Portfolio

mbsbahru@umich.edu | +62-895-3411-02014 | mbsbahru

Profile Summary

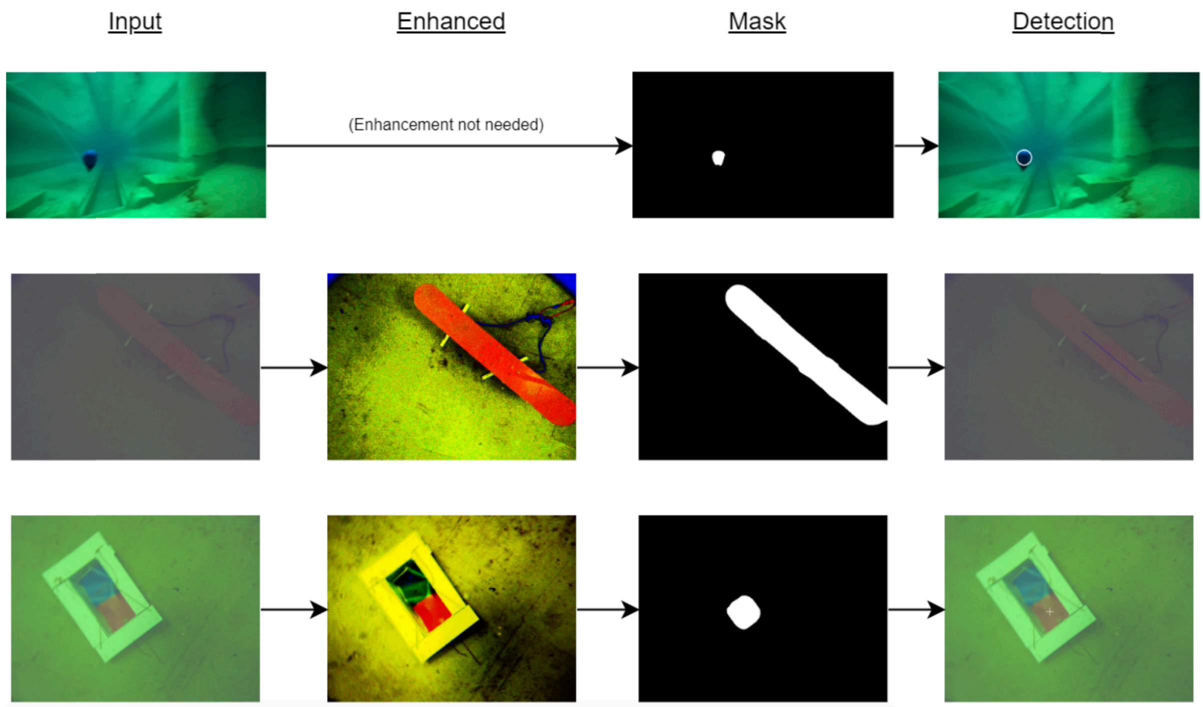
Marine robotics enthusiast with academic degrees in Marine Engineering. Actively participated in university robotics teams (Barunastra Roboboat ITS and Michigan Robosub) and control-related laboratories while also completing robotics-related courses.

Featured Projects

RoboNation's 27th International RoboSub Competition (2024)

As a software developer at the Michigan Robotic Submarine team
Team: Nolan Kuza, *Muhammad Bahru Sholahuddin*, et al.
My Responsibility: Detecting and localizing the competition's underwater objects efficiently.
Strategy:
- Enhanced image color using histogram equalization combined with manual white balance configuration.
- Segmented objects using a computationally efficient Hue-Saturation-Value (HSV) thresholding method.
- Refined segmentations using smoothing techniques (Median and Gaussian), morphological operators (Erode and Dilate), and Convex Hull.
- Classified objects by their features (vertices, areas, and edge ratios).
- Extracted and visualized key information (centroid position from the robot's 2D frame of reference, object area, and orientation).
- Created a Python desktop program to test the algorithms on acquired videos and image sequences from the robot's rosbag.
- Modified the team's existing ROS '.py' pipeline and '.yaml' parameter files to integrate the new algorithms.

Links: GitHub · Paper · Competition Rules and Tasks
Preview:

