Autonomous Design Report

Nikolaos Adamopoulos Antonios Doukakis Aris Koumplis Dimitrios Papailiopoulos Evangelos Poutos
Emmanouil A. Serlis Paris Stentoumis Vasileios Vrettos

***Abstract*—This research paper presents the autonomous system pipeline of P23, the first driverless race car developed by PROM Racing. The hardware components, perception pipeline, velocity estimation, localization & mapping, motion planning, and safety measures of P23's autonomous system are outlined.**

***Keywords*—** Autonomous Vehicle, Formula Student Driverless, Perception, Velocity Estimation, SLAM, Motion Planning, Safety

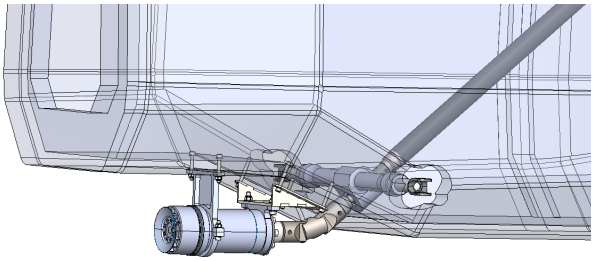

Figure 1: Steering Actuation Mechanism

1 Introduction

This research paper presents the autonomous system pipeline of P23, the first race car with Driverless (DV) capabilities developed by PROM Racing. The objective of P23 is to successfully complete all DV disciplines in the 2023 season Formula Student (FS) competition format, while simultaneously laying the groundwork for future performance-focused iterations.

Throughout the design, development, and testing phases of P23, two significant constraints had to be addressed. Firstly, the project commenced in early 2022 without any prior expertise in the field, leaving limited time to accumulate relevant knowledge and skills. This lack of experience necessitated an efficient and accelerated learning process, which involved studying other teams' approaches and reviewing scientific standards to ensure that P23 met the ambitious goals set for its autonomous system pipeline. Secondly, as PROM Racing aimed to maintain sustainability, limited resources were allocated to the high-risk DV project. This limitation demanded careful resource management and innovative solutions to overcome said financial barriers.

The paper is organized as follows: Section 2 elaborates on the powertrain, actuators, sensors, and computing system featured in P23. Section 3 describes the system architecture, including aspects related to functional safety and cyber security. Section 4 outlines the perception pipeline implemented in P23. Section 5 focuses on the application of a velocity estimation filter within the autonomous system pipeline. Section 6 delves into the localization and Simultaneous Localization And Mapping (SLAM) modules. Section 7 details the motion planning process employed by P23. This encompasses a geometric path planner, followed by two variants of kinodynamic path followers.

2 Hardware

2.1 Powertrain

P23 is an electric rear-wheel drive vehicle equipped with an Emrax 228 HV CC motor, which is constrained to a maximum power output of 80 kW in adherence to the FS regulations. The motor is controlled by a Unitek Bamocar PG-D3 700V motor controller. Additionally, P23 is equipped with a limited slip differential to enhance traction and handling capabilities.

2.2 Actuators

The steering actuation system consists of a Maxon EC 60 Flat 200W brushless DC (BLDC) motor with a nominal torque of 0.536 Nm at 3240 rpm, in conjunction with a Maxon GP 52C planetary gearbox featuring a 66:1 reduction ratio. This motor and gearbox combination achieves a system performance of 26.5 Nm of torque after accounting for losses, surpassing the maximum steering torque required during worst-case vehicle conditions, calculated to be 22 Nm. The selected configuration achieves a lock-to-lock time of 0.7 seconds during nominal operation. Backlash arising from connections between the motor, the pinion gear, the universal joint, and the steering rack is minimized through position control implemented at the steering rack displacement rather than the motor angle. Feedback is provided by a linear potentiometer that directly measures the rack displacement.

Regarding the service braking system, a brake-by-wire approach is employed using a similar, albeit less powerful, BLDC motor and gearbox combination. Specifically, a Maxon EC45 Flat 120W motor with a nominal torque of 0.174 Nm at 4520 rpm was selected, along with a Maxon GP 42C planetary gearbox featuring the same 66:1 reduction ratio as the steering actuation system. This combination enables the system to deliver 8.6 Nm of torque, which exceeds the torque required to lock the brakes, calculated to be 7 Nm. In this case, a straightforward torque control strategy is employed, where the controller maps the motor torque to the desired brake pressure.

