

# RoboCup 2026 - Smart Manufacturing League

## Robo-Erectus SML

### Technical Description Paper

Yusuf Pranggonoh (Team Leader), Carlos Acosta (Faculty Member), Chan Zi Heng, Ko Shao Hong, Subramani Arjun, Wang Junhui, and Yosuke Nakashima  
(Student Members)

Singapore Polytechnic,  
School of Electrical and Electronic Engineering (SEEE),  
500 Dover Road, Singapore, 139651  
yusuf\_pranggonoh@sp.edu.sg  
<https://www.team-robo-erectus.com>

**Abstract.** This paper presents a technical overview of Robo-Erectus SML, developed for participation in the RoboCup Industrial Smart Manufacturing League at RoboCup 2026. The system comprises multiple robotic platforms, including a warehouse robot with a four-wheel mecanum (omni-directional) mobile base integrated with a GLUON-2L6-4L3 robotic arm and customised end-effector, a Booster T1 humanoid robot, and a workbench robot based on the AgileX PiPER arm. The software architecture is built on the Robot Operating System (ROS) framework, enabling modular integration and scalable development. The robots are equipped with advanced object recognition, precise manipulation, and motion planning capabilities to support reliable operation in dynamic smart manufacturing environments.

## 1 Introduction

The Robo-Erectus project was established in 2002 at Singapore Polytechnic. It was one of the pioneering humanoid teams to compete in the RoboCup Humanoid League, focusing on autonomous soccer-playing robots. In 2013, the team expanded with the formation of Robo-Erectus@Work, which went on to achieve strong performances in subsequent RoboCup competitions. Building on this legacy, a new team, Robo-Erectus SML, has been established to compete in RoboCup 2026 under the Smart Manufacturing League (SML) of the RoboCup Industrial.

The team's primary research focuses on **mobile manipulation** in controlled and semi-outdoor environments, including object grasping from floors, shelves, and workstations; **omni-directional localization and navigation** in unconstrained spaces, and **object recognition and handling** within manufacturing settings.

The team is composed of students from the Diploma in Electrical and Electronic Engineering (DEEE) and Diploma in Computer Engineering (DCPE).

Year	Venue	Result
RoboCup German Open 2025	Nuremberg, Germany	<b>2nd Place</b>
RoboCup 2024	Eindhoven, Netherlands	<b>4th Place</b>
RoboCup 2023	Bordeaux, France	<b>5th Place</b>
RoboCup 2022	Bangkok, Thailand	<b>3rd Place</b>
RoboCup 2021	Virtual (Worldwide)	<b>8th Place</b>
RoboCup 2019	Sydney, Australia	<b>3rd Place</b>
RoboCup 2018	Montreal, Canada	<b>3rd Place</b>
RoboCup 2016	Leipzig, Germany	<b>3rd Place</b>
RoboCup 2015	Hefei, China	<b>2nd Place</b>
RoboCup 2013	Eindhoven, Netherlands	<b>4th Place</b>

**Table 1.** Team Robo-Erectus Result in RoboCup@Work League

Through their academic training, members have developed the competencies required to address the project’s technical challenges. Core expertise includes the Robot Operating System (ROS) [1], programming in Python and C++, electronic circuit design, system prototyping, and Artificial Intelligence (AI) applications, all which contributed significantly to the continued development and achievements of the Robo-Erectus SML project.

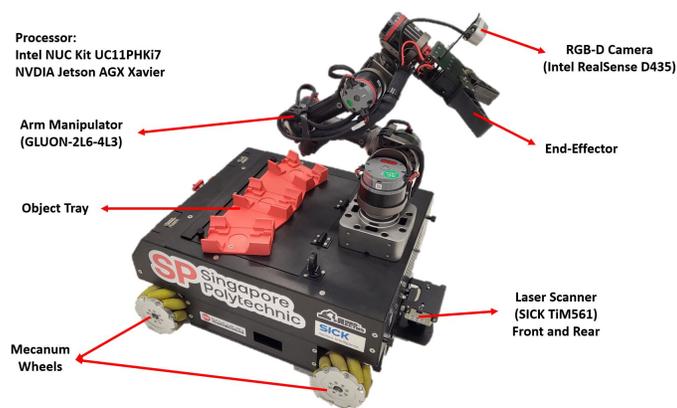
## 2 Robot Platform

### 2.1 The Warehouse Robot

The Robo-Erectus SML comprises two warehouse robotic platforms, an autonomous mobile robot and a humanoid robot based on the Booster T1 platform. The mobile platform is built on a four omni-directional drive system and integrated with a six-degree-of-freedom GLUON-2L6-4L3 [2] robotic arm, equipped with dual-parallel ROBOTIS Dynamixel XM430 as its end-effector. This configuration enables flexible and precise object handling in smart manufacturing tasks.