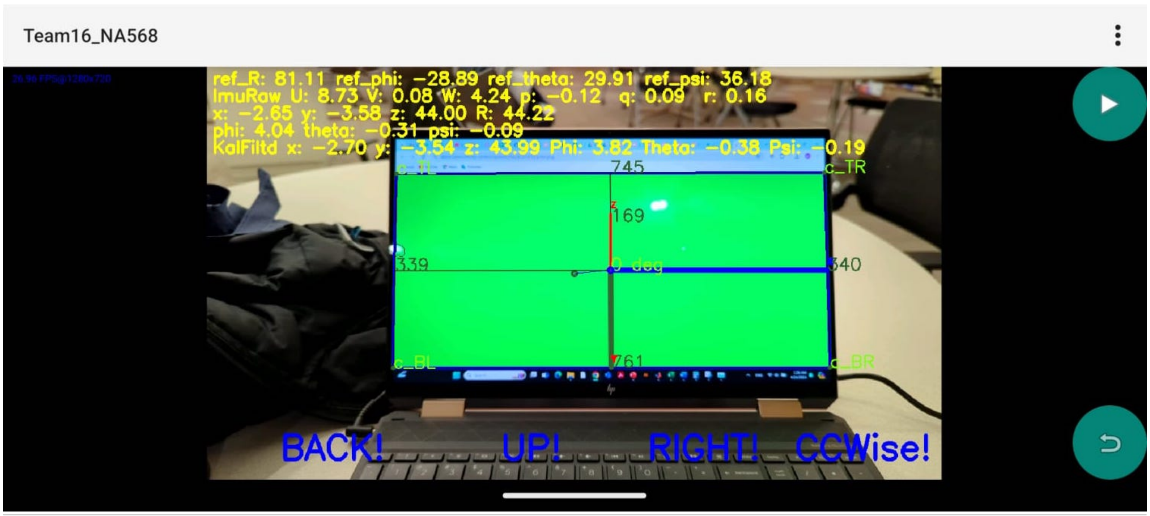


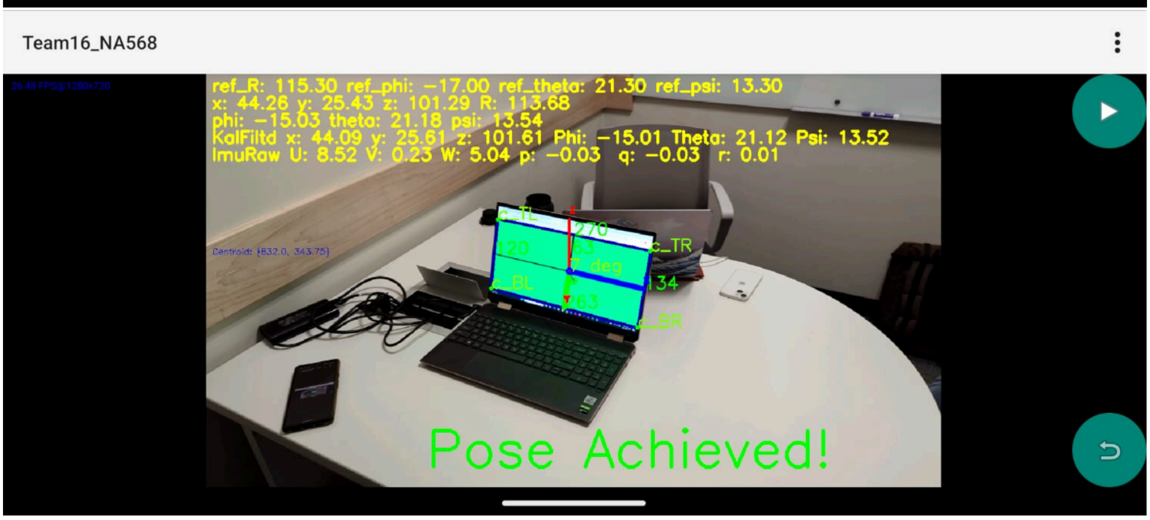
› Pose Perfect - An Android Pose Estimation & Guidance App (2024)

(ROB530-Mobile Robotics) self-directed class final project at the University of Michigan
Team: *Muhammad Bahru Sholahuddin*, Mohammed Buhlaigah, Zih-En Tseng, Usman Shahzad.
My Responsibility: Developed an Android application to perform 3D linear and angular (surge, sway, heave, roll, pitch, yaw) pose estimations for a single-vision camera relative to the object's frame of reference. Refined the estimations using Android accelerometer and gyroscope sensors with a Kalman filter and integrated the team's filtering and control algorithms into the Android program.
Strategies:

- Implemented Hue-Saturation-Value (HSV) object segmentation, noise reduction (smoothing and morphological operations), and contour refinement (Convex Hull).
- Extracted object features (centroid, edges, vertices) and displayed their positions within the camera's 2D frame.
- Performed camera calibration to obtain intrinsic parameters (optical center, focal length, skew coefficient) and radial-tangential distortion coefficients of the Android camera.
- Applied the Perspective-n-Point (PnP) algorithm to obtain the object's translational vector and rotational matrix relative to the camera frame of reference.
- Performed inverse frame transformation to use the object as the frame of reference and extracted the camera's linear (x, y, z) and Euler angular positions (ϕ, θ, ψ).
- Processed Android accelerometer and gyroscope variance data as measurement and process noises for Kalman filtering.
- Integrated the team's frequency-based filtering and bang-bang control algorithms into the Android project.
- Developed an algorithm to save and reset the desired 6 DoF position and orientation and displayed motion guidance based on any in-frame position and orientation.

Links: GitHub · Poster · Paper · Application
Preview:



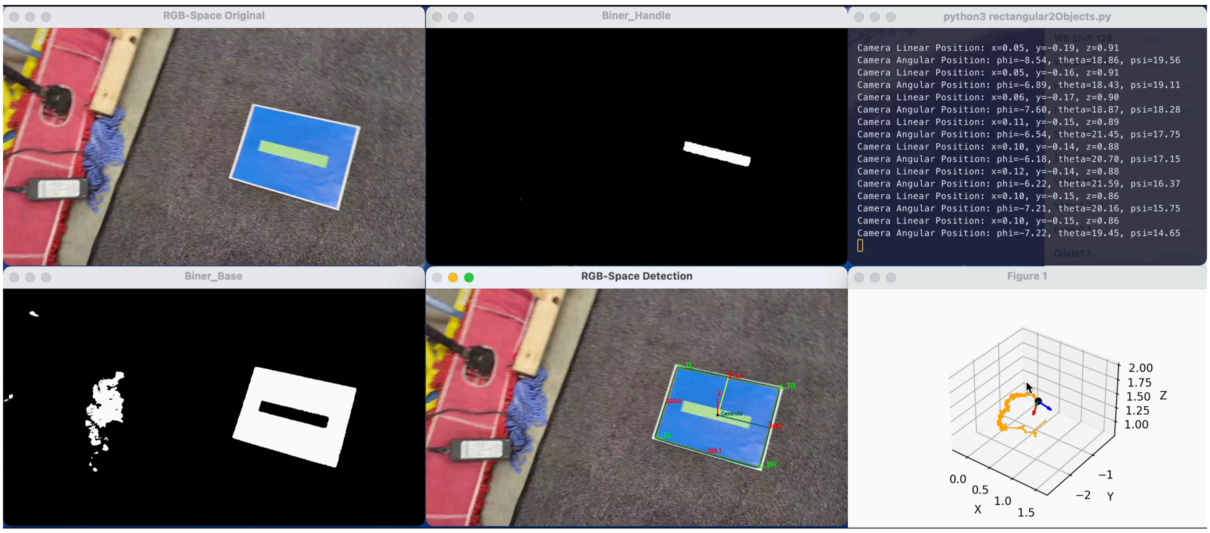


› Visual Path Planning for Underwater Manipulation (2024)

(ROB572 - Marine Robotics) self-directed class final project at the University of Michigan
Team: *Muhammad Bahru Sholahuddin*, Kathryn Wakevainen.
My Responsibility: Developed a Python program to estimate the 3D linear and angular (surge, sway, heave, roll, pitch, yaw) poses of underwater objects relative to a camera frame of reference and vice versa. The application provided either recorded or real-time visualization of the 6-Degree-of-Freedom (6-DoF) motion trajectory, displaying the camera's position and orientation. Evaluated the algorithm's performance to ensure the accuracy and reliability of pose estimations.
Strategy:

- Segmented two underwater objects using Hue-Saturation-Value (HSV) thresholding to differentiate between various color profiles. The segmentation process was further refined with smoothing techniques, contour treatment, and color enhancements to improve detection accuracy in low-visibility underwater environments.
- Developed an advanced algorithm to identify and track the corner vertices of the objects in real-time. This algorithm extracted multiple features, such as vertex positions and edge lengths, to classify and distinguish between different objects based on their geometric properties.
- Performed a camera calibration using Matlab, which involved estimating the camera's intrinsic parameters, such as optical center, focal length, skew coefficient, and radial-tangential distortion coefficients. This calibration was essential to correct lens distortions and enhance the accuracy of the pose estimation.
- Applied the Perspective-n-Point (PnP) algorithm, Rodrigues' rotation formula, and inverse transformation methods to compute the camera's linear (x, y, z) and Euler angular positions (ϕ, θ, ψ) relative to the object's frame of reference. This process enabled precise tracking of both the camera and the objects in 3D space.
- Implemented an online low-pass filter to smooth the trajectory data and reduce the impact of noise and sudden fluctuations. This filtering technique ensured a more stable and reliable representation of the camera and object movements over time.
- Calculated and displayed the reprojection error for each frame to assess the accuracy of the pose estimations. The reprojection error analysis provided valuable insights into the system's performance and helped identify areas for further improvement.
- Demonstrated the complete 6-Degree-of-Freedom (6-DoF) motion trajectory of the camera relative to the object's frame, using Matplotlib. The visualization included detailed plots showing the path and orientation of the camera and objects, providing a clear representation of the system's capabilities.

Links: Paper & Code Attachment · Video
Preview:



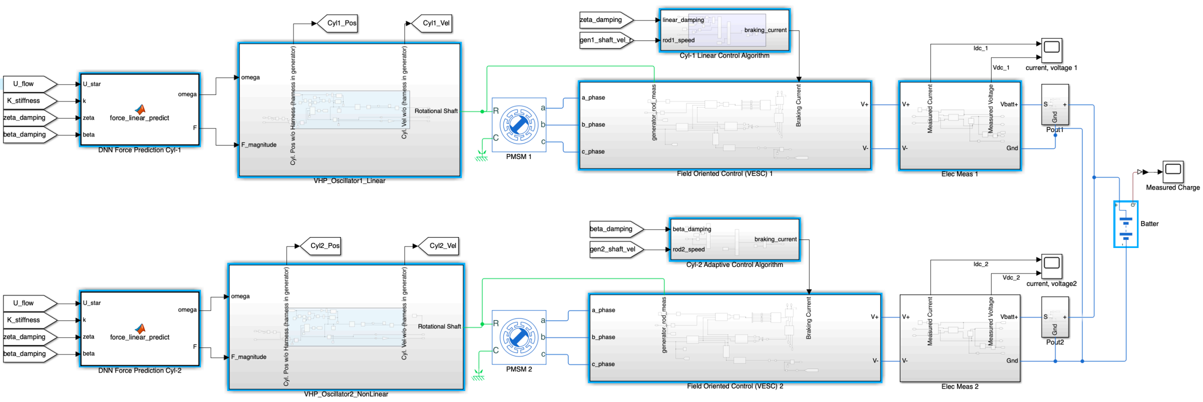
› Control Co-Design of VIVACE Renewable Energy Converter (2023-2024)

Master's research project at the Marine Renewable Energy Lab (MRELab)
Team: *Muhammad Bahru Sholahuddin*, Prof. Michael M. Bernitsas (Advisor).
My Responsibility: Selected the novel control strategy and designed a comprehensive system model of the VIVACE (Vortex Induced Vibration Aquatic Clean Energy) marine hydrokinetic converter in Matlab Simulink. Developed the control algorithm, tested the generators, and optimized the damping parameters to enhance the overall energy harnessing efficiency.

Strategy:

- Designed a complete system model of the VIVACE converter in Matlab Simulink, incorporating components such as the hydrodynamic model, spring-mass-damper system, control algorithm, power generation unit, and battery load system, ensuring all elements interacted cohesively for optimal performance.
- Modeled complex hydrodynamic interactions using a deep neural network, trained on a large dataset of experimental measurements, to accurately represent fluid-structure interactions and predict the converter's dynamic response under various flow conditions.
- Applied the Field-Oriented Control (FOC) methodology with a Permanent Magnet Synchronous Motor (PMSM) to dynamically adjust the damping force and efficiently convert mechanical energy into electrical energy. This approach allowed precise control of the system's response to changing water flow rates and environmental conditions.
- Conducted system identification procedures to accurately estimate key parameters of the selected generators, including flux linkage (λ), internal resistance (R), and inductances (L_q and L_d), ensuring the control system was precisely tuned for real-world conditions.
- Tested two real-world generator setups to evaluate their performance characteristics, focusing on efficiency versus rotational speed and power output. This analysis was for selecting the most suitable generator configuration for different operational scenarios.
- Fine-tuned the control gains of the FOC system to optimize the dynamic response of the energy converter, achieving stable operation and high efficiency across a range of environmental conditions and operational settings.
- Developed a control algorithm capable of real-time adjustment based on instantaneous damping coefficients and generator speed, ensuring the system could maintain optimal performance and adapt to varying flow conditions.
- Analyzed and evaluated the power output and overall efficiency of the energy converter system under various test conditions, including different water flow rates (m/s), spring stiffnesses (N/m), linear damping coefficients (ζ, Ns/m), and non-linear damping coefficients (β, Ns^2/m^2). Created a surrogate model to represent the system's performance across these variables.
- Performed and evaluated optimization of both linear and non-linear damping parameters using advanced meta-heuristic algorithms, such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA), to identify the optimal settings for maximum energy extraction under diverse environmental conditions.

