2020 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
AUTONOMOUS GROUND SYSTEMS (AGS) TECHNICAL SESSION
AUGUST 11-13, 2020 - NOVI, MICHIGAN

DEVELOPING AUTONOMOUS VEHICLES WITH OPTIMAL SENSOR SELECTION, INTEGRATION & CONTROL IMPLEMENTED AT 2019 INTELLIGENT GROUND VEHICLE COMPETITION (IGVC)

Andrew Kosinski
Mechanical Engineer
US Army GVSC GVR
Warren, MI

Kiran Iyengar
Electrical Engineer
US Army GVSC GVR
Warren, MI

Jane Tarakhovsky (Self-Drive Chair, Volvo/Veoneer)
Jerry Lane (GLS&T, IGVC Co-Founder)
KaC Cheok (Oakland University, IGVC Co-Founder)
Bernie Theisen (US Army GVSC GVR, IGVC Co-Chair)
Sami Oweis (Oakland University, Adjunct Professor)

ABSTRACT

The IGVC offers a design experience that is at the very cutting edge of engineering education, with a particular focus in developing engineering control/sensor integration experience for the college student participants. A main challenge area for teams is the proper processing of all the vehicle sensor feeds, optimal integration of the sensor feeds into a world map and the vehicle leveraging that world map to plot a safe course using robust control algorithms. This has been an ongoing challenge throughout the 27 year history of the competition and is a challenge shared with the growing autonomous vehicle industry. High consistency, reliability and redundancy of sensor feeds, accurate sensor fusion and fault-tolerant vehicle controls are critical, as even small misinterpretations can cause catastrophic results, as evidenced by the recent serious vehicle crashes experienced by self-driving companies including Tesla and Uber Optimal control techniques & sensor selection/integration into these autonomous ground vehicles will be the focus of this technical paper.

INTRODUCTION

The IGVC is a college level autonomous unmanned ground vehicle (UGV) competition that encompasses a wide variety of engineering professions – mechanical, electrical, computer engineering and computer science. It requires engineering students from these varied professions to collaborate in order to develop a truly integrated engineering product, a fully autonomous UGV, where optimal control and sensor selection/integration play a large role in competitor's autonomous vehicle performance and operation.

This industry aligned, vehicle control/sensor selection focus of this competition has been further emphasized over the last few years, with the third Self-Drive Challenge carried out in 2019, requiring vehicles to perform autonomous, street-legal vehicle road operations including lane keeping, lane switch, merging, avoiding crossing obstacles (simulated pedestrians/vehicles), taxi pickup of passengers, simulated pothole detection, stop and crosswalk lines detection, right/left turn and intersection detection/logic, navigation to GPS waypoints and autonomous parking.

Figure 1. Self-Drive Challenge course (Oakland University Incubator, 419 Golf View Ln, Rochester, MI 48309).

Adding a further industry relevant emphasis for autonomous vehicles, a new challenge, the Cyber Challenge, was incorporated in 2019 with the goal to educate & promote knowledge of vehicle cyber security best practices for autonomous, intelligent & smart vehicles. Understanding of the NIST RMF process is a primary objective of this competition and will be given special attention by the judges. Understanding of the NIST RMF process will be demonstrated by a written report describing the process in general, followed by a specific case study using either a provided or novel threat concept applied to a specific vehicle. An oral presentation will be delivered during the IGVC competition and will demonstrate team understanding of the NIST RMF process as well as how it was applied to the choice, design, and implementation of cyber controls for team robots specific to chosen threat scenario.

With regards to general autonomous vehicle control, potential methods of control algorithms that could be applied to military platooning convoys involve path planning and maneuvering command. These are both critical steps in autonomous driving vehicle systems. Path planning is on the rising edge for robotics control, feedback monitoring is the next step of planning and confirmation to command assignment, adding the control to smooth the projected path is state of the art in robotics control, and it has many advantages; such as smooth cornering and curve/ramp handling. Robotics tracking is a noticeable advancement that can be achieved in many ways, by utilizing simulation to report locations and position and orientation. Robotics tracking has undergone noticeable advancements and can be achieved in many ways including Attitude Stabilization Control of robotic systems which can be achieved using specific control methods, as will be more extensively discussed in this technical paper.

