4^{th} Edition

Emirates Robotics Competition At Mohammed Bin Rashid Library Phase 1 - Simulation Phase

This document is a companion to the Overview guide and provides the technical detail needed to develop, test, and submit a solution for Phase 1 of the competition.

1 Setup

1.1 Getting Started

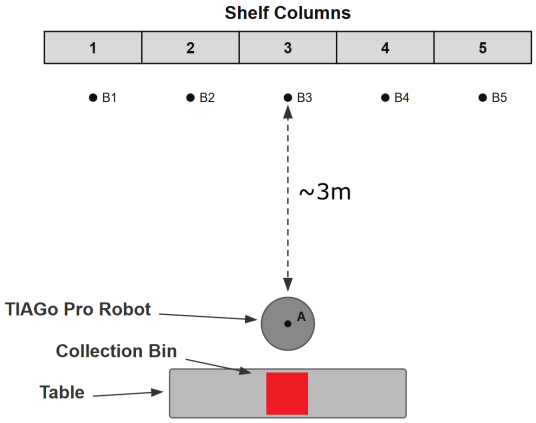
The ERC 2026 Docker Image and all competition packages are hosted on the competition GitHub repository. To get started, navigate to the latest release in the ERC 2026 repository in the link below, download the `.zip` file and follow the setup instructions in the `README.md` file included in the repository.

`https://github.com/dfl-rlab/erc_sim_2026`

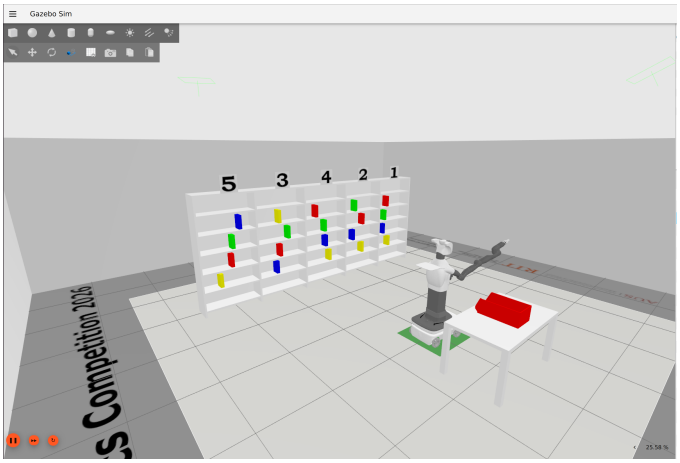
1.2 Virtual Environment

Figure 1 shows the virtual environment used for the Simulation phase consisting of:

- **Shelves Unit**: The shelves unit contains five shelf columns and six shelf rows and is roughly 3 meters from the starting point. All shelf columns and rows are fixed and reachable by TIAGo Pro. To identify the target shelf column, a numerical marker (1-5) is placed overhead and is randomized upon each load of the simulation environment. This is to ensure that all team solutions use computer vision to identify the target shelf column. The shelves unit's dimensions are 210 (h) x 525 (l) x 30 (d) cm.
- **Books**: Only the middle 4 rows of the shelves unit will contain books, giving a total of 20 books (5 columns $\times$ 4 rows). Books are represented in four distinct colours that appear exactly once per shelf column: **red, blue, green, and yellow**. The position of each book is randomised horizontally between 25–75% of the row's width, and the colour assigned to each shelf row is randomised vertically, meaning the same colour will not always appear on the same row between runs. All books are reachable by TIAGo Pro and fit in the 2-fingered PAL parallel gripper (end effector). A single book weighs 300g and its dimensions are 25 x 16 x 2 cm.
- **Collection Bin**: A single red collection bin is placed on a table at a fixed height reachable for TIAGo Pro. All books should be placed into this bin to attain points. The collection bin's dimensions are 50 x 31 x 21 cm and the table it sits on is 140 x 80 x 73 cm.
- **Start/End Zone**: An area marked in green and 0.8 x 0.8 m wide where the robot starts and ends each trial. The collection bin is reachable from here given the robot is rotated to the correct orientation.