Links: Control Algorithm · NDA Agreement (Files containing VIVACE data are still undisclosed)
Preview:

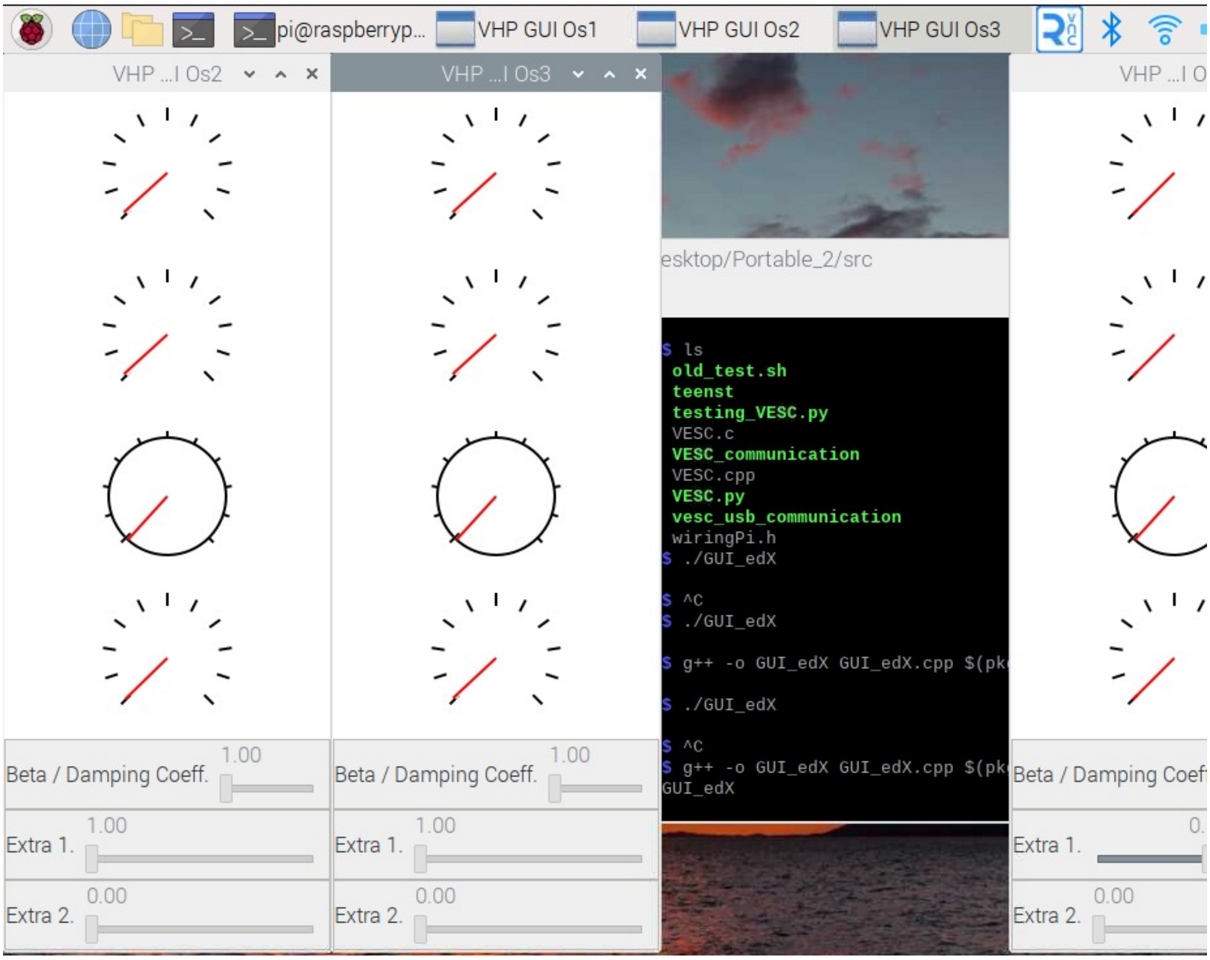


› Control Interface of the VIVACE Renewable Energy Converter (2023)

Summer Research Project at the Marine Renewable Energy Lab (MRELab)
Team: *Muhammad Bahru Sholahuddin*, Pratik Shiveshwar, and Prof. Michael M. Bernitsas (Advisor).
My Responsibility: Converted the existing Raspberry Pi GUI Python program, used for interfacing with VIVACE parameters, into a C++ application. Implemented the Lightweight Communications and Marshalling (LCM) library for inter-process communication and real-time data visualization. Ensured compatibility with the lab's hardware setup and reduced control and monitoring latency of the system.
Strategy:

- Emulated the Raspberry Pi OS using QEMU (Quick Emulator) CLI to test the program without utilizing lab hardware, allowing for more efficient development and testing cycles.
- Migrated the existing Python-based GUI developed using 'tkinter' to the 'gtkmm' library for building the GUI in C++, providing improved performance.
- Integrated the LCM library to facilitate communication between the GUI and various sensor systems, such as the encoder and programmable load controllers, ensuring a more real-time monitoring.
- Developed widgets for real-time visualization, including dynamic gauges and sliders, to display critical parameters such as generator voltage, current, encoder position, and velocity.
- Implemented a multi-threaded architecture to handle concurrent data processing and GUI updates, improving responsiveness and reliability of the interface under high-frequency data transmission.
- Conducted testing of the application on the emulated environment to identify and resolve potential issues before deployment on the actual hardware.