Teams are required to document their approach to sensor selection, sensing, processing and vehicle control algorithms in their design report each year which is evaluated by a panel of industry judges with extensive automotive/autonomous vehicle knowledge/experience. Each student team provides a documented design report which will be used as the primary references of this paper. Below is an example of the Host University discussion on sensors and controls. This paper will address other university sensor & control approaches and map the team's performance in the IGVC autonomous driving challenges.

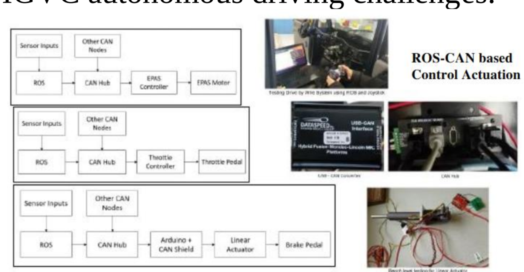

Figure 2. Oakland University's Self-Drive Vehicle Control Diagram.

Figure 2 explains the high level vehicle control scheme for their Self-Drive vehicle, which required autonomously operating a street legal

Polaris GEM 2 electric vehicle through drive-by-wire modifications to the existing chassis to control vehicle actuation (steering, throttle and brake). An existing Dataspeed Advanced driver-assistance systems (ADAS) kit was used, with the setup of a laptop computer running Robot Operating System (ROS) controlling a CAN hub which then fed commands to control the throttle and to actuate the steering wheel/brake pedal.

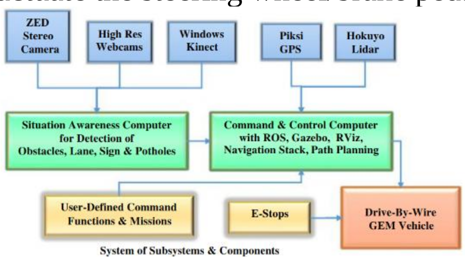

Figure 3. Oakland University's Self-Drive Vehicle Sensor Functions Diagram.

As shown in Figure 3, sensors used included a ZED stereo camera, webcams, Windows Kinect, GPS and Hokuyo Lidar, utilizing existing ROS packages for processing sensor feeds. The camera and Kinect were used for detection of obstacles, lane marking, signs and potholes, fusing with the GPS/Lidar information for overall vehicle navigation/path planning decisions. This system provides for increased robustness, as multiple sensors are supporting similar autonomous vehicle functions such as obstacle detection, which is critical for the very high reliability demands for autonomous vehicle systems, even at lower speeds (5mph max allowable speed during IGVC competition).

Exact correlation of performance to sensor/control approaches will not be guaranteed due to many other competition factors at time of runs but an overview perspective will be provided together with university references for further collaboration on individual techniques.

Section 1. Vehicle Machine Vision – Sensor Selection/Processing/Integration

Vehicle machine vision is a huge part of a successful autonomous vehicle, as the vehicle is completely on its own while operating in the various relevant applicable environments. As mentioned above, teams normally use mono/stereo cameras and LADAR. Component redundancy is important, even more-so with regards to sensors, with some teams adding multiple cameras for redundancy as well as to increase the sensors' field of view for detection. Teams have also installed planar LADARs on pan-tilt assemblies to allow for 3-D sweeping detection. 3-D sweeping is especially important for detecting negative obstacles, like potholes.

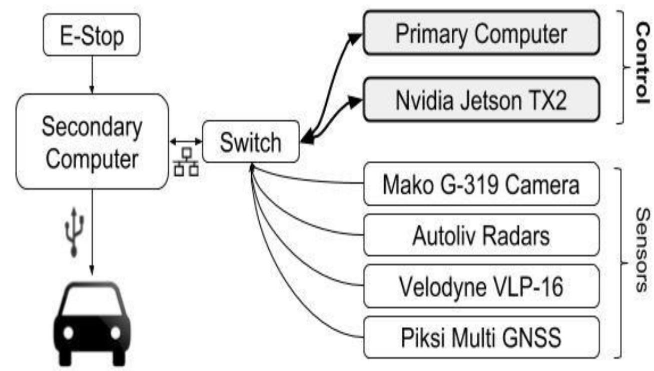

Figure 4. Lawrence Technological University's Self-Drive vehicle safety/processing/sensor overview schematic.

