

RoboTeam Twente Extended Team Description

Paper for RoboCup 2025

Adham Elhabashy^{1,3}, Amirreza Akrami^{2,3}, Crina Gurev^{1,3}, Deef van der Linde^{1,3}, Dylan Bruggeman^{2,3}, Farida Elsadany^{1,3}, Flim de Jong^{1,3}, Hamza Elkady^{1,3}, Jonathan Jeuring^{1,3}, Lu Na Izquierdo Collantes^{1,3}, Marijn Buitenweg^{1,3}, Raswanth Vetrivel^{1,3}, Reyhan Iri^{1,3}, and Yuying Ba^{1,3}

¹ University of Twente (UT), Enschede, the Netherlands

² Saxion University of Applied Sciences, Enschede, the Netherlands

³ RoboTeam Twente, Capitoool 25 Enschede, the Netherlands

info@roboteamtwente.nl

<https://roboteamtwente.nl>

Abstract. RoboTeam Twente continues its participation in the Small Size League of RoboCup, pushing the boundaries of robot performance and control. This year, advancements include the integration of feedforward control, incorporating static friction effects, along with an improved friction estimation method that accounts for directional dependencies. Trajectory planning has been enhanced through the use of Bang-Bang trajectories with jerk-limited acceleration, leveraging the Ruckig library for smoother motion control. These developments aim to increase the precision, adaptability, and efficiency of the robotic system developed by RoboTeam Twente.

Further improvements include the adoption of larger motors and a complete overhaul of control code generation using MATLAB and Simulink for automated deployment. These innovations collectively contribute to a more robust and intelligent robotic soccer system, advancing RoboTeam Twente's competitive edge in RoboCup 2025.

Keywords: RoboCup · Feedforward Control · Friction Estimation · Simulink Code-Gen · Bang-Bang Trajectories · Motor Performance Evaluation

1 Introduction

RoboCup is an international robotics competition that fosters advancements in artificial intelligence and autonomous robotics, with the ultimate goal of developing a fully autonomous team capable of beating human soccer players by 2050. RoboTeam Twente competes in the Small Size League (SSL), which emphasizes high-speed, precise motion control, strategic team coordination, and intelligent decision-making.

Over the past year, RoboTeam Twente has focused on improving control and motion planning. By introducing feedforward control mechanisms that account

for static friction and other dynamic factors, the robots can achieve more precise velocity tracking. Additionally, refining trajectory planning through jerk-limited Bang-Bang motion reduces wheel slippage and enhances motion stability.

This paper details the technical advancements and innovations implemented in RoboTeam Twente's 2025 robot design. Improvements in motion control through friction estimation and feedforward compensation, the benefits of integrating Simulink-generated control code, and the effects of jerk-limited trajectory planning are all discussed. Additionally, a transition to more powerful motors is highlighted. These developments collectively aim to enhance the performance, reliability, and competitiveness of RoboTeam Twente in the upcoming RoboCup competition.



Fig. 1: Render of the 2024 version of the robot.

Table 1: Robot Specifications

Robot Version	v2024	v2025
Dimension	∅179 x149mm	∅179 x149mm
Driving motor	ECXFL32L 48V 50 Watt	TBD
Dribbling motor	Maxon DCX19S EB SL 24V	Maxon DCX19S EB SL 24V
Wheel diameter	56 mm	66 mm
Wheel gear ratio	1:1	1:1
Encoder driving motors	MILE 2048IMP	TBD
Dribbling bar diameter	14 mm	14 mm
Dribbling bar length	70 mm	70 mm
Encoder dribbler bar	ENX10 EASY 1024IMP	ENX10 EASY 1024IMP
Dribbler gear ratio	5:3	5:3
Microcontroller	STM32F767ZI	STM32F767ZI
Ball sensor	Custom Infrared sensor	Custom Infrared sensor
Motor driver	TI DRV8323SRTAR	TI DRV8323SRTAR
Inertial Measurement Unit	Xsens MTi-3-8a7g6t	Xsens MTi-3-8a7g6t
Battery	6S1P 22.2V 150C LiPo	6S1P 22.2V 150C LiPo
Kicker-and-chipper-board Capacitor	1000 μ F; Working voltage 200V	1000 μ F; Working voltage 200V
Wireless Communication	SX1280 2.4GHz	SX1280 2.4GHz

2 Friction Estimation and Feedforward Control

2.1 Static Friction Estimation

To accurately model and compensate for static friction in the robot’s movement, a steady-state test was conducted. This test was designed to measure the relationship between the applied motor voltage and the resulting body velocity while ensuring that the robot maintained a constant reference velocity.