For localization and navigation within the competition arena, two SICK TiM561 outdoor laser scanners are mounted at the front and rear of the robot. An Intel RealSense D435 RGB-D camera is mounted on the end-effector to enable object detection and recognition during manipulation tasks. The system is powered by an Intel NUC11PHKi7 processor, which handles computer vision processing, artificial intelligence decision-making, and overall motion control. In addition, an NVIDIA jetson AGX Xavier Development Kit is incorporated to support deep learning application . Figure 1 illustrates the fully assembled Robo-Erectus SML mobile base, SPlackie.

The Booster T1 [3] humanoid robot features 23 degree-of-freedom, enabling full-body motion, dynamic balancing, and posture control. This capability allows the humanoid platform to perform complex movement such as stable walking and adaptive task execution in structured warehouse environments. Figure 2 show the Booster T1 humanoid robot, Sam.



**Fig. 1.** Team Robo-Erectus SML Mobile Robot, SPLackie



**Fig. 2.** Team Robo-Erectus SML Humanoid Robot, Sam

## 2.2 The Workbench Robots

The Robo-Erectus SML has one workbench robotics platform using AgileX PiPER Arm [4], which is a six-degree-of-freedom, equipped with dual-parallel ROBOTIS Dynamixel XM430 as its end-effector. An intel RealSense D435 RGB-D camera is mounted on the end-effector to enable object detection and recognition during manipulation tasks. The system is powered by GEEKOM Mini IT13 processor, which handles computer vision processing, artificial intelligence decision-making, and overall motion control. Figure 3 illustrates the workbench manipulator, PiPER.

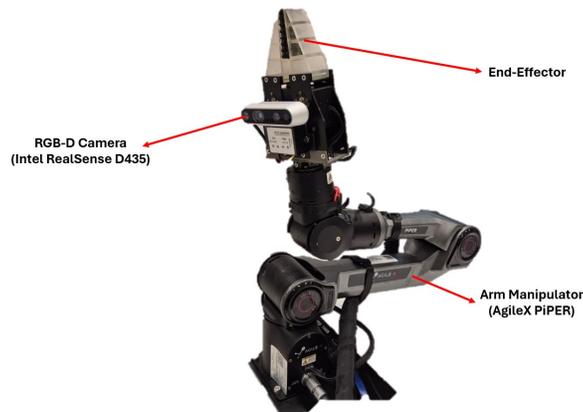


Fig. 3. Team Robo-Erectus SML Workbench Robot, PiPER

### 3 Software

SPlackie’s software architecture is structured into four principal frameworks: Navigation, Manipulation, Perception (Vision), and Task Sorter, as illustrated in Figure 4. This modular design promotes scalability, maintainability, and clear functional separation between high-level decision-making and low-level robotic control.

Competition tasks are initiated by the **atwork\_commander**, which transmits structured task commands to the **Task Sorter** framework. The Task Sorter interprets these commands, decomposes them into executable sub-tasks, and organizes them into a logical execution sequence implemented through a state-machine architecture. This ensures systematic task progression while handling conditional branching and exception recovery.

At the core of the system, the **State Machine** functions as the central coordinator. It dispatches goal commands to the Navigation, Perception, and Manipulation frameworks, while continuously monitoring their feedback and execution status. Based on real-time responses, it performs state transitions, error handling, and task re-planning when necessary.

Each framework is activated upon receiving specific goals from the State Machine, enabling synchronized and goal-driven execution across the entire robotic system.

