynamics with nonlinear tire tracking force, nonlinear drag force and actuator dynamics [5]. Conventional automatic motion controllers for cars use separate controllers for steering and throttle, which tends to limit the performance potential of the vehicle. In order to effectively cope with the nonlinear and time-varying nature of ground vehicle motion control. conventional systems may use what is known as a Model Predictive Control (MPC) technique. MPC runs a simulation of the vehicle motion with currently computed controller gains. MPC performs an on-line optimal control design to obtain a new set of gains, and repeats the process at every control decision step, typically between 50 and 100 times per second. Such controllers are extremely time consuming yet with limited performance and a lack of stability. Compare to MPC, TLC is high computational efficiency and effectiveness for high-order nonlinear plant. The controller design has been filed an international patent application under the PCT.

Object	Grayscale
Unexplored area	0
Explored free area	10
Traffic light	20
Start line	30
Finish line	40
Left line	50
Right line	60
Shortcut entrance	70
Shortcut exit	80
Cone	90
Path	100

Table 5. Grayscale table for the objects

The control algorithm employs Trajectory Linearization Control (TLC) based on singular perturbation theory. TLC is a model-based controller, and it provides a nonlinear time-varying controller that combines nonlinear dynamic inversion (nominal controller) with linear time-varying feedback stabilization (feedback controller), as shown in Fig 10. The nominal controller approximately cancels the plant nonlinearity, thereby reducing the tracking error to facilitate linearization of the nonlinear tracking error dynamics. The linearized error dynamics are then exponential stabilized using time-varying Proportional-Integral (PI) state feedback control law. A TLC based controller can be viewed as the gain-scheduling controller that is designed at each point on the trajectory to provide robust stability.

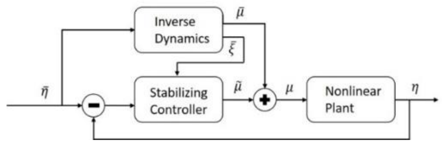

Figure 10. TLC structure

The overall closed-loop system consists of 4 loops as shown in Fig 11, which are the guidance outer and inner loop, and steering outer and inner loop. Each loop employs the TLC structure as shown in Fig 10.

The high speed control algorithm is running on the Quanser's ® HiQ Aero microcontroller with a Gumstix Verdex CPU running at 0.6 GHz. It equipped with a 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer, 1 TTL serial GPS port, and 10 channels PWM output. The real-time control software Quarc generates real-time code for Gumstix Verdex directly from Matlab/Simulink®. The on-board sensor data and controller gains update at 100 Hz, which satisfy the high speed trajectory tracking performance.

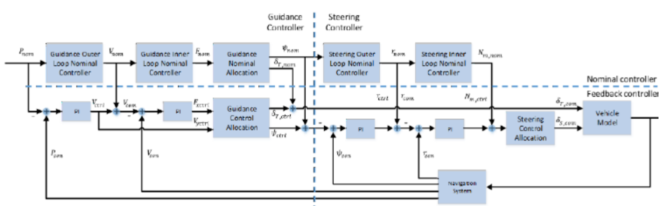

Figure 11. 3DOF Trajectory Linearization Controller Block Diagram.

The guidance subsystem which generates a feasible trajectory for the trajectory tracking controller is running on a NVIDIA TX1 board. The generated trajectory is sending to the HiQ through a serial communication port at 10Hz. Since the different sampling times running on two boards, a low-to-high data interpolation preprocessing is running on HiQ in order to get a smooth trajectory.

The hardware tuning procedure is separated into two steps: the nominal controller tuning and the feedback controller tuning. The dynamics (bandwidth) of the nominal controller is designed based on the desired bandwidth of the open-loop controlled system. The guidelines for designed feedback controller gains are: 1) the closed-loop bandwidth of the innermost-loop is at least three-time smaller than the actuator's; 2) an outer-loop closed-loop bandwidth is at least three-time smaller than its immediate inner-loop; 3) the bandwidth of the outer-most position loop should satisfy position error transient requirement. Here are some trajectory tracking hardware test results.

VIII. Experimental Results

A. Vehicle Subsystem

The finished vehicle is shown in Fig 12 below. It is under 12 kg with a maximum speed of 10 m/s. The E-stop button is on top of the vehicle.