(a) 2D top-down representation of the virtual environment that will be used for the Simulation phase



(b) 3D virtual environment in Gazebo that will be used for the Simulation phase

Figure 1: Visualizations of the environment for the Simulation phase

Note that point B in Figure 1a is determined by the target shelf column and book described in Section 1.5, and that the starting pose of the robot may not be facing the shelf as shown in Figure 1b. Teams must therefore ensure that the robot is correctly orientated at the start of their trial and can navigate to the correct locations using its onboard sensors and vision.

1.3 Hardware Platform

Teams will use a TIAGo Pro mobile manipulator platform by PAL Robotics for this competition which contains the components described below.

- **Mobile Base:** 4 mecanum wheels providing full omnidirectional movement with a maximum speed of 1 m/s forwards and backwards, and 0.7 m/s in diagonal movements.
- **LiDAR Sensors:** 2 LiDAR sensors (front and rear) providing a combined 360° field of view with 10 m detection range.
- **Lifting Torso:** 1 DoF torso with 35 cm of vertical travel, allowing the arm to reach objects at different heights.
- **Head:** The head has 2 DoF (pan/tilt) with an RGB-D camera (colour and depth) and a 10-inch touchscreen display mounted on it.
- **Arms:** Two 7 DoF arms with a vertical reach of 92 cm and a combined horizontal reach of 236 cm, capable of carrying a payload of up to 3 kg. Equipped with a 2-fingered PAL parallel gripper suitable for grasping the book objects used in this competition. For this challenge, participants must use only one arm to complete the task.
- **Robot PC:** Intel i5 processor, 16 GB RAM, 512 GB SSD, running Ubuntu LTS with ROS 2 and the PAL SDK pre-installed.

Teams are **not permitted** to modify the hardware or attach additional sensors. All development must be at the software level using ROS 2. For full technical specifications, refer to the official TIAGo Pro datasheet.

1.4 Software

Teams will be provided with an **ERC 2026 Docker Image** containing the full software environment required to develop and test their solution. The image includes:

- Ubuntu 22.04 with ROS 2 Humble.
- Gazebo Harmonic with the competition environment pre-loaded and a simulated TIAGo Pro model with identical ROS 2 interfaces to the physical robot (same topics, controllers, and sensor outputs).
- A base launch file that brings up TIAGo Pro, all of its sensors and the simulated world.

```
ros2 launch erc_bringup simulation.launch.py
```

- RViz 2 for real-time visualisation of sensor data and robot state.
- Nav2 and MoveIt 2 are pre-installed for TIAGo Pro in the Dockerfile.
- Camera topics published at:
  - For the RGB: `/head_front_camera/head_front_camera/color/image_raw`
  - For the depth: `/head_front_camera/head_front_camera/depth/image_rect_raw`
  - For the pointcloud: `/head_front_camera/head_front_camera/depth/color/points`

Teams are not permitted to modify the software environment of the provided Docker image, including the operating system, ROS 2 distribution, Gazebo version, or robot model. Additional packages such as IDEs may be installed inside the container as needed however, any submission that depends on modifications to the environment, or does not run correctly inside the provided Docker image, will not be evaluated. Teams are therefore strongly advised to test their solution on multiple machines before submitting.

1.5 Challenge Trials

The team's solution must be launched using the following command, where `shelf_column_number` and `book_colour` are passed as arguments:

```
ros2 launch YOUR_TEAM_PACKAGE_NAME solution.launch.py shelf_column_number:=2 book_colour:=red
```

Note that package name YOUR_TEAM_PACKAGE_NAME above is a placeholder and should be replaced by the name the team chooses for the solution package. The shelf column numbers are randomized each time the simulation is loaded and as such, `shelf_column_number:=2` could lead to a different location upon each load. Therefore, localization is entirely vision-based and teams must make sure they can identify the number "2" as part of their solution. The team's launch file `solution.launch.py` will be evaluated on the ability to complete the following in a single *challenge trial*:

1. Start TIAGo Pro from the Start/End Zone.
2. Use computer vision to identify the location of the target shelf column that the user entered as a parameter.
3. Autonomously navigate to the target shelf column and position TIAGo Pro within reach of the books.
4. Identify the target book using computer vision.
5. Pick the target book using the TIAGo Pro gripper.
6. Autonomously return to Start/End Zone with the book.
7. Identify the collection bin using computer vision and place the book inside it.