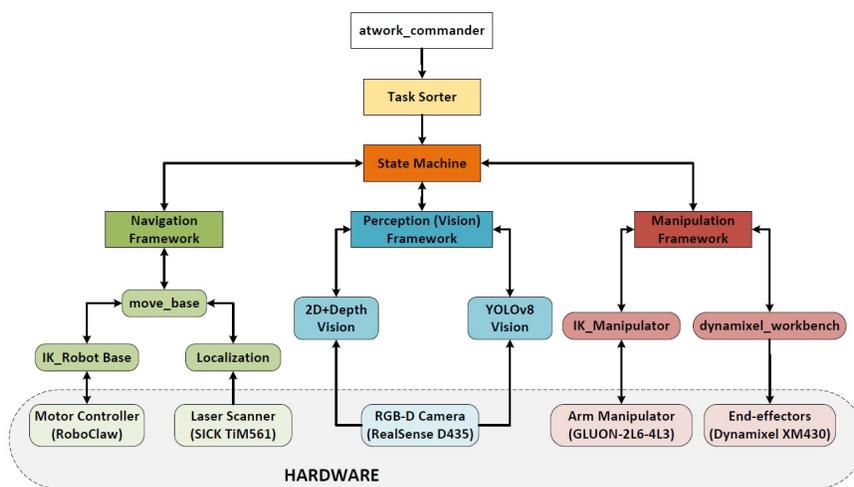


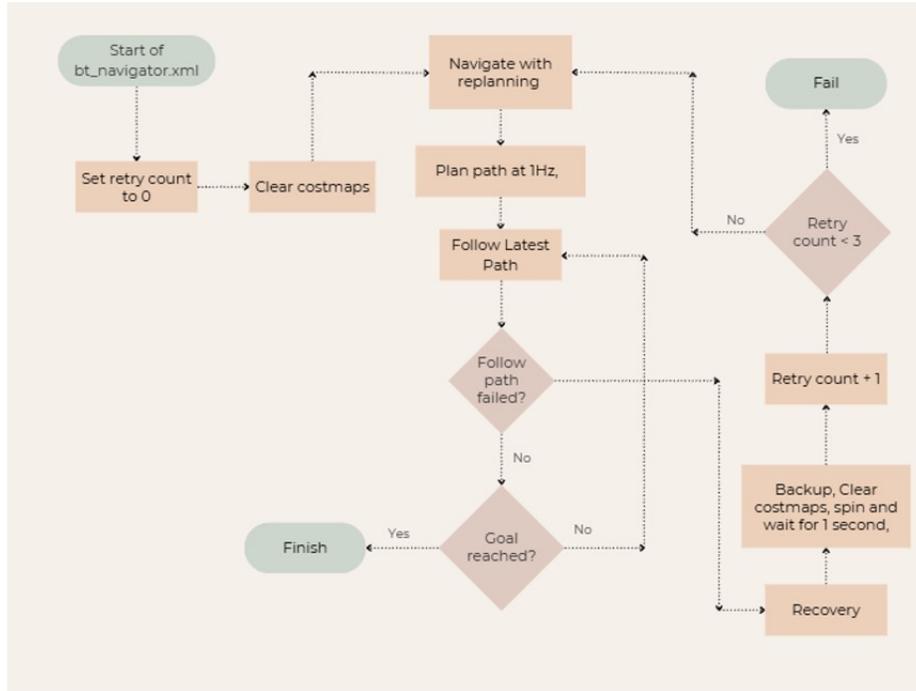
Fig. 4. SPLackie Robot Software Architecture

### 3.1 Navigation Framework (Behaviour Tree)

The Navigation framework governs how the robot reaches a designated goal safely, efficiently, and without collision, while optimizing travel time. SPLackie adopts a custom Behaviour Tree (BT) [5] architecture to manage navigation logic, enabling flexible execution flow, structured recovery handling, and improved robustness during dynamic operation.

As illustrated in Figure 5, once a navigation goal is received from the State Machine, the system initializes the retry counter to zero and clears both the local and global costmaps. This ensures that any outdated obstacle information is removed before computing a fresh global path. The planner then generates an updated trajectory, which the robot follows until the goal is successfully reached.

If the robot fails to reach the target, the Behavior Tree transitions into a recovery state. In this state, the system clears the costmaps again, performs a rotational scan (spin behaviour) to update environmental perception, and recomputes a new path. This recovery cycle is repeated up to three retry attempts. If the robot still fails after the third attempt, the navigation task returns a failure status and no further reattempts are executed, allowing the State Machine to handle subsequent decision-making.