2.3 Sensors

2.3.1 Perception Sensors

Perception sensors play a critical role in detecting, identifying, and measuring the distance from cones, which serve as track boundaries. Within the automotive industry, four types of sensors have gained widespread usage and are worth considering for our application: ultrasonic, Radar, LiDAR, and cameras.

Firstly, ultrasonic sensors provide limited range, typically less than 5.5 meters, and do not provide information about the color of the cones [11]. Consequently, they were deemed unsuitable for both cone identification and distance measurement.

Secondly, Radar sensors, while also unable to provide information about cone color [11], additionally struggle with accurately detecting low dielectric constant objects [5], such as plastic cones, making them unfit for our specific application.

One commonly used perception sensor is LiDAR, which offers satisfactory range and highly accurate depth estimation [11]. Consequently, LiDAR is ideal for a sensor fusion setup, where it can be combined with another sensor capable of providing useful information about cone color. However, the main drawback of LiDAR is its steep price, which is expected to change with the introduction of Micro-Electro-Mechanical Systems (MEMS) technology. As

newer models enter the mass production phase and become available at competitive prices, it is part of our immediate future plans to acquire a MEMS LiDAR sensor.

The last remaining type of perception sensors are the cameras, with P23 featuring two monocular cameras spanning a combined field-of-view of 105 degrees. Alternative options, such as Time-of-Flight (ToF) or structured-light cameras, were considered but found to offer limited range, typically less than 7.5m, and high sensitivity to other light sources [7], while stereo-vision setups require twice the sensors compared to our monocular setup for the same field-of-view.

2.3.2 Velocity Estimation Sensors

Velocity estimation sensors serve the purpose of estimating the vehicle's velocity and acceleration. P23 is equipped with a dual-antenna GPS-aided Inertial Navigation System (INS), a separate Inertial Measurement Unit (IMU), and a custom-made Hall effect wheel-speed sensor for each of the four wheels. While further improvements in accuracy could have been achieved by incorporating a non-contact ground speed sensor, the associated cost outweighed the relative gain in accuracy.

2.4 Computing System

Due to our limited experience in the field of autonomous vehicle development, accurately estimating the requirements for our computing system posed a significant challenge. In light of this constraint, we prioritized high configurability to ensure adaptability to evolving needs. Our current computing system incorporates components commonly found in desktop PCs, including a motherboard, CPU, RAM, SSD, and Tensor Processing Unit (TPU), which can be easily replaced if deemed inadequate.

Notably, we opted for a unique configuration by employing a TPU in conjunction with the CPU's integrated graphics processor to handle the computational demands of the perception pipeline. This innovative approach offers several advantages, including a substantial reduction in power consumption of over 70W compared to a traditional GPU setup with negligible sacrifices in accuracy.

Part Title	Vendor	Part Name
Motherboard	Asrock	IMB-X1233-WV
CPU	Intel	i7-12700
RAM	Corsair	Vengeance 2x16GB 3.2GHz
SSD	Western Digital	Black SN750 1TB
TPU	Google (Coral)	M.2 Accelerator

Figure 2: List of components used in the computing unit

3 System Architecture

Each of the modules mentioned above is designed as a separate process in the computing unit. The communication between each process is achieved via the Robot Operating System (ROS2 Humble) [10], where each module is developed as a ROS node with managed lifecycle. Communication with the rest of the vehicle is achieved via serial protocol with a custom interface from ROS messages to CAN bus style messages. A supervisor node is responsible for managing all the system modules and synchronizing perception and velocity measurements.

Moreover, a heartbeat-based mechanism is implemented into the supervisor node to ensure the operation of all other modules. In the event of a module ceasing to function due to an unhandled error, the manager initiates its shutdown. If the failed module is responsible for controlling the actuators or facilitating communication between the Vehicle Control Unit (VCU) and the Autonomous System, the system enters an AS Emergency state. For all other modules, an attempt is made to recover while simultaneously bringing the vehicle to a safe operational state.

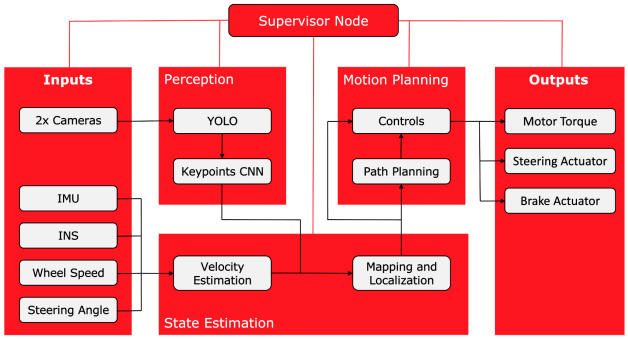

Figure 3: Driverless System Pipeline

Regarding cyber security, protection against man-in-the-middle (MITM) attacks is achieved by implementing a whitelist approach, which restricts connectivity to the private network of the computing system. Only specific MAC addresses associated with devices owned by authorized team members and utilized for the telemetry system are permitted to establish connections. This measure ensures that communication within the network remains secure and mitigates the risk of unauthorized access or tampering.