A significant sensor challenge is not just processing and analyzing a sensor's data feed, but then integrating it with the other vehicle sensors to build a coherent world map of the vehicle's environment. Normally simultaneous localization and mapping (SLAM) algorithms are used for this purpose. SLAM also serves as a good redundancy to the data pulled from the vehicle's high precision differential GPS.

This then immediately ties into requiring robust software coding, building in a comprehensive ruleset to be able to segment out irrelevant data and filter noise, as well as segment and recognize important parts of the world map corresponding to obstacles (barrels, potholes, ramps) and other items of interest (flags, spray painted course boundary lines, etc.). In addition to categorizing these items, there needs to be further logic with regards to flags and spray painted course lines.

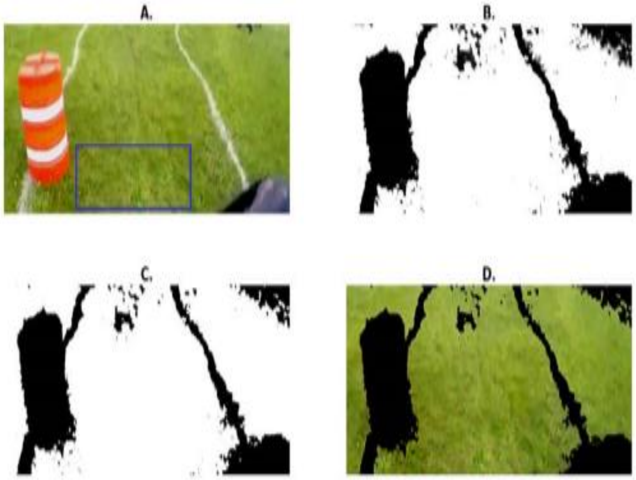

Figure 5. Stony Brook University vehicle camera extracted histogram projection.[1]

The logic for spray painted lines is straightforward, to have the vehicle stay between the two boundary lines. The logic for flags is more involved, requiring the machine vision system to first not only detect the flags, but accurately determine their color (red or blue), and then after knowing the color, program the vehicle to stay to the left of the red flags and to the right of the blue flags.

Sensor noise can become extremely problematic, requiring implementation of additional processing techniques, such as the Oakland University team's application of an Artificial Neural Network (ANN) to assist in the determination of the white course boundary lines. Using self-learning approaches can be very helpful in situations like this, where hard coding white line extraction algorithms that will be applicable in real-life IGVC implementation become challenging. The ANN white line detection process the Oakland University team used is characterized below:

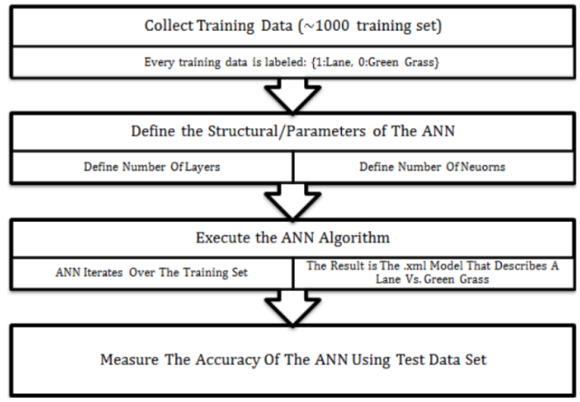

Figure 6. Oakland University Team's ANN White Line Detection Process.[2]

Section 2 Optimal Vehicle Control Through Simulation/Real-Life Testing

Optimizing vehicle control through testing of the vehicle is critical and it can take the form of real-life testing and/or simulation. See below for a mock IGVC course created by the Indian Institute of Technology Bombay team for vehicle testing/evaluation:

Figure 7. Indian Institute of Technology Bombay mock IGVC course.[3]

An obvious advantage control algorithm refinement though the use of simulation over real-life testing is that the vehicle can be worked on while evaluating its (virtual) performance on a computer. An obvious drawback to simulations is that it is only as good as the input data, simplifying assumptions, etc. Another advantage of a simulation is that the (virtual) vehicle can be evaluated many times faster than real-time.