Figure 12. Vehicle Subsystem

B. Electrical and Electronics Subsystem

The finished electrical and electronics panel is shown in Fig 13, along with the wireless E-stop transmitter.

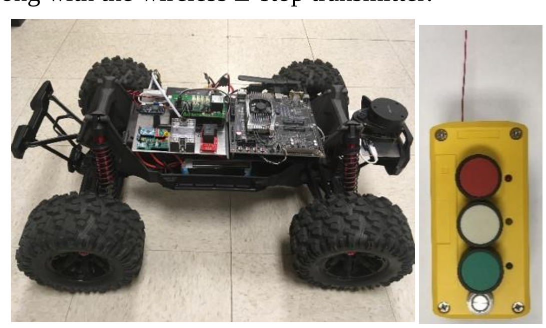

Figure 13. E-E Panel and E-stop Transmitter

C. Navigation (Localization) Subsystem

Fig 14 shows a sensor fusion test result with GPS, INS (accelerometer and gyroscope) and shaft encoder. The result is overlaid to Google Earth map with adequate accuracy. Fig 15 shows the vision subsystem test results with traffic light in difficult lighting conditions, white lines and orange cones, respectively. The detection speed is deemed adequate.

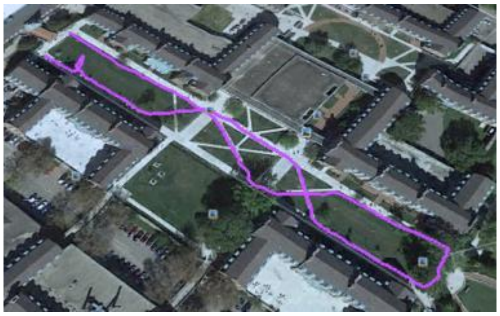

Figure 14. GPS/INS Test result shown on Google Earth

Figure 15. Vision detection (light, cone, line)

D. Cognitive Control Subsystem

Figs 16-17 show the computer simulation of the shortcut completion scenario and a high-speed LOS PPG scenario. The results verified the design with high level of confidence.

E. Motional Control Subsystem

Figs 18-20 show trajectory tracking control hardware test performed on a Traxxas E-MAXX vehicle for a 4-pedal rose-curve trajectory at 2 m/s and circular trajectory at 5 m/s. Since the TLC control algorithm is readily scalable, these results give us high confidence for application on the larger X-MAXX.

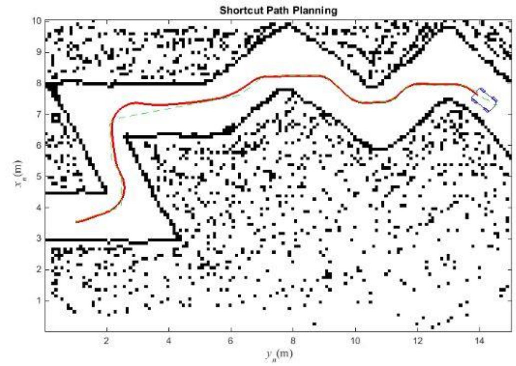

Figure 16. Path Planning in the shortcut.

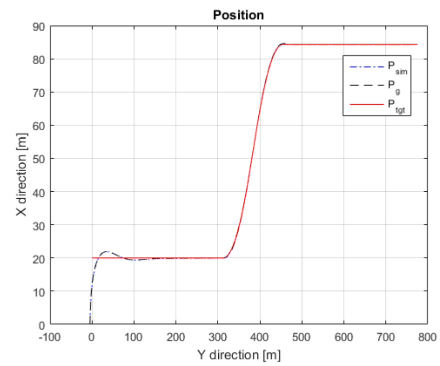

Figure 17. Simulation test of LOS guidance.