4 Perception

The primary objective of the perception module is to utilize information captured by the monocular cameras to detect cones, identify their colors, and accurately measure the distance between them and the vehicle.

The initial tasks of cone detection and color identification are accomplished through the implementation of the You Only Look Once (YOLO) neural network [14]. YOLO is a cutting-edge, real-time object detection network that takes an RGB image as input and generates bounding box coordinates and cone color information as output. In our current setup, we utilize the YOLOv5n variant with int8 precision to facilitate inference on the TPU.

YOLOv5n was trained on data from the FSOCO dataset [17]. It achieved a mAP of 76.1%, however, the percentage is significantly higher for cones within a range where distance estimation is accurate with the current pipeline.

Measuring the distance between the vehicle and the cones can also be achieved using information extracted solely from the YOLO output. Given that the cone dimensions are known a priori, information about the associated bounding box, such as its width and height, can be leveraged to estimate the distance between the cone and the camera. However, due to the imperfect fit of the bounding box on the cone as well as the cone's inherent lack of symmetry, any measurements that use this method are inaccurate.

An alternative method [12] entails detecting several distinctive points on the image of each cone whose relative positions in the 3D world are known a priori. Given that for these points both their image coordinates and their 3D position, for any point on the cone being selected as the origin, are known, a Perspective-n-Point problem can be formed, where n is the number of distinctive points on each cone.

To detect these distinctive points, a custom convolutional neural network was designed that outputs the x and y coordinates of seven keypoints, as depicted in the picture. It was trained on a custom dataset using the Supervisely annotation tool, which was subsequently augmented by rotating, cropping, jittering, and gamma compressing or expanding existing images.

The Perspective-n-Point problem is then solved using the *solvePnP* function from the OpenCV library [3]. The resulting translation vector is subsequently rotated and translated to derive the accurate distance of the cone from the center of mass of the vehicle. By employing this method, more precise distance

Figure 4: The stages of the perception pipeline: original image, bounding boxes provided by YOLO, keypoints provided by custom CNN (from left to right)

measurements can be obtained, surpassing the limitations of the bounding box-based approach.

The resulting perception pipeline achieves a Root Mean Squared Error (RMSE) of less than 0.01 rad on measurements of the angle at which the cones are spotted. In terms of distance measurement, the pipeline achieves an RMSE of 0.20m for cones located within a range of 5m and 0.58m for cones within the range of 5m-10m

5 Velocity Estimation

The objective of the velocity estimation module is to continuously provide an estimate of the longitudinal velocity u_x, the lateral velocity u_y, and the yaw rate r of the vehicle. These estimations serve as inputs to the localization/SLAM, velocity planning, and controls modules.

The available sensor suite comprises an IMU, INS, four wheel encoders, and onboard measurements such as the steering angle. To fuse these measurements effectively, an Extended Kalman Filter (EKF) is employed. The EKF was chosen over slightly more precise alternatives, such as the Particle Filter, due to its low computational cost, its ease of implementation and the abundance of documentation available.

The state vector of the filter consists of the longitudinal and lateral velocities, yaw rate, and their respective accelerations. The EKF utilizes a kinematic bicycle model, as shown below:

$$\begin{aligned}\dot{\vec{v}} &= \vec{a} - r \times \vec{v} + q_v \\ r &= a_r + q_r \\ \dot{\vec{a}} &= q_a \\ \dot{r} &= q_{a_r}\end{aligned}$$

where q_* are the independent process noises of the system. Additionally, the measurement prediction is computed based on the predicted state vector using the following equations:

$$\begin{aligned}\vec{u}_{ins} &= \vec{v} + r \times \vec{x}_{ins} + \varepsilon_{ins} \\ r_{imu} &= r + \varepsilon_{imu,r} \\ \vec{a}_{imu} &= \vec{a} + r \times \vec{x}_{imu} + r \times (r \times \vec{x}_{imu}) + \varepsilon_{imu,a} \\ \omega_f &= \frac{1}{R_{wheel}} [u_x \cdot cos(\delta) + v_y \cdot W_f \cdot sin(\delta)] + \varepsilon_w \\ \omega_r &= \frac{v_x}{R_{wheel}} + \varepsilon_w\end{aligned}$$

where the measurement noises ε_* are assumed to be independent for each sensor used in the system.