The base launch file `simulation.launch.py` must be running before the team's `solution.launch.py` file can be launched. The organizing committee will use the above launch command to test all submissions, so teams must ensure their launch file accepts these two arguments correctly. Any solution that does not accept these arguments correctly will not be evaluated.

2 Phase 1 Deliverables

In the first phase of the competition, all registered teams will develop and test their software using the provided **Gazebo simulation environment** in the Docker image. Teams must submit the following **deliverables** by the Simulation phase deadline shown in the `ERC 2026 Overview` document. The organizing committee will then evaluate all submissions using the qualification rubric in Section 3.2, and the top-performing teams will advance to Phase 2.

The *Phase 1 Deliverables Submission Form* will be provided to participants closer to the Phase 1 deadline.

- **Deliverable 1: Code**
  - Submit a GitHub repository link via the submission form.
  - Must contain all ROS 2 packages including the team's solution package that contains the `solution.launch.py` launch file and a `README.md` with information on the packages the team created for the solution.
  - Must include all dependencies that the team's solution depends on in the package.xml file.
- **Deliverable 2: Video**

  - Submit a YouTube link via the submission form with a maximum of 5 minutes.
  - Must be an unedited run completing the tasks described in the Scoring Criteria in Section 3.1.
  - Team name must be visible before the trial begins.
  - Must begin by launching simulation.launch.py then solution.launch.py.

- **Deliverable 3: Report**
  - The report must be a maximum of 5 pages (excluding title page) and submitted as a PDF file via the submission form.
  - The report must follow the structure defined in Section 3.2.

3 Scoring Criteria and Qualification Rubric

3.1 Scoring Criteria

The scoring criteria defined in Table 1 will be used during both phases of the competition. Points are awarded progressively as the robot completes each stage of the task. Total scores have a floor of 0 meaning that teams cannot finish with a negative score. A timer will begin for each challenge trial when `solution.launch.py` is launched and will end once the target book is in contact with the collection bin (this can be detected through the ROS 2 topic `/bin_contacts`). In case of tied scores between two or more competitors, the **fastest time** to complete all tasks listed below will serve as the tie-breaker.
Note that points awarded for identifying the correct shelf column/row are cumulative, therefore publishing to the ROS 2 topic awards +1 point, and saving an annotated image awards a further +2 points for a maximum of +3 points for that task. Points awarded for placing the target book in the collection bin are mutually exclusive as shown in Table 1.

Behaviour	**Points**
Identifying the correct shelf column location	
By publishing to ROS 2 topic	+1
By saving an image with a bounding box around the target shelf column	+2
Navigating to the correct shelf column (points awarded upon successful book grasp)	+3
Identifying the shelf row (1-4) that contains the target book	
By publishing to ROS 2 topic	+1
By saving an image with a bounding box around the target book	+2
Grasping and picking up the target book	+3
Navigating to the collection bin with the book in-hand (points awarded upon successful book delivery)	+3
Placing the target book to the collection bin	
Dropped into the collection bin	+2
or	
Gently placed into the collection bin	+4
Collision with shelf or other objects (per collision)	-0.5

Table 1: Challenge Scoring Criteria

Publishing to ROS 2 topics
For the tasks above that include publishing to ROS 2 topics, teams must publish to the following topics using the specified message types:

- Shelf column identification: publish to `/erc/shelf_column_identification` using `std_msgs/msg/Int32` with the column number (e.g. `2`).
- Row identification: publish to `/erc/shelf_row_identification` using `std_msgs/msg/Int32` with the row number (e.g. `3`).