The University of New South Wales (UNSW) team's simulation environment allowed for the simulation to be run up to 5 times faster than real-time and in parallel. The advantages of this can be extreme, assuming wise creation of the simulation environment as a whole and informed determination of the necessary input data, simplifying assumptions, control algorithms, etc., to ensure a highly accurate representation of the real-life vehicle conditions/environment/operations. This can allow for a huge scaling in the amount of vehicle testing within a timeframe, which can greatly improve overall vehicle operation/performance in future real-life testing and at the actual IGVC competition.

Obviously huge amounts of data are generated from these virtual vehicle runs, which then necessitates quick/accurate analysis in order to be useful. For this purpose, the UNSW team developed and incorporated several tools to "automatically analyze and collect statistics regarding the performance in a simulated run of the competition. These statistics, which include average speed, localization error, and proximity to obstacles, allow for quick tuning and verification of parameters to determine which combination of these parameters optimizes the performance of the system as a whole."[4]

The California State University, Northridge (CSUN) team developed their simulation program using LabVIEW. As they state, "The simulation was developed as a method to allow testing of new codes without endangering the vehicle with a previously untested code, which may have bugs that create unsafe conditions for El Toro...Virtual LRF (laser range finder) data is created, while inducing specified levels of Gaussian white noise to more realistically represent the stream of data that would come from the sensors. This allows the vehicle to choose different paths each time it navigates through the simulation. The simulated data gathered by the LRF and compass is passed to the navigation and system integration code, allowing the vehicle to run autonomously."[5]

The Gazebo simulation environment is especially popular with IGVC teams as can be seen below:

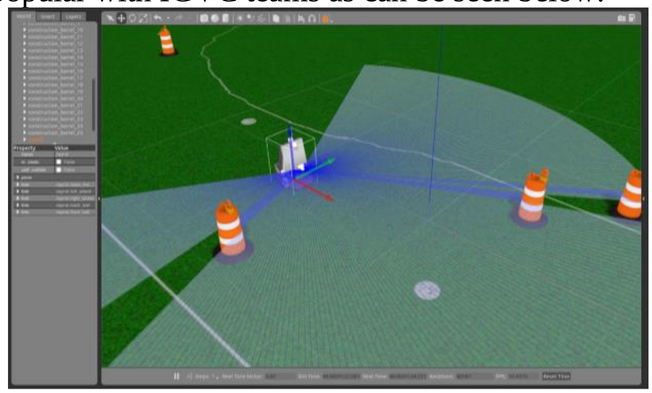

Figure 8. Georgia Institute of Technology Gazebo simulation.[6]

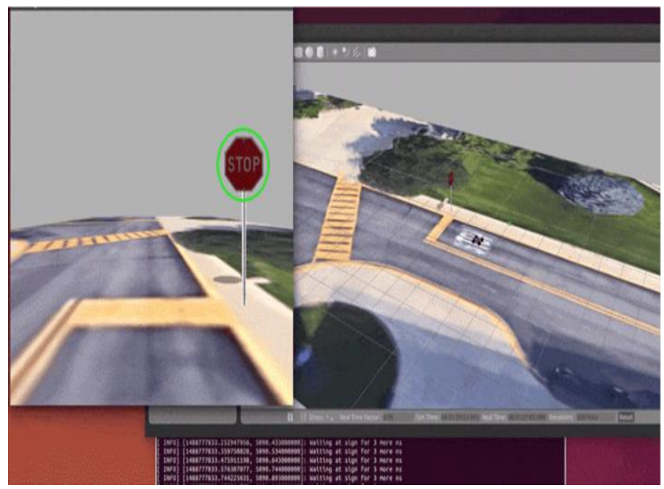

Figure 9. LTU Self-Drive vehicle simulation testing with relevant simulated environment (road signs, lane markings, etc.)[7]

The real-life improvements of a system, such as for these IGVC vehicles, from utilizing effective simulations that feed the optimization of vehicle control algorithms cannot be overstated, especially with the growing virtual toolset for improved simulation, analysis and optimization of real-life system performance. Such toolsets include optimization routines such as neural networks and evolutionary systems, as well as deep learning, which was displayed in a limited, though dramatic degree, with regards to a virtual tool (deep learning computer program AlphaGo) quickly optimizing its performance of the game of GO, as well as the vast improvements demonstrated by later versions of the deep learning software in reduced time periods (AlphaGo Master/Zero). Deep learning has expanded into many fields including speeding up drug analysis/discovery, self-driving vehicle control/behavior optimization, additional "game playing" applications such as OpenAI beating the best Dota 2 team in 2019, artificial general intelligence, etc.