Steady-State Test The steady-state test involved commanding the robot to move at various velocities and directions while maintaining a fixed yaw angle. The reference velocity, represented by its magnitude ρ and direction θ , was sent to the robot in a global coordinate frame. The actual movement is measured in the robot’s local coordinate frame. Understanding the transformation between these frames is crucial for analyzing the test results accurately. The **global coordinate frame**, shown in Figure 2, is fixed to the center of the field:

- The **x-axis** points towards the right goal.
- The **y-axis** is perpendicular to the x-axis, forming a right-handed coordinate system.
- A positive angle is defined in a counterclockwise direction.

This frame is used to define the robot’s commanded velocity (ρ, θ) before being transformed into its local frame.

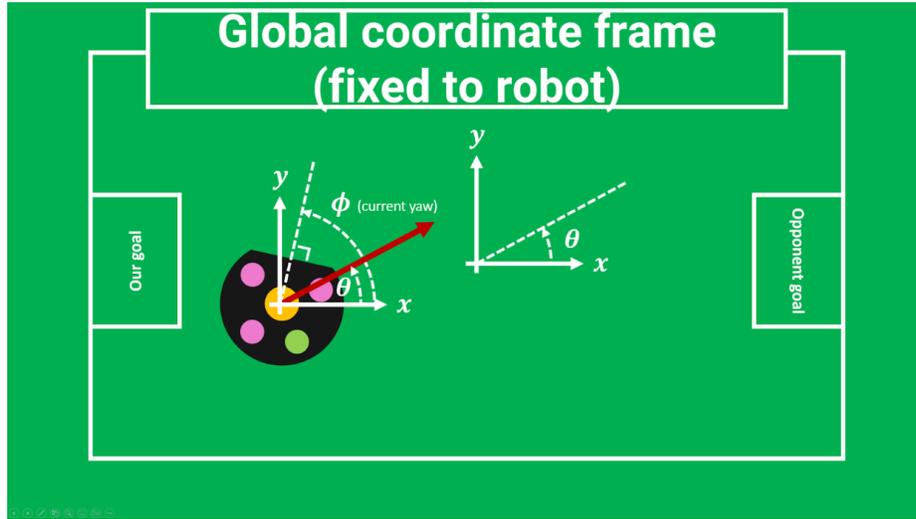


Fig. 2: Global coordinate frames used for velocity commands and measurements.

The **local coordinate frame**, shown in Figure 3, is attached to the robot:

- The **x-axis** points forward in the direction the robot is facing.
- The **y-axis** points to the left side of the robot.
- The transformation between the global and local frame depends on the robot's yaw angle (ϕ).

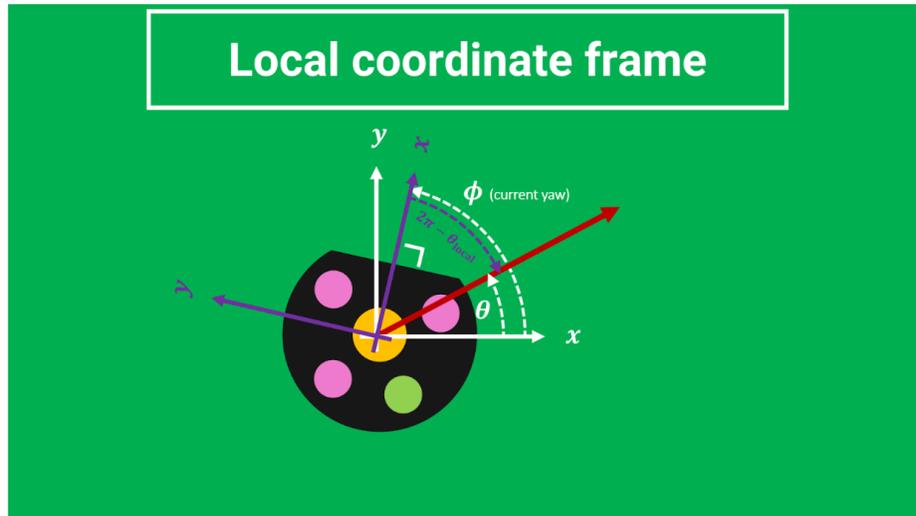


Fig. 3: Local coordinate frames used for velocity commands and measurements.