To enhance the robustness of the system, the Kalman filter is designed to operate in the presence of sensor failures. It incorporates a partial update mechanism by utilizing measurements received during the previous time step, ensuring redundancy and maintaining estimation accuracy. Furthermore, the probabilistic nature of the filter enables the detection of outlier measurements by computing the Mahalanobis distance from the predicted measurement vector, facilitating their exclusion from the update process [16].

The combination of these techniques results in highly accurate vehicle state estimation, even in the event of sensor failures, while maintaining a high update frequency due to the computational simplicity of the algorithm.

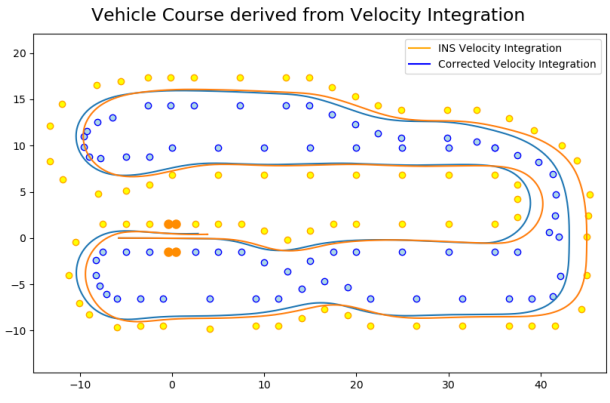

Figure 5: Velocity integrations of vehicle course for INS velocity measurements (orange) and corrected EKF measurements (blue)

6 Localization/SLAM

The objective of the localization module is to continuously provide an estimate of the vehicle's pose within a known track. On the other hand, the objective of the SLAM module is to provide an estimate, not only of the vehicle's pose, but also of the track layout i.e. the positions and the colors of the cones, in case the track is not known in advance.

Given that FS circuits typically have relatively flat terrain and the cones used to mark the track boundaries are uniform in shape, a preferred representation of the track is a 2D map. In this representation, the cones are represented as individual points on the map, allowing for a simplified and efficient representation of the track layout.

There are two major categories of SLAM algorithms that are used for solving this sort of problem: filter and graph SLAM. Graph SLAM approaches are generally superior in terms of accuracy as they combine all the pose estimates in the computation procedure, unlike filter SLAM algorithms that consider the last pose only and the motion model [1]. This comes at a cost in memory space and computation time but advancements such as iSAM2 [9], which is used in our autonomous system, have mitigated these drawbacks enabling the online execution of graph-based SLAM.

iSAM2 constructs a graph, where the vertices consist of vehicle and cone poses and the edges consist of odometry and observation measurements. The lengths of these edges are determined by integrating the velocity estimates and the process described in Section 4, respectively. The covariance matrices of the edges are derived from the EKF and from experimenting on test data, respectively.

One important aspect of both localization and SLAM modules is the data association i.e. the matching of the observed cones to already mapped cones. The cones are matched using the Nearest Neighbor algorithm and the Mahalanobis distance as a distance metric.

To address the issue of false positives on the map, only a selected set of cones that have been observed more than a specific number of times is sent to the path planning module.

The localization module will be used in the Acceleration and Skidpad events. The SLAM module will be used in the Autocross runs, until a satisfactory map of the circuit is obtained. As soon as the map is available, P23 shall enter localization mode again, reducing the computational load and preventing false positives from causing a DNF.

7 Motion Planning

The objective of the motion planning module is to determine a path for the autonomous vehicle to follow and compute the necessary inputs for executing that path. During the design process, a governing principle was followed to establish a robust framework that will remain suitable even when the team's primary focus shifts on performance. Safety factors are to subsequently be applied to ensure the accomplishment of this year's goal, which is to complete all DV disciplines.

7.1 Path Planning

The objective of the path planning module is to provide the centerline to the control module when required. The Acceleration, EBS Test, Skidpad, and Trackdrive tracks are known beforehand either from the rules or, in the case of Trackdrive, from the Autocross track. Therefore, only during the Autocross discipline is online path planning required.

The cone map and the pose of the vehicle is obtained directly from the SLAM module. To focus on relevant cones, a local map is constructed, considering only the cones around and ahead of the vehicle when creating the path.