**Fig. 5.** Custom Behaviour Tree Architecture

### 3.2 Manipulation Framework (Behaviour Tree)

Object manipulation is a critical capability for successful operation in smart manufacturing environments. In the Robo-Erectus SML system, both the warehouse robot (SPlackie) and the workbench robot (PiPER) are required to perform manipulation tasks. To manage these operations, a customised Behaviour Tree (BT) architecture is implemented to coordinate task validation, object detection, and the execution of pick-and-place actions.

The detailed manipulation architecture for SPlackie is illustrated in Figure 6. Within this framework, object information is obtained from the perception module, while the TF2 library is used to transform the detected object pose from the camera coordinate frame to the robotic arm base frame. This transformation enables MoveIt to accurately plan and execute the required arm trajectories, ensuring precise manipulation and improving the overall efficiency and reliability of the system.

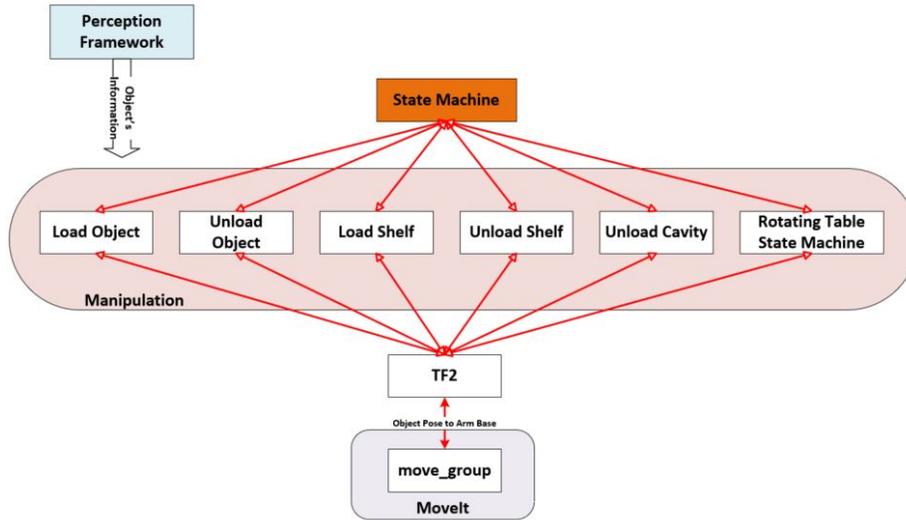


Fig. 6. SPLackie Manipulation Architecture

### 3.3 Perception Framework

The perception framework enables the robot to visually perceive, identify, and localize objects and various elements within the arena using a vision system enhanced by deep learning techniques. Operating in dynamic environments, where lighting conditions may vary and surfaces or virtual walls may introduce complexity and pose significant challenges for reliable perception. To address these challenges, both SPLackie and PiPER integrate a deep learning based perception module utilizing the **YOLOv8** model [6] [7]. This model supports multiple detection tasks, including object detection, cavity detection, container identification, barrier tape recognition, and hand sign detection. Known for its high processing speed, strong accuracy, and efficient learning capability, YOLOv8 provides a robust solution for real-time object detection in robotic applications.

With the introduction of a framework to explore human-robot interaction (HRI), which enables cooperative task execution such as performing assembly or disassembly at a workbench and reorganizing task sequences, the team developed additional Coworker Assembly Test (COT) objects. The number of COT objects was expanded from one to three, along with the inclusion of three hand signs and one card sign for interaction. The three COT objects are assembled from components listed in previous RoboCup@Work Competition object sets: COT\_1 (Spacer, M20\_100 and M20), COT\_2 (Axis2 and M20), and COT\_3 (Bearing2 and Housing). The interaction classes for the hand and card signs are defined as Shake, Nod, Okay, and Card. Examples of the detection results are illustrated in Figure 7.