Section 2.1 below provides a novel stabilization controller implementation applied to a lunar lander, of which can be generalized to a full range of unmanned vehicle applications.

Section 2.1 Attitude Stabilization Controller of a Lunar Lander

Lunar Modular Stabilization of its control system is part of its module guidance, navigation, and control system. Lunar Module Guidance, Navigation, and control system designed to control vehicle attitude and translation about or along all axes during a lunar module mission. [15]

3 Coordinate Systems Utilized

2.1.1 Coordinate Systems

The inertial coordinate system is defined such that the X-Y plane forms the landing surface and Z-axis is perpendicular to the X-Y plane and upwards. All the rigid-vehicle and landing-gear-footpad equations of motion are expressed. [15]

The gravity coordinate system is shown in Figure 10. The Coordinate Transformation (T_{IG}) from inertial system to gravity is:

$$[T_{IG}] = \begin{matrix} 1 & 0 & 0 \\ 0 & cos\boldsymbol{\alpha} & sin\boldsymbol{\alpha} \\ 0 & -sin\boldsymbol{\alpha} & cos\boldsymbol{\alpha} \end{matrix}$$

(1)

The Body coordinate system is fixed in the landing vehicle so that the origin concurs with the idealized-rigid-vehicle center of mass. The Z_B -axis is directed upwards and parallel to the vehicle vertical center line. The body system is related to the inertial system by the set of Euler angles θ_x , θ_y , and θ_z . The transformation from body angular rates to Euler angular rates is given by:

$$[TBE] = \begin{matrix} \cos\theta_z / \cos\theta_y & -\frac{\sin\theta_z}{\cos\theta_y} & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ -\sin\theta_y \cos\theta_z / cos\theta_y & \sin\theta_y \sin\theta_z / cos\theta_y & 1 \end{matrix}$$

(2)

After transformation from Euler angular rates to body angular rates is done, we arrive at:

$$\begin{matrix} \dot{\theta}x \\ \dot{\theta}y \\ \dot{\theta}z \end{matrix} = [T_{BE}] = \begin{matrix} \boldsymbol{\Psi}x \\ \boldsymbol{\Psi}y \\ \boldsymbol{\Psi}z \end{matrix}$$

(3) [15]

2.1.2 Newton's Laws of Motion

The three translational equations of motion of the rigid-vehicle center of mass are obtained by the summation of all forces acting on the rigid vehicle. This is shown by the following which applies the Newton's Laws of Motion:

$$\ddot{X} = \frac{\mathrm{FX}}{\mathrm{m}} + g_x$$
$$\ddot{Y} = \frac{\mathrm{F}Y}{\mathrm{m}} + g_y$$
$$\ddot{Z} = \frac{\mathrm{F}Z}{\mathrm{m}} + g_z$$

(4) [15]

FX, the FY, FZ is summation of forces on the idealized rigid vehicle resolved along the X-, Y-, and Z-axes, respectively. M is the mass of the idealized rigid vehicle. $\ddot{X}, \ddot{Y}, \ddot{Z} =$ inertial accelerations of rigid-vehicle center of mass. g_x, g_y, g_z are components of the gravitational acceleration vector expressed in the inertial coordinate system [15]

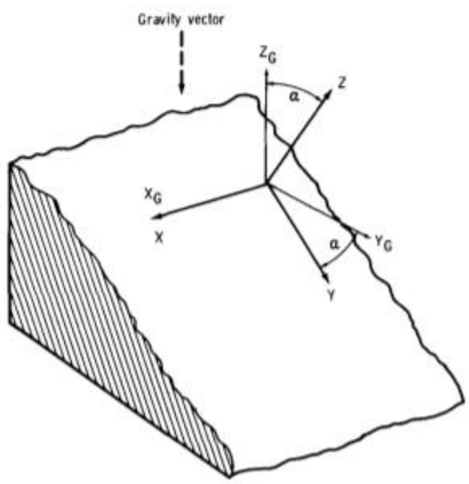

Figure 10. Top view diagram of gravity coordinate system[15]