The test was structured into cycles, where the robot alternated between homing to a predefined position and executing velocity commands.

During each test cycle, the following data was recorded:

- Wheel encoder readings, converted to wheel speeds.
- Vision-based velocity measurements, considered the ground truth.
- Robot-reported velocity estimates, incorporating slippage compensation.
- PWM duty cycles applied to each motor.
- Battery voltage to compute the actual motor voltage.

Each velocity command was maintained for a fixed duration to ensure the system reached steady-state behavior. The recorded data was post-processed to determine the relationship between motor voltage and wheel speed, allowing for friction and damping estimation, see Figure 4. The colorcoding is shown in Figure 5 and used for any subsequent figure.

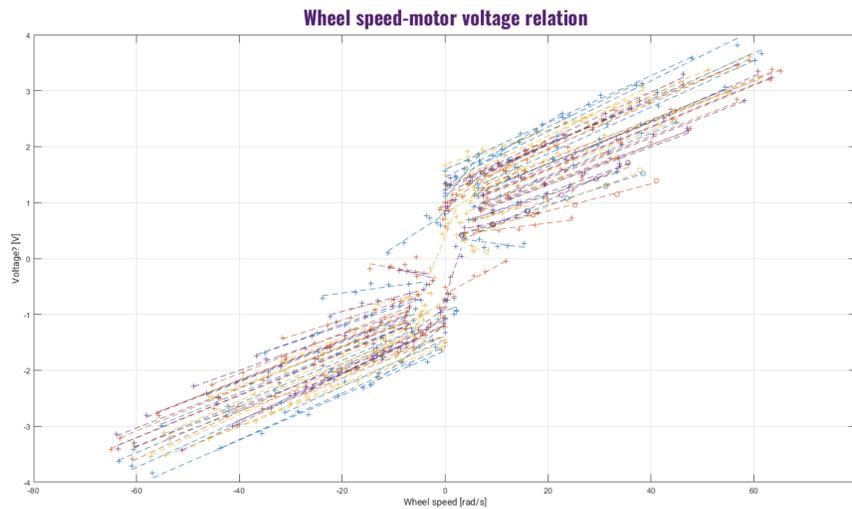


Fig. 4: Static friction offset values for different driving angles.

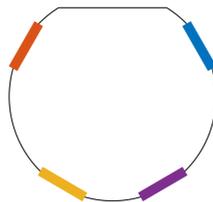


Fig. 5: Color coding of the wheels

So the wheels are numbered in the following way: $\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}$

Friction Estimation The relation between motor voltage and wheel speed was analyzed by plotting the steady-state speed of each wheel against the estimated voltage applied. The motor voltage was computed as:

$$V_{\text{motor}} = \text{PWM} \cdot V_{\text{battery}}, \quad (1)$$

where V_{motor} is the estimated voltage applied to the motor, PWM is the normalized duty cycle, and V_{battery} is the measured battery voltage.

Each wheel's data points were fitted linearly, revealing a characteristic offset in voltage when extrapolated to zero speed. This offset represents the static friction, which depends on the driving direction in global coordinates. Figure 6 shows the offset values for different driving angles. It can be noted that the sign of the offset changes with the driving direction of the respective wheel.

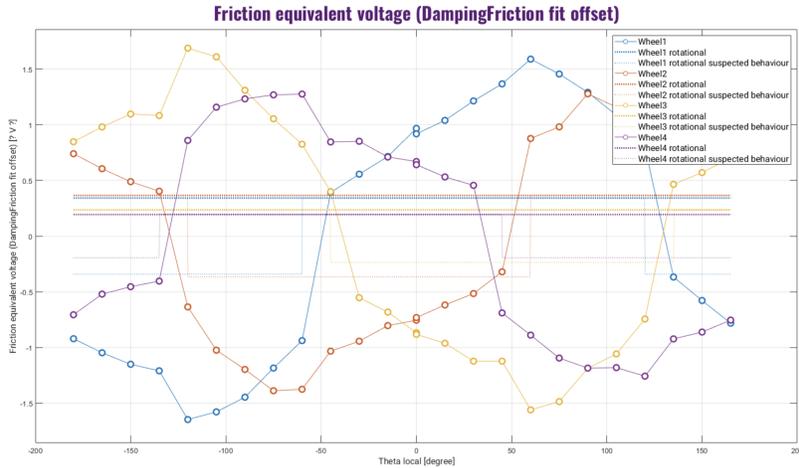


Fig. 6: Static friction offset values for different driving angles.