These topics will be monitored by the organizing committee during evaluation to award points to teams.

Saving Images
Annotated images must be saved by the team's ROS 2 node during the challenge trial using live camera feed data to a designated folder named `/erc_images/` in the team's repository. Images edited manually or produced outside of the trial run will not be accepted. A timestamp must be included in each image for the organizing committee to verify that it was captured during the trial.

Collisions
A collision is any unintended physical contact between TIAGo Pro (its mobile base, torso, arm, or gripper) and the shelves unit, books, table, or collection bin that is not a deliberate part of the intended grasping or placing action. Contacts between the gripper and the target book are not penalized as those are intentional. Continuous contact with the same object counts as a single collision event (-0.5 points), as long as contact remains unbroken. A new penalty can only be triggered once contact has fully stopped and a separate contact occurs, with a cooldown of 1 second between penalties on the same object.
Collisions are detected by contact sensors in Gazebo during the Simulation phase, and in the physical round by judges' direct observation.

3.2 Qualification Rubric

Advancement from the Simulation Phase to the Physical Deployment Phase will be determined by a combined score out of 100%, comprising the performance of the team's solution and their submitted deliverables (Code, Video, and Report). The organizing committee will rank all registered teams by their qualification rubric score, and the top-performing teams will qualify to advance. The rubric is weighted as follows:

Component	**Weight**
Evaluation Challenge Trial	50%
Deliverable 1: Code	20%
Deliverable 2: Video	10%
Deliverable 3: Report	20%

Table 2: Qualification Scoring Weights

Evaluation Challenge Trial (50%)
The organizing committee will clone the team's submitted GitHub repo and launch the team's `solution.launch.py`. The best of **three attempts** will be taken as the team's final score. The points attained during this trial will constitute the team's score for this component, calculated according to the Scoring Criteria in Section 3.1. The evaluation trial will run under the same conditions as their submitted video and in the same environment and the solution must perform consistently and without manual intervention.

Code (20%)
The code must be able to replicate the results of the submitted video on any machine. Evaluation based on:

- Organization of the code submitted, including comments explaining functions/sections of code.
- Modularity of code (whether perception, planning, and execution are separated into distinct nodes, or contained in a single file).
- Error handling of the code (whether the system detects and recovers from failures e.g. missed grasps, lost topics, or unexpected sensor input).

Video (10%)
Evaluation is based on:

- The solution successfully completes all tasks required for this challenge.
- Each step of the task is clearly shown with no cuts or edits.
- Video should not be sped up, include a timer on screen when beginning the challenge trial until finish.
- The video must begin by launching `simulation.launch.py` followed by `solution.launch.py`.

Report (20%)
The report must follow the section below in order for the organizing committee to evaluate.

1. **System architecture:** a diagram showing how the ROS 2 nodes are connected and their roles. Describe how the perception, navigation, and manipulation nodes are connected to complete the challenge.
2. **Perception:** how the team detects the shelf column number and identifies the target book colour and why this approach was selected. Use quantitative data to show how accurate detection is across a number of trials.
3. **Navigation:** how the robot navigates to the correct shelf column and collection bin. Use quantitative data to show how travel time to key locations is minimized. Show screenshots of the navigation path planned vs executed in RViz.
4. **Manipulation:** how the robot grasps and places the book, including the approach used to position the arm and gripper relative to the target book, and how the placement into the collection bin is executed.
5. **Results:** a discussion of outcomes including quantitative data such as scores attained across 5 simulation trials, detection and grasping success rates, and average time to complete a trial.
6. **Limitations and Future Work:** what challenges were faced and how they were addressed, and what improvements would be made to the code given more time.

4 Revision History

Version	Date	Author	Description
1.0	05/08/2025	Organizing Committee	Initial draft of Phase 1 document