2.1.3 Engine Thrust and Nozzle-Crushing Forces

Certain spacecraft landing procedures may cause the descent-stage rocket engine thrust forces being present during touchdown. The form the thrust tailoff curve takes is a function of the time at which the command is given to terminate engine thrust, electrical-mechanical delays in the engine systems once the command to terminate engine thrust is given, and the thrust tail-off properties of the particular rocket engine. The landing procedure used for the Lunar Module is one where a commanded descent rate is given such that the vehicle approaches the landing site at constant velocity. [15] T_1 is the time of rocket engine thrust termination. T_p = represents the time of the position on the thrust curve when the footpad touches the surface. The thrust curve in figure 13 is approximated in the range from t_0 to t_2 by constant F_1 and from t_2 to t by an exponential function. The thrust force is given by the equation:

$\overrightarrow{FTHRUST} = F_1$ $t_0<t<t_2$ and FTHRUST= $F_1\ e^{-K(t-t_2)} + M_0 + M_1(t-t_2) + M_2(t-t_2) + M_2\ (t-t_2)^2 + \dots$ $t > t_2$

(5) [15]

where F_1, K, M_0, M_1, etc. are determined by fitting the appropriate function to the developed rocket engine thrust tailoff data.

The total thrust force is given by: $\overrightarrow{TAF}$ = AMP $(\overrightarrow{FTHRUST})$ (6)

Here, $\overrightarrow{TAF}$ is directed through the vehicle center of mass
The total force vector due to the engine thrust and nozzle crushing load is shown by

$\vec{F}_{TF} = \overrightarrow{TAF} + \overrightarrow{FFL}$ (7)

where $\overrightarrow{FFL}$ is the nozzle crushing load vector [15]

.

2.1.4 Forces and Moments from Reaction Control Systems

The 3 basic modes of the control system operation are considered, including attitude hold, rate command, and downward translation. The control system applies to either a positive or negative torque of fixed magnitude to the X_B, Y_B, Z_B axes depending on the value of a linear combination of attitude and attitude rate errors associated with the respective body axis. The attitude rate errors are as follows:

$$E\omega_x = \omega_x - C\omega_x$$
$$E\omega_y = \omega_y - C\omega_y$$
$$E\omega_z = \omega_z - C\omega_z$$

(8) [15]

$C\omega_x$, $C\omega_y$, and $C\omega_z$ are specified commanded attitude rates. The commanded values for the Euler angles are specified by data input and are denoted by : $C\theta_x, C\theta_y$,and

$C\theta_z$. [15] The deviations in Euler angles from the commanded values are computed from the following:

$$D\theta_x = \theta_x - C\theta_x$$
$$D\theta_y = \theta_y - C\theta_y$$
$$D\theta_z = \theta_z - C\theta_z \qquad (9)$$

The attitude errors for the body system axes are given by:

$$\begin{matrix} E\theta_x \\ E\theta_y \\ E\theta_z \end{matrix} = [T_{EB}] \begin{matrix} D\theta_x \\ D\theta_y \\ D\theta_z \end{matrix} \qquad (10)$$

For the attitude-hold mode, the following error equations are evaluated:

$$EX = k_{11}E\theta_x + k_{12}E\omega_x$$
$$EY = k_{21}E\theta_y + k_{22}E\omega_y$$
$$EZ = k_{31}E\theta_z + k_{32}E\omega_z$$

(11) [15]

The values $k_{11}, k_{12}, k_{21}, k_{22}, k_{31}$, and k_{32} are constants, given by data input, which then model a given control mode and additionally the commanded attitude rates are 0. Additionally, the errors EX, EY, and EZ are compared to upper and lower bounds given by data input to determine if RCS (Reactive Control Systems) thrusters should fire to produce a torque on the given axis. [15]

2.1.5 Sub Control System (SCS)

SCS provides the backup guidance system that would permit attainment of a safe lunar orbit if primary guidance were lost. It forms an integral part of both the primary and abort subsystems. [16]

2.1.6 Stability Criteria

The performance evaluation of a lander requires the simulation of many touchdown conditions. When a large number of touchdown simulations are required, the speed at which the computer can execute these simulations becomes a factor. In the evaluation of a soft-landing system for stability (tipover) performance, the computer running time can be reduced by defining stability criteria such that the computer can make the decision as to whether the landing being simulated will eventually be stable or unstable. The length of vector $\overrightarrow{SD}$ is stability distance. This is calculated by the equation:

$$\vec{SD} = \vec{ACM} * \vec{G} X \vec{AB} / |\vec{G} X \vec{AB}|$$

(12)[15]