2.2 Feedforward Control

To compensate for the estimated friction and damping experienced by the wheels, a feedforward control strategy was implemented. By incorporating both static friction and damping feedforward terms, the final feedforward voltage command for each wheel is given by:

$$\begin{aligned} V_{\text{ff}} &= V_{\text{damping}} + V_{\text{friction}} \\ &= d \cdot \omega_{\text{ref}} + f(\theta) \end{aligned} \quad (2)$$

where d is an experimentally determined damping coefficient, ω_{ref} is the reference angular velocity of the wheel, and $f(\theta)$ is a fitted function representing the friction offset as a function of the driving angle θ .

This ensures that the robot compensates for resistive forces before relying on feedback control, leading to more accurate velocity tracking.

Friction Compensation For the friction compensation, different fitting approaches were considered, including a simple sign-based model and sinusoidal approximations. Ultimately a combination was used resulting in a general formula of:

$$f(\theta) = a \cdot \sin(\theta) + b \cdot \text{sign}(\sin(\theta)), \tag{3}$$

where a and b are determined experimentally. The function $f(\theta)$ was determined by fitting the experimental data, as illustrated in Figure 7.

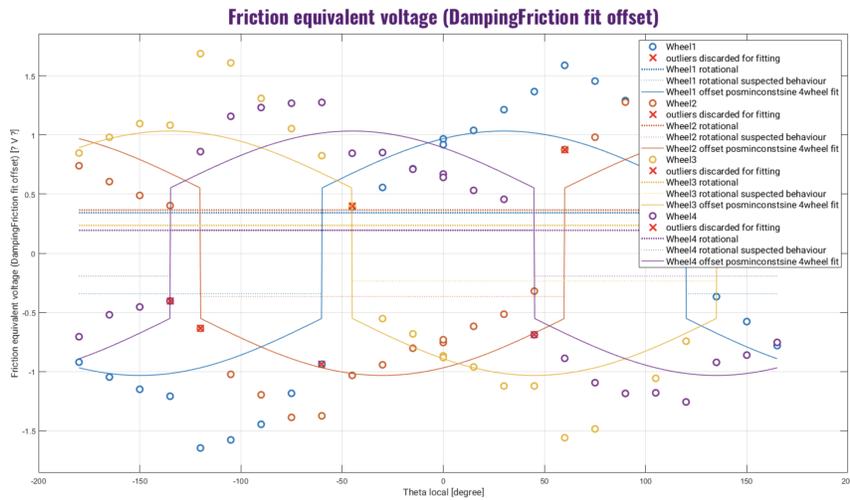


Fig. 7: Fitted function for static friction compensation.

Damping Feedforward A damping feedforward term was introduced, approximating the damping effect as a linear relation between wheel velocity and required voltage. Outliers caused by unintended wheel movement were excluded, and an average damping coefficient was computed. To explain, a non zero value point is considered an outlier if the respective wheel should have zero rotational velocity for the given θ based on the robot’s geometry. Figure 8 shows the estimated damping coefficients versus θ .

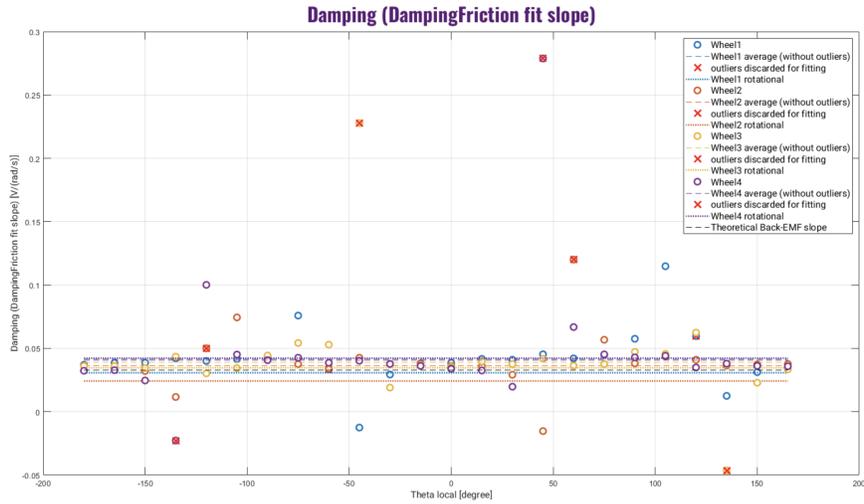


Fig. 8: Estimated damping coefficients for different driving angles.