To discretize the space, a triangular mesh of the cone set is constructed. A Delaunay triangulation is performed, due to its property of maximizing the minimum angle of the triangles [6], thus avoiding narrow triangles. The Computational Geometry Algorithms Library (CGAL) [15] is employed to perform the triangulation and facilitate all other geometric constructions.

The path planning algorithm begins by selecting the triangulation face that corresponds to the initial position of the vehicle. It then proceeds recursively to visit neighboring faces, excluding those that have already been visited, in a depth-first search manner. The search process continues until one of the following conditions is met: (a) the border of the triangulation is reached, (b) all neighboring faces have been visited, or (c) a predetermined maximum depth is reached.

Once each search reaches its conclusion, the formulated path is evaluated using a cost function. At each junction, the best path is retained based on the evaluation results. The developed cost function penalizes paths that cross edges of the same color, exhibit large angle changes, or have excessive distances between selected midpoints. Considering that errors in the calculated path tend to occur towards the end, where the reliability of the perception pipeline diminishes, the selected path is trimmed (shortened) until its cost falls below a specified threshold, ensuring it is safe to follow. By computing the best path along the midpoints of the triangulation edges, an approximate centerline of the track is obtained.

To achieve a smooth and densely sampled path, a cubic spline is interpolated over the midpoints that constitute the selected path before being provided to the control module. The system of equations arising from the second derivative constraints of the cubic spline is solved using the Thomas algorithm, while the Sherman-Morrison formula is employed for closed-loop curves [2]. Finally, the curve is parametrized using arc length to ensure a constant speed traversal of the curve points with respect to the curve parameter.

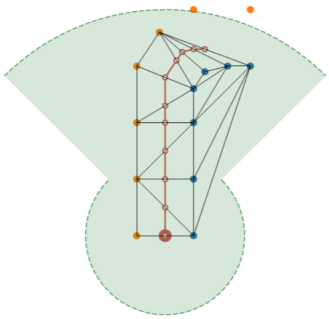

Figure 6: The area (green) of the cones considered during path planning and the path produced

7.2 Controls

The objective of the control algorithms is to optimize the vehicle's progress along a given path while considering vehicle and track layout constraints. As a result, two control strategies are employed: one for scenarios where the full map is known, and another for situations with limited lookahead. In the former case, predictive methods can be used to maximize vehicle performance, while in the latter, a more simplistic approach based on the local map is employed.

7.2.1 Model Predictive Control

In the known track layouts of the competition (Acceleration, Skidpad and Trackdrive), a Model Predictive Controller is employed. The controller is based on the commercial FORCES PRO NLP Solver (Version 6.0.1) [18], which solves a constrained nonlinear optimization problem using Sequential Quadratic Programming (SQP) methods. A single optimization problem is formulated every 25ms, considering $N = 40$ lookahead points.

Regarding the vehicle model, a linearly blended kinematic and dynamic bicycle model is employed, similar to Kabzan et al. [8]. The blending is based on the longitudinal velocity and occurs within the range of $u_{min} = 4\ m/s$ and $u_{max} = 7\ m/s$. Below the minimum and above the maximum velocity, the controller only considers the kinematic and dynamic model equations, respectively.

The optimizer is provided with a set of hard constraints regarding the vehicle state and input vectors, along with a weighted cost function. The primary objective of the cost function is to ensure accurate tracking of the spline-fitted midpoints while rewarding an increase in track progress percentage. Additionally, the cost function penalizes potential causes of instability, such as sharp variations in input commands and high values of tyre slip.

The final outputs of the MPC pipeline are the wheel angle of the front tyres and the longitudinal rear tyre force (F_{rx}), which are converted into desired commands for the low-level controller of the steering actuator and the VCU, respectively. It should be noted that converting F_{rx} to a torque command is a valid option for small tyre slip ratios.

7.2.2 PID - Pure Pursuit Control

In the case of an unknown environment, a decoupled approach is adopted for the longitudinal and lateral controls of the vehicle. For longitudinal control, a Proportional-Integral-Derivative (PID) controller based on a velocity profile is utilized, while for lateral control, a Pure Pursuit controller with dynamic look-ahead distance has been developed.

The velocity profile is created using a method similar to Subosits & Gerdes [13], known as the "Three-pass method." This technique plans the target velocity along the path by considering traction and engine limits. Given the primary objective of the autonomous system is reliability, certain constraints such as maximum friction force and motor torque are scaled down by a safety factor, ensuring that the vehicle operates within stability limits. Moreover, the target speed at the end of the available path is set to zero, allowing the vehicle to safely stop or reduce speed to navigate any turn that may not yet be visible to the autonomous system.