$\vec{AB}$ is the vector connecting two adjacent landing gear footpads. $\vec{ACM}$ is the vector connecting the rigid vehicle center of mass and a landing gear footpad. $\vec{G}$ represents the gravity vector. **A** stability distance is computed by using the stability distance vector for each stability wall which is given by the following:

$$\vec{SD} = SD(\vec{G} X \vec{AB} / |\vec{G} X \vec{AB}| \qquad (13)$$

[15]

2.2 Net Forces and Torque Vectors

The total force vector (without gravity forces) acting on the idealized rigid vehicle center of mass is the sum of individual external force vectors and is given by the equation:

$$\vec{F} = \begin{matrix} FX \\ FY \\ FZ \end{matrix} = \overrightarrow{FCMG} + \overrightarrow{FTFI} + \overrightarrow{FRCSI}$$

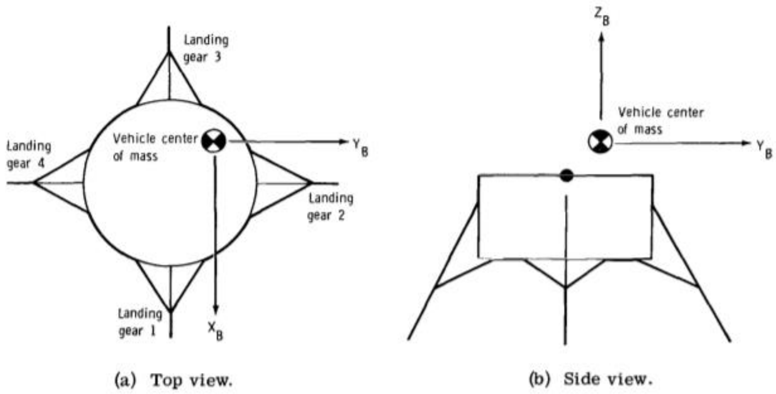

Figure 11. Top and Side Views of the lander[15]

The total force vector (without gravity forces) acting on the idealized rigid vehicle center of mass is the sum of individual external force vectors and is given by the equation:

$$\vec{F} = \begin{matrix} FX \\ FY \\ FZ \end{matrix} = \overrightarrow{FCMG} + \overrightarrow{FTFI} + \overrightarrow{FRCSI}$$

(14)

The net vehicle forces FX, FY, and FZ are used in union with equations (4) to calculate the vehicle translational and rotational accelerations.
[15]

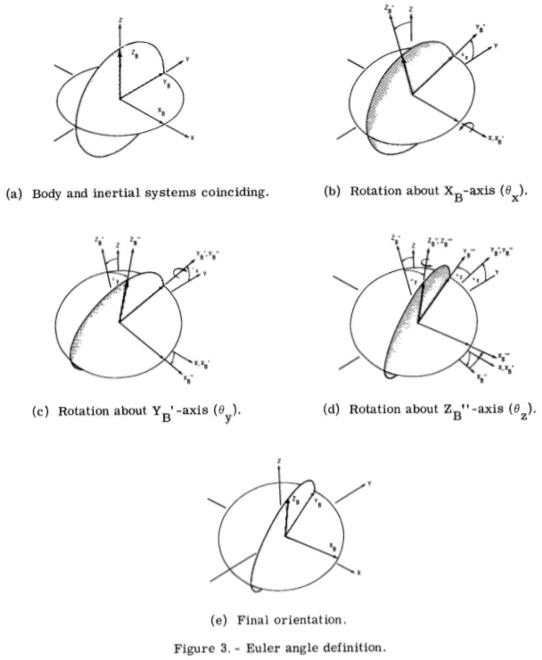

Figure 12: Orientation of the body coordinate system in the rigid vehicle [15]

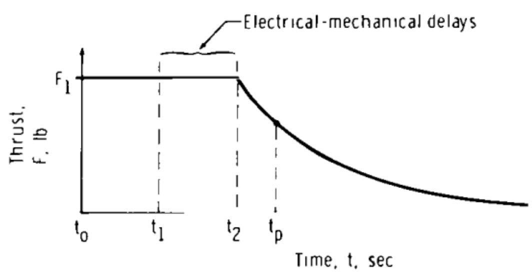

Figure 13 Rocket thrust engine characteristics [15]