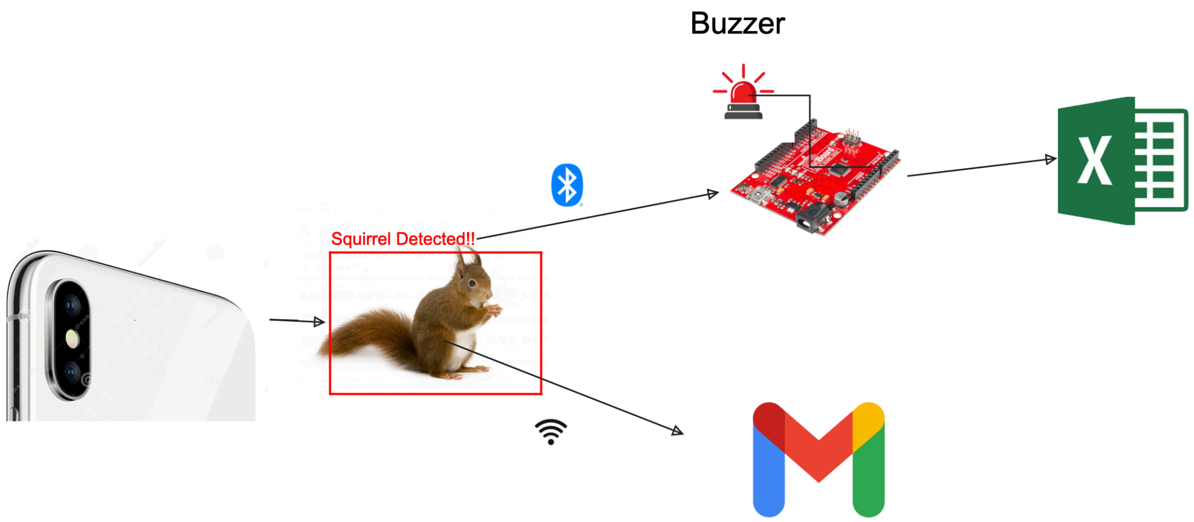
Links: Project Code
Preview:

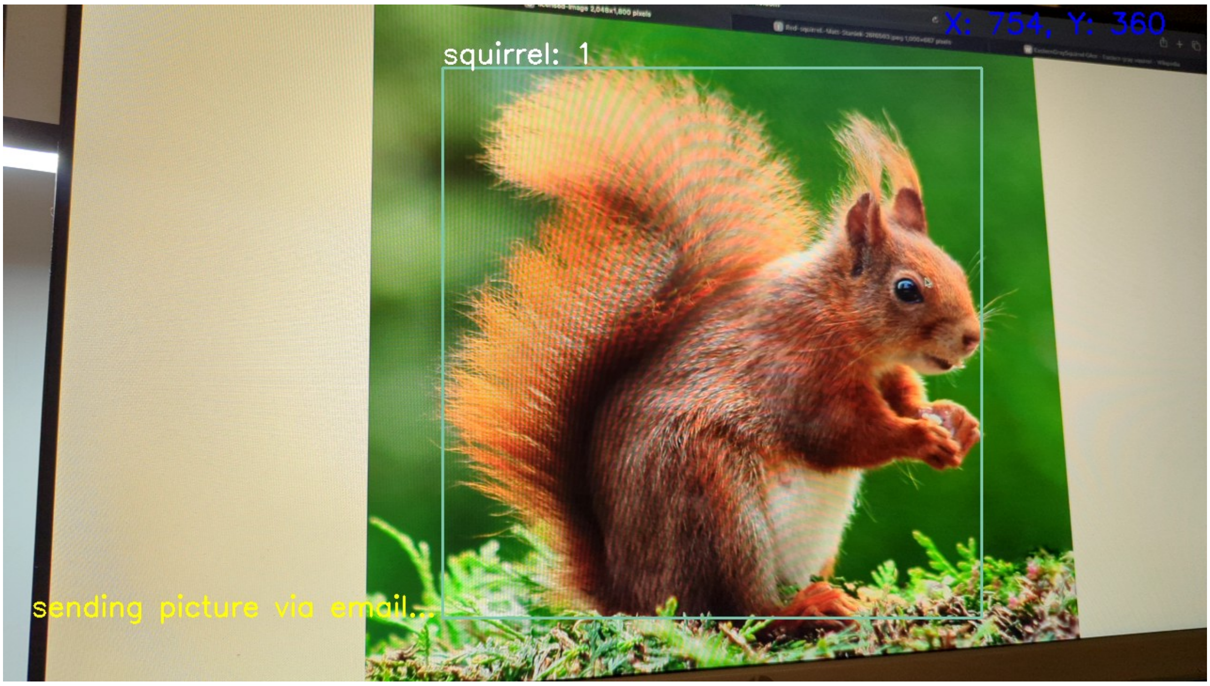


› Android Squirrel Image Detection and Alert using YOLO (2023)

(CEE575 - Sensors, Data, Smart Systems) self-directed class final project at the University of Michigan
Team: *Muhammad Bahru Sholahuddin* and Stefano Sion Pizzi.
My Responsibility: Developed an Android application to detect the presence of squirrels, alert the user via a buzzer, update their position in a 2D frame using Google Sheets, and send their pictures.
Strategy:
- Collected 1,810 squirrel images and annotations from Google APIs to create the dataset.
- Implemented the You Only Look Once (YOLO) version 3 algorithm for training and testing the dataset using Google Colab, and analyzed the optimal kernel size.
- Created an Android project using the OpenCV library to read the trained data. When a squirrel is detected, the app sends its position within the 2D frame via Bluetooth and emails the picture to the user.
- Programmed the SparkFun Photon RedBoard via Particle Web IDE to receive data from the Android app using the HC-05 module through the Universal Asynchronous Receiver Transmitter (UART) protocol. When a squirrel is detected, the system rings the buzzer and updates the squirrel's position in Google Sheets using the If This Then That (IFTTT) protocol.

Links: GitHub · Application · Video
Preview:



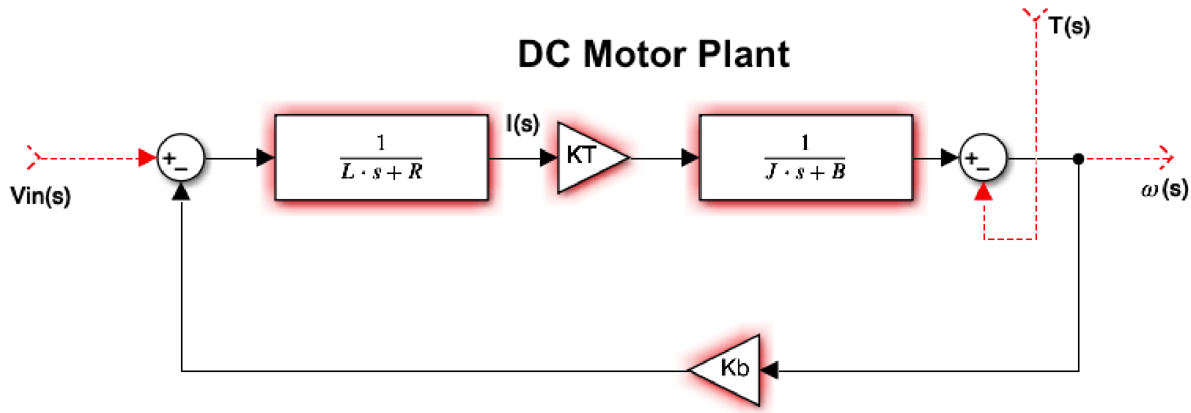


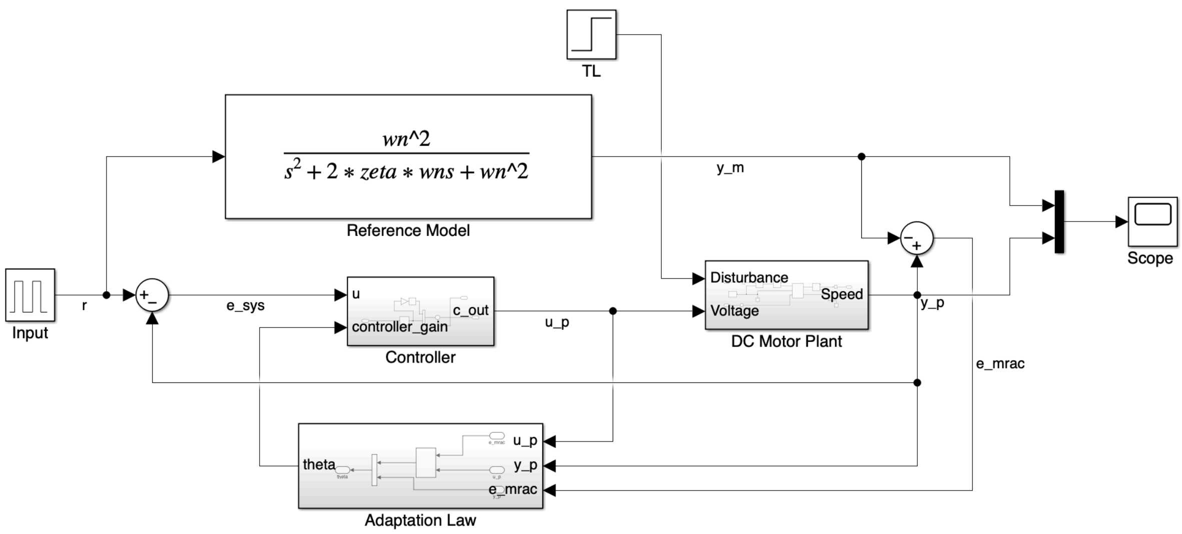
› DC Motor Speed Control using Model Reference Adaptive Control (2022)

(NA583 - Adaptive Control) self-directed class final project at the University of Michigan
Team: Max Wu, Nick Boston, *Muhammad Bahru Sholahuddin.*
My Responsibility: Derived the mathematical model of the DC motor and designed the Model Reference Adaptive Control (MRAC) system in Simulink.
Strategy:

- Derived the mathematical model of the DC motor by applying Kirchhoff's Voltage Law to the electrical circuit and Newton's second law to the mechanical system. This resulted in a transfer function that represents the relationship between input voltage and angular velocity. The model was characterized as a typical second-order system with parameters including torque constant (K_t), armature inductance (L), armature resistance (R), shaft moment of inertia (J), viscous friction coeff (β), and back EMF constant (K_b).
- Designed the Model Reference Adaptive Control (MRAC) system using the direct Lyapunov method, ensuring the stability of the system through the Strictly Positive Real (SPR) condition. The reference model was implemented as a second-order system, which was chosen to match the desired performance characteristics of the DC motor.
- Implemented the adaptive control law in Simulink, which continuously adjusts the control gains to minimize the error between the actual motor speed and the desired reference model response. The adaptation mechanism was designed to cope with uncertainties in the motor parameters and external perturbations.
- Developed a custom script to automate the initialization of the controller and plant parameters in Simulink. This script allowed for quick reconfiguration of the system, such as different motor configurations.
- Simulated different operating conditions in Simulink to test the MRAC system's robustness, and analyzed the results performances.

Links: Project Report
Preview:

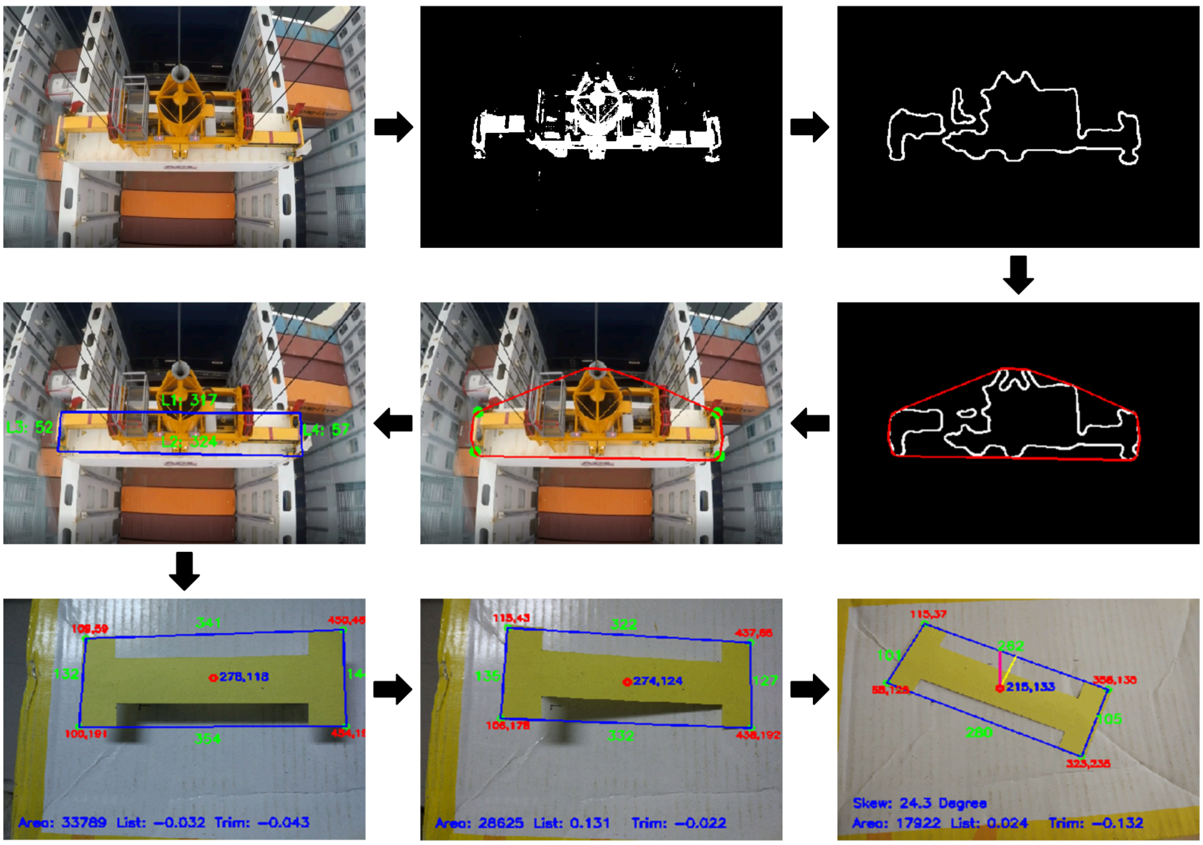




› Pose Detection Prototype of CC Spreader using Computer Vision (2017)

Final Project at the Institut Teknologi Sepuluh Nopember
Team: *Muhammad Bahru Sholahuddin*, Dr. A. A. Masroeri (Advisor), Juniarko Prananda, M.T. (Adv.).
My Responsibility: Developed a computer vision-based system to detect the Container Crane (CC) spreader and estimate its 3D orientation. Designed and implemented algorithms for image segmentation, noise reduction, and feature extraction to identify the spreader's position and motion parameters.
Strategy:
- Conducted experiments to determine how changes in lighting conditions affect the accuracy of HSV-based color segmentation. This analysis was intended to develop a segmentation algorithm that could handle varying illumination in an outdoor environment.
- Converted RGB images to HSV color space and applied adaptive thresholding to segment the spreader from the background. Refined the segmentation results using a combination of Median and Gaussian filters to reduce noise, followed by morphological operations such as erosion and dilation to eliminate small artifacts. Applied the Convex Hull algorithm to accurately delineate the spreader's contours.
- Implemented an algorithm to identify the spreader's corner points and calculate its geometric properties, including centroid, area, and orientation angles. Used the detected features to estimate the spreader's pose (position and orientation) relative to the camera's frame of reference. Calculated motion parameters such as sway, skew, trim, and list rates based on changes in these features over time.
- Integrated the computer vision algorithm into a simulated environment. Evaluated the system's performance in detecting the prototype-spreader's motion, and compared the results with ground-truth measurements to validate the accuracy of the pose estimation.
- Conducted testing under different lighting and environmental conditions to assess the robustness of the algorithm. Fine-tuned the exposure parameters of the segmentation and feature extraction processes to improve detection accuracy. Proposed enhancements for real-time application, such as using more sophisticated tracking algorithms and adapting to varying weather conditions.

Links: Project Report & Code Attachment · Journal · Article Paper
Preview:



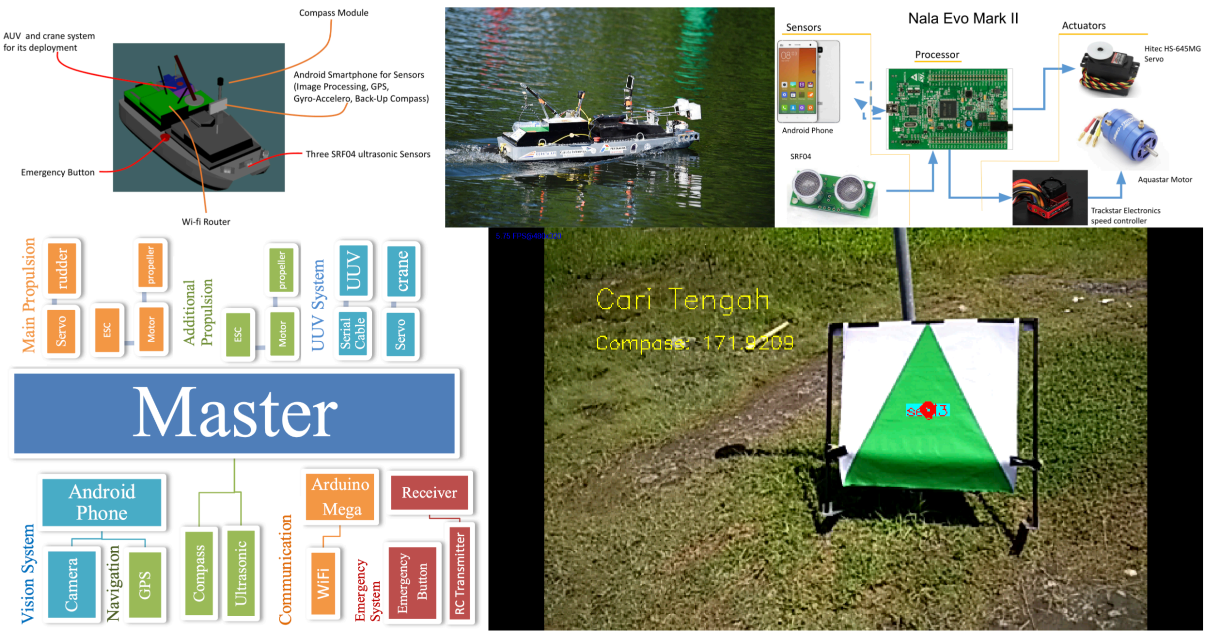
› Object Detection and Communication for Roboboat 2016 Challenge (2016)

(9th International Roboboat Competition) self-directed project at Barunastra Roboboat ITS
Team: Rudy D., Darwin S., Yohan P., *Muhammad Bahru Sholahuddin*, Ericza D. S., Anas M. N.
My Responsibility: Developed an Android-based object detection system using OpenCV, integrated with external sensors, and designed a communication pipeline between the Android application and a microcontroller via Bluetooth.