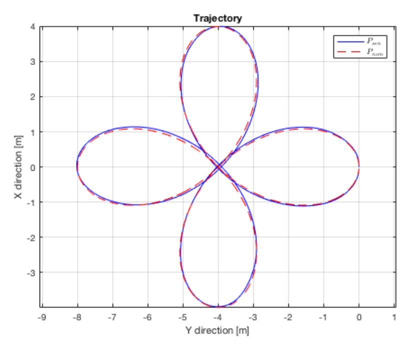

Figure 18. Rose: Vehicle Trajectory

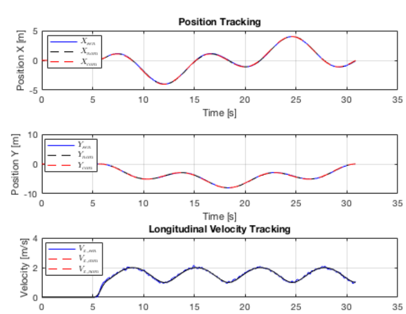

Figure 19. Rose: Position and Velocity Tracking Response

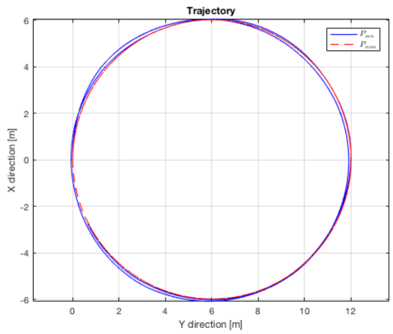

Figure 20. Circle: Vehicle Trajectory

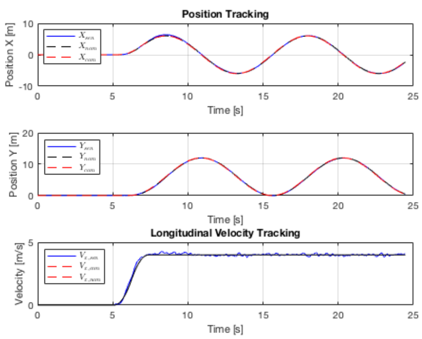

Figure 21. Circle: Position and Velocity Tracking Response

IX. Innovations

This project encompasses several innovations. A novel bio-psychically inspired cognitive autonomous control architecture [6]. The navigation (localization) employs sensor fusion with a Kalman filter to enable inexpensive sensors for accurate and high-speed operations. Machine vision system employs advanced deep-learning neural network algorithms for object identification with good results. The mission trajectory planning system is constructed by using a line-of-sight pure pursuit guidance trajectory generator [1] and a switching control based path planner [3]. A PCT patent application has been filed for the former, and a US provisional patent application has been filed for the latter. A 3DOF Trajectory Linearization Controller for non-holonomic car-like vehicles is implemented for simultaneous and precise high-speed drive and steering control [2] [4]. The control algorithm is based on nonlinear vehicle and tire dynamic models. Therefore it has good scalability and adaptability. An international patent application under the PCT had been filed for the control algorithm.

X. Current Status and Future Work

At this time the vehicle construction is complete, and critical subsystems have been tested. In the remaining time before the competition, integrated tests under the completion scenarios will be conducted.

Acknowledgment

The OU-PAVE team express their gratitude to the Russ Vision grant from the Russ College of Engineering and the School of EECS for funding this project. They would also like to thank Professors Maarten Uijt Haag, Jundong Liu and David Chelberg for their generous support either by providing critical components or their valuable technical insights.

References

[1] Duan, Dagao, et al. "An improved Hough transform for line detection." *Computer Application & System Modeling (ICCASM), 2010 International Conference on*. Vol. 2. IEEE, 2010.

[2] Asano, Tetsuo, and Naoki Katoh. "Variants for the Hough transform for line detection." *Computational Geometry* 6.4 (1996): 231-252.

[3] Sandler, Mark, et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[4] Chen, Yuanyan, and J. Jim Zhu. "Pure Pursuit Guidance for car-like Ground Vehicle Trajectory Tracking." *ASME 2017 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2017.

[5] Chen, Yuanyan, and J. Jim Zhu. "Car-Like Ground Vehicle Trajectory Tracking by Using Trajectory Linearization Control." *ASME 2017 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2017.

[6] J. Zhu and X. Xu, "Biopsychically Inspired Cognitive Control for Autonomous Mobile Agents Based On Motivated Learning," Plenary Presentation, Proceedings, 2012 IEEE International Conference on Methods & Models in Automation & Robotics, Miedzyzdroje, Poland, August, 2012.