2.3 Effectiveness of the Feedforward Control

After implementing the feedforward control, noticeable improvements were observed, particularly in low-speed reference tracking. Static friction compensation significantly enhanced performance. Further testing is required to determine the exact improvements.

3 Simulink Code-Generation for Control

The control of the propulsion of the robot was thusfar implemented in pure embedded C-code. This results in a moderately efficient (though far from well-optimized), and largely functional, control subsystem. There are, however, a few key areas in which this approach left more to be desired:

- There is no high-level overview of the system unless this overview is actively maintained.
- Refactoring is time-consuming and potentially error-prone
- Control and modelling are difficult to integrate

In previous years it was noticed that this approach led to a relatively slow development iteration cadence, and a similarly slow onboarding process. In an attempt to avoid these problems, a workflow that utilizes Matlab Simulink to generate the control code has been integrated. Preliminary results suggest good controller performance with a reduction in code complexity and iteration time.

4 Jerk-Limited Bang-Bang Trajectories Using Ruckig

4.1 Motivation for Jerk-Limited Acceleration

Traditional acceleration-limited Bang-Bang [1] trajectory planning maximizes efficiency by applying the highest allowable acceleration and deceleration. However, RoboTeam Twente’s recent robots experienced significant wheel slippage with this approach, necessitating a refined method. A viable solution was found in jerk-limited motion. Jerk, defined as the rate of change of acceleration, plays a key role in ensuring smooth motion transitions.

- **Slippage:** Sudden acceleration changes led to significant wheel slip.
- **Overshooting and Oscillations:** Due to slippage, robots frequently overshoot targets and oscillated before stabilizing.

To mitigate these issues, RoboTeam Twente integrated jerk-limited trajectories using the Ruckig library, which ensures smoother transitions between acceleration phases, thereby reducing slippage [2].

4.2 Implementation with Ruckig

Ruckig provides real-time trajectory generation by incorporating third-order kinematic constraints, specifically limiting jerk j , alongside acceleration a and velocity v constraints. Unlike acceleration-limited Bang-Bang approaches, Ruckig ensures smooth transitions between motion phases, minimizing slippage.

Given an initial state \mathbf{x}_0 and a target state \mathbf{x}_f , where:

$$\mathbf{x} = [p \ v \ a], \quad (4)$$

with p as position, v as velocity, and a as acceleration, Ruckig computes a time-optimal trajectory satisfying:

$$v_{\min} \leq v(t) \leq v_{\max}, \quad (5)$$

$$a_{\min} \leq a(t) \leq a_{\max}, \quad (6)$$

$$j_{\min} \leq j(t) \leq j_{\max}, \quad (7)$$

for all $t \in [0, T_f]$, where T_f is the trajectory duration. If the initial acceleration or velocity exceeds these limits, Ruckig quickly corrects it.

Figure 9 illustrates the relationship between jerk, acceleration, velocity, and position over time.

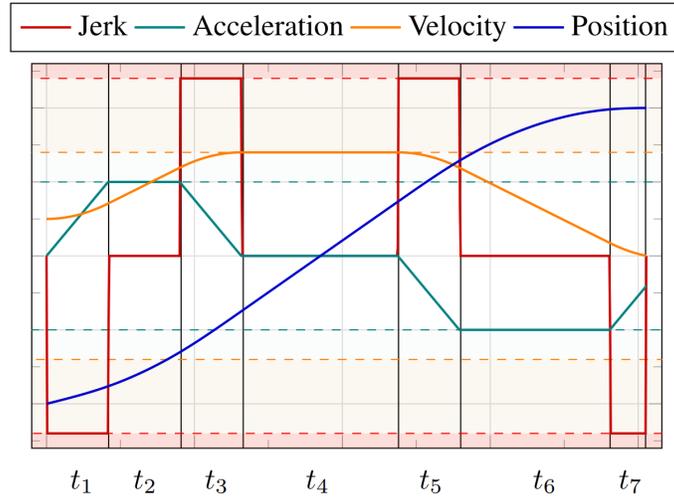


Fig. 9: Visualization of jerk-limited trajectory motion phases as functions of time [2].