The PID controller has been modified to prevent Integral Windup by clipping the output of both the Integral term and the controller itself to some maximum value, disabling the integrator when the error is outside a controllable region. Additionally, the velocity profile guarantees gradual setpoint changes, preventing the integrator from accumulating error. The controller has been tuned using the Ziegler-Nichols method, with slight adjustments to prevent overshooting.

The Pure Pursuit controller, as described in [4], incorporates a dynamic look-ahead distance, as constantly high values can lead to corner cutting and constantly low values can cause oscillations when attempting to aggressively rejoin the path. To mitigate these issues, the controller adapts the look-ahead distance based on the current speed of the vehicle and the curvature of the path. Higher speeds and lower curvatures necessitate a larger look-ahead distance, while lower speeds and higher curvatures warrant a smaller look-ahead distance. The eventual value is bounded within predefined minimum and maximum limits to maintain stability.

8 Conclusion

Overall, the autonomous system pipeline of P23 demonstrated promising results in perception, velocity estimation, localization/SLAM, and motion planning. Looking ahead, early-season testing on the actual vehicle will provide valuable insights into the system's performance and allow for fine-tuning and optimization, while acquiring a LiDAR sensor would greatly enhance the perception capabilities of the vehicle by providing detailed depth information.

References

[1] Bashar Alsadik and Samer Karam. The simultaneous localization and mapping (slam)-an overview. *Surv. Geospat. Eng. J*, 2:34–45, 2021.

[2] Milan Batista and Abdel Rahman A Ibrahim Karawia. The use of the sherman–morrison–woodbury formula to solve cyclic block tri-diagonal and cyclic block penta-diagonal linear systems of equations. *Applied mathematics and computation*, 210(2):558–563, 2009.

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[4] R Craig Coulter. Implementation of the pure pursuit path tracking algorithm. Technical report, Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, 1992.

[5] Daniel Díaz Martínez. Object detection with radar: present and future automotive technology. B.S. thesis, Universitat Politècnica de Catalunya, 2022.

[6] Herbert Edelsbrunner. Triangulations and meshes in computational geometry. *Acta numerica*, 9:133–213, 2000.

[7] Sergi Foix, Guillem Alenya, and Carme Torras. Lock-in time-of-flight (tof) cameras: A survey. *IEEE Sensors Journal*, 11(9):1917–1926, 2011.

[8] Juraj Kabzan, Miguel I Valls, Victor JF Reijgwart, Hubertus FC Hendrikx, Claas Ehmke, Manish Prajapat, Andreas Bühler, Nikhil Gosala, Mehak Gupta, Ramya Sivanesan, et al. Amz driverless: The full autonomous racing system. *Journal of Field Robotics*, 37(7):1267–1294, 2020.

[9] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[10] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66):eabm6074, 2022.

[11] Francisca Rosique, Pedro J Navarro, Carlos Fernández, and Antonio Padilla. A systematic review of perception system and simulators for autonomous vehicles research. *Sensors*, 19(3):648, 2019.

[12] Kieran Strobel, Sibo Zhu, Raphael Chang, and Skanda Koppula. Accurate, low-latency visual perception for autonomous racing: Challenges, mechanisms, and practical solutions. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1969–1975. IEEE, 2020.

[13] John Subosits and J Christian Gerdes. Autonomous vehicle control for emergency maneuvers: The effect of topography. In *2015 American Control Conference (ACC)*, pages 1405–1410. IEEE, 2015.

[14] Juan Terven and Diana Cordova-Esparza. A comprehensive review of yolo: From yolov1 to yolov8 and beyond. *arXiv preprint arXiv:2304.00501*, 2023.

[15] The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition, 2023.

[16] Miguel I Valls, Hubertus FC Hendrikx, Victor JF Reijgwart, Fabio V Meier, Inkyu Sa, Renaud Dubé, Abel Gawel, Mathias Bürki, and Roland Siegwart. Design of an autonomous racecar: Perception, state estimation and system integration. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 2048–2055. IEEE, 2018.

[17] Niclas Vödisch, David Dodel, and Michael Schötz. Fsoco: The formula student objects in context dataset. *arXiv preprint arXiv:2012.07139*, 2020.

[18] Andrea Zanelli, Alexander Domahidi, Juan Jerez, and Manfred Morari. Forces nlp: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs. *International Journal of Control*, 93(1):13–29, 2020.