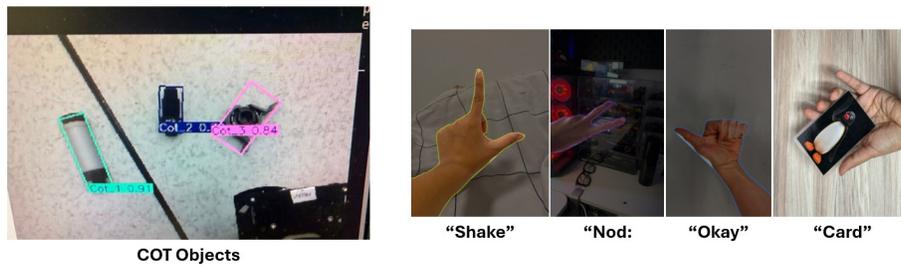


Fig. 7. (L) COT Objects Detection, (R) Hand and Card Detection

## 4 Human- and Multi-Robot Interaction

Having gained experience in the Technical Challenge from previous competitions in the Coworker Assembly Test (COT), the team has successfully demonstrated the capability to interact with the mobile robot, SPLackie, using hand signs for command and task coordination. Building on this achievement, the team is now expanding the interaction framework to include a humanoid robot using the Booster T1 robot, Sam. Through this development, human operators and robots can communicate through intuitive gestures and signals, allowing tasks such as assembly guidance, task sequencing, and collaborative manipulation to be carried out more efficiently within a shared workspace. Refer to Figure 8 below that shows human-robot and multi-robot interaction in a shared workspace.

This advancement supports the long-term vision of the Smart Manufacturing Lab (SML) [8] in creating a smart manufacturing environment where humans and multiple robots can collaborate seamlessly. In such a setting, robots are not only autonomous agents but also responsive coworkers capable of understanding human instructions and coordinating with other robotic systems. By integrating mobile robots, humanoid platforms, and human hand sign-based communication, the system aims to enhance flexibility, safety, and productivity in industrial operations while demonstrating scalable human–multi-robot interaction for future manufacturing applications.



Fig. 8. (L) Human-Robot Interaction, (R) Multi-Robot Interaction

## 5 Conclusion

In this paper, we presented the technical design and system architecture of Robo-Erectus SML, developed for participation in the RoboCup 2026 under the RoboCup Industrial Smart Manufacturing League (SML). The proposed system integrates multiple robotic platforms, including a warehouse mobile manipulation robot, a humanoid robot, and a workbench robotic arm, to address various tasks in a smart manufacturing environment. By leveraging a modular software architecture based on the Robot Operating System (ROS), the system combines perception, navigation, manipulation, and human-robot interaction capabilities to support flexible and reliable operation. Advanced perception using deep learning and behaviour-based task coordination further enhances the robot's ability to operate in dynamic industrial scenarios.

Future work will focus on improving system robustness, optimizing manipulation and navigation performance, and strengthening human-robot collaboration in cooperative manufacturing tasks. Continuous development and testing will be conducted to enhance perception accuracy, task planning efficiency, and overall system integration across the different robotic platforms. Through participation in RoboCup 2026, the Robo-Erectus SML team aims to validate its research outcomes, contribute to the advancement of intelligent robotic systems for smart manufacturing, and provide valuable learning opportunities for students involved in the project.

## 6 Acknowledgement

The team would like to express its sincere appreciation to both past and present project members who have contributed to the development of the SPLackie robot.

The team also acknowledges the dedication and support of the current students and technical staff from the School of Electrical and Electronic Engineering at Singapore Polytechnic. Special thanks are extended to the institution's senior management for their continuous support, which has been vital to the ongoing growth and advancement of Team Robo-Erectus SML.

## References

1. Willow Garage, "Robot Operating System", <https://www.ros.org>, 2007
2. Mintasca, GLUON-6L3, <http://wiki.mintasca.com/>, 2019
3. Booster Robotics, T1 Model, <https://www.booster.tech/booster-t1/>, 2023
4. AgileX Robotics, PiPER, <https://www.booster.tech/booster-t1/>, 2025
5. M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI: An Introduction*. Boca Raton, FL: CRC Press, 2018.
6. Glenn Jocher, "YOLOv8 in PyTorch", <https://github.com/ultralytics/ultralytics>, 2023
7. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", p. 779-780, 2016
8. Marco Masannek, Christoph Steup, Asad Norouzi, Wataru Uemura, "Embodied Autonomous Intelligence for On-Demand Manufacturing: A RoboCup Industrial Workshop Proposal", p. 10 , 2026