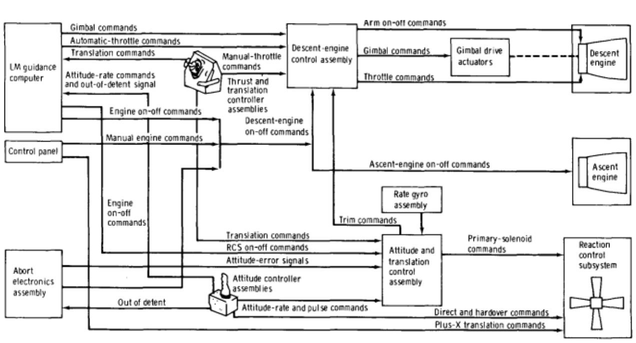

Figure 14. block diagram of the SCS[16]

CONCLUSION

The 2019 IGVC was a successful autonomous ground vehicle competition which further developed team capabilities in the very industry/Military relevant engineering skillset areas of optimal sensor selection, integration & control through the creation and evaluation of functional autonomous vehicles capable of real-world navigation. Teams gained valuable engineering experience which will benefit them in their future careers. The refinement of the Self-Drive Challenge proved a success in providing a highly industry/Government relevant street legal vehicle competition which further develops the necessary skills engineers should have in the growing fields of autonomy, AI, machine learning, self-driving vehicles, etc. The control and mathematical foundations of aerial and ground vehicles has a lot of relevance in the fields of sensor fusion, stabilization, navigation, etc. Additionally, we can start to understand how different types of robotic systems relate to each other by a further in depth mathematical analysis of the systems.

REFERENCES

[1] Stony Brook University Design Report. 2016. http://www.igvc.org/design/2016/22.pdf.

[2] Oakland University Design Report. 2015. http://www.igvc.org/design/2015/16.pdf.

[3] Indian Institute of Technology Bombay Design Report. 2017. http://www.igvc.org/design/2017/8.pdf.

[4] University of New South Wales (UNSW) Design Report. 2015. http://www.igvc.org/design/2015/24.pdf.

[5] California State University-Northridge (CSUN) Design Report. 2015. http://www.igvc.org/design/2015/4.pdf.

[6] Georgia Institute of Technology Design Report. 2017. http://www.igvc.org/design/2017/5.pdf.

[7] Lawrence Technological University Research Day 2018 Presentation. 2018. https://www.ltu.edu/provosts_office/research_day.asp

[8] J. Park and B. Kuipers, "A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment," 2011 IEEE International Conference on Robotics and Automation (ICRA), May 2011, pp. 4896-4902. W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[9] Olson, Edwin. "A primer on odometry and motor control." (2004).

[10] J. Park and B. Kuipers, Autonomous Person Pacing and Following with Model Predictive Equilibrium Point Control, 2011.

[11] Cheok, KaC, Kiran Iyengar, and Sami Oweis. "Smooth Trajectory Planning for Autonomous Leader-Follower Robots.*" Proceedings of 34th International Confer 58 (2019): 301-309.*

[12] P. Foster, Z. Sun, J. Park and B. Kuipers, VisAGGE: Visible Angle Grid for Glass Environments *LIDAR mapping for glass, metal, mirrors, and other non-diffuse surfaces, 2011.

[13] K. Iyengar, "Smooth control path planning and tracking of an autonomous mobile robot in real-time," M.S. thesis, School of Eng. And Computer Sc., Oakland Univ., Rochester, Michigan, USA, Aug. 2015. Accessed on: May. 28, 2018.

[14] Hargadine, Calvin S. Mobile robot navigation and obstacle avoidance in unstructured outdoor environments. Diss. Monterey, California: Naval Postgraduate School, 2017.

[15] Doiron, H. H., and G. A. Zupp Jr. "A mathematical procedure for predicting the touchdown dynamics of a soft-landing vehicle." (1971).

[16] Shelton, D. Harold. "Apollo experience report: Guidance and control systems. Lunar module stabilization and control system." (1975).