Strategy:

- Developed an object detection pipeline that performed segmentation, filtering, and object detection. The pipeline utilized color thresholding, noise reduction with smoothing techniques (Median and Gaussian filters), and morphological operations (Erosion and Dilation) to refine object boundaries.
- Implemented algorithms to classify objects based on Hue-Saturation-Value (HSV) color space and geometric features, such as form factor and number of edges, allowing the system to distinguish between various objects in the competition environment.
- Tested the detection algorithm on recorded videos from the water field using a custom C++ program built with CMake to evaluate the system's performance.
- Created individual Java activities for each object type, extracting feature data (e.g., Hue, Saturation, Value, shape parameters, and morphological properties like erosion and dilation) for real-time analysis within the app.
- Utilized the Android camera's auto-focus, white balance, and exposure settings to manually adjust the view for varying outdoor lighting conditions, improving the robustness of the detection system.
- Retrieved and processed raw sensor data from Android's Global Positioning System (GPS), light, accelerometer, gyroscope, and magnetometer to provide orientation data. Designed the system to reset the orientation values easily, allowing better synchronization with the detection process.
- Established communication with an STM32F4 microcontroller via Universal Asynchronous Receiver Transmitter (UART) over Bluetooth. Implemented a custom-designed parsing system to transmit real-time 2D object position data, along with sensor data (GPS, light, orientation), to the microcontroller for decision-making. Also retrieved command information from the microcontroller to update the Android system with the current mission state.

Links: GitHub · Paper · Competition Rules and Tasks · Application
Preview:



› Android Touch HSV Color Picker (2014)

Self-project during my role as a junior programmer at Barunastra Roboboat ITS
My Responsibility: Developed an interactive Android application to segment objects based on their Hue-Saturation-Value (HSV) color and enable easy selection and adjustment of HSV parameters through a touch interface.

Strategy:
- Implemented OpenCV for Android to process live camera frames and convert them to HSV color space.
- Allowed users to adjust HSV range values dynamically using a touch interface, which triggered updates to filter and display objects based on selected HSV values.
- Detected and highlighted objects in real-time by applying color-based segmentation, while displaying the detected object's features such as Hue, Saturation, and Value.
- Developed interactive SeekBars for adjusting the HSV thresholds and provided live feedback as users refined the settings.
- Designed custom UI components to enable users to touch areas on the live camera feed and retrieve the HSV value of the selected point.
- Used OpenCV's contour detection to extract object boundaries and provide a visual overlay on detected objects in the camera feed.
- Enabled color calibration through manual selection and tuning of HSV parameters, making the application useful for vision-based robotic applications such as obstacle detection.

Links: GitHub · Application
Preview:

Relevant Courses

- **Mobile Robotics – ROB530/EECS568/NAVARCH568** (view syllabus)
  Topics: Bayes Filters, Kalman Filtering, Nonlinear Filtering, Rigid Body Motion, Matrix Lie Groups, Robot Motion and Invariant EKF, Localization, Mapping, Graph SLAM and Registration.

- **Marine Robotics – ROB572/NAVARCH569** (view syllabus)
  Topics: Underwater Vehicle Design, Mathematical Modeling (Kinematics, Hydrostatics, Hydrodynamics), Guidance and Control, Sensing and Navigation, Operations.

- **Self Driving Cars: Perception and Control – ROB535/NAVARCH565** (view syllabus)
  Topics: Linear Algebra, Ordinary Differential Equations, State Space, Linearization, Stability, State Feedback, Linear Quadratic Regulator (LQR), Vehicle Kinematics and Dynamics, Model Predictive Control, Nonlinear Trajectory Optimization, Sampling-based Motion Planning, Perception (Camera Models, Geometry, Calibration, Object Detection, Localization, RANSAC), Deep Learning.

- **Navigation and Guidance: From Perception to Control – AEROSP584** (view syllabus)
  Topics: Vectors, Matrices, 3D Geometry, Kinematics, Sensor Specifications, Position Fixing, Inertial Navigation, Optimal Estimation, Kalman Filtering, Non-Inertial Navigation, Non-Inertial Navigation in GPS-Denied Environments, Guidance for Pursuit-Evasion.

- **Infrastruct Sensing / Sensors, Data, and Smart Systems – CEE575** (view syllabus)
  Topics: Sensors (Resistive, Capacitive, Inductive, Piezoelectric, etc.), Analog-to-Digital Converters (ADC), Op-Amps, Delta-Sigma Modulator, Multiplexing, Fourier Analysis and Transform, Low-High-Band Pass Filters, System Design and Integration, Internet of Things (IoT).

- **Adaptive Control – NAVARCH583** (view syllabus)
  Topics: Stability and Passivity Theory, Parametric Models for Dynamic Systems, Online Parameter Identification, Adaptive Observers, Model Reference Adaptive Control, Indirect Adaptive Control, Robust Adaptive Laws, Nonlinear System, Multivariable System, Hybrid Adaptive System, Adaptive and Learning Control, Applications of Adaptive Control.

- **Marine Control Systems – NAVARCH483** (view syllabus)
  Topics: Control-oriented System Modeling, Dynamic Behavior and Stability, Linear Systems and State Feedback, Frequency Domain Analysis and Design, Proportional-Integral-Derivative (PID) Control, Control Implementation and Computer-controlled Systems, Marine Control Applications, Design Optimization and Optimal Control.

- **Intro to Physical Oceanography – CLIMATE421** (view syllabus)
  Topics: The Physical Properties of Seawater, Air/Sea Interaction and Atmospheric Circulation, Development of Fluid Dynamical Equations Including Shallow Water Equations, Near-Inertial Motions, Shallow-Water Gravity Waves, Tides, Geostrophy and Thermal Wind, Ekman Flows and Wind-driven Oceanic General Circulation, Meridional Overturning Circulation and Energy Sources for Mixing, Stratification Dynamics and Internal Waves, Equatorial Waves and El Nino.

Skills

- Programming: C, C++, Java, Python, MATLAB, ROS.
- Document preparation: MS/Libre Office softwares, LaTeX.
- Analysis: Inferential Statistics, Root Cause Analysis, Bayesian Networks, Machine Learning.
- Design and Simulation: AutoCAD, Eagle, Simulink, ETAP, Maxsurf.