These 1D Bang-Bang trajectories extend naturally to 2D trajectories, as explained in

4.3 Effects on Path Planning and Tracking

Incorporating jerk-limited motion into the Bang-Bang trajectory planner improves path tracking and control in the following manners:

- **Reduced Slippage:** Smooth acceleration transitions reduce wheel slip.
- **Better Path Feasibility:** Acceleration-limited approaches sometimes generate paths that robots cannot follow due to excessive slip. Jerk-limited control ensures dynamically feasible paths.

This method has not only reduced slippage but also enhanced motion predictability for the robots currently used by RoboTeam Twente.

4.4 Testing and Validation

RoboTeam Twente conducted iterative tests on its indoor field. While no objective tests were performed, practical observations indicated:

- **Slippage Significantly Reduced:** Robots followed paths with improved precision and less slip.

- **Lower Wear on Omniwheel Subwheels:** Fewer subwheels detached, suggesting reduced mechanical stress, though this remains to be verified if this is directly the result of the jerk limiting.

Since maximum jerk values were tuned through observation, future research is necessary for quantitative validation.

4.5 Applicability to Other Teams

This approach is not universally advantageous. RoboTeam Twente’s motors could not instantly reach maximum acceleration, making jerk-limiting beneficial. However, teams with more powerful motors may not require it. For teams facing slippage issues, this method presents a viable solution.

General guidelines for evaluating the need for jerk-limiting include:

- **Motor Response Time:** Faster motors may not benefit significantly from jerk-limiting.
- **Friction Conditions:** High-friction wheel designs reduce slippage issues.
- **Control System Constraints:** Controllers with high update rates may compensate for acceleration spikes without explicit jerk constraints.

This limitation motivated developments in Section ?? (??) to integrate motors capable of directly achieving maximum acceleration.

4.6 Conclusion

Replacing acceleration-limited Bang-Bang trajectory planning with a jerk-limited approach using Ruckig reduces slippage, leading to smoother and more stable motion. This method serves as a good alternative to the standard acceleration-limited Bang-Bang trajectory planning for any team that experience the same problems with regards to slippage.

5 Hardware Design and Implementation

The hardware consists of all the physical components of the robot. This is divided into mechanical and electronic parts. In recent years, RoboTeam Twente has been working on creating a more modular and robust hardware design [3], [4]. This year's team continued on this path by completing the redesigns of the hardware to improve the reliability of the robot. Details of the new design can be found on the wiki of RoboTeam Twente, and the new design is also shown in Figure 10.

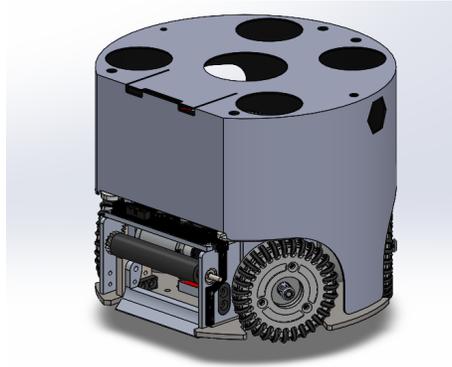


Fig. 10: View of the v2025 robot

5.1 Mechanical Design Improvements

The reliability of the hardware has been a recurring challenge over the years. Last year, the mechanics subteam successfully redesigned the entire robot to enhance its reliability [4]. This year, the subteam focused on further improvements, with the most significant changes involving motor verification and the resulting redesign of the bottom assembly for larger motors. It was found out that bigger motors are needed to address the overheating issue experienced by previous year. This will be further explained later on in this report. Additionally, due to less time required for major redesigns compared to last year, the team was able to conduct research as well. These changes are detailed below.

Motor Performance Evaluation Last year, the team transitioned to a direct-drive system using the Maxon ECXFL32L motor [4]. However, during RoboCup, these motors failed when playing on a Division A field. At that time, the cause of failure was unclear, and no immediate solution was found. Following discussions with Maxon, temperature calculations were done to verify the operating conditions and determine whether the motors exceeded their specifications.

The analysis begins with the equation for I_{RMS} (Equation 8) and proceeds to compute the stator temperature (Equation 11) and winding temperature (Equation 12):

$$I_{\text{RMS}} = I_{\text{ON}} \cdot \sqrt{\frac{t_{\text{ON}}}{t_{\text{ON}} + t_{\text{OFF}}}} \quad (8)$$

$$\Delta T_W = \frac{(R_{\text{th1}} + R_{\text{th2}}) \cdot R_{TA} \cdot I_{\text{RMS}}^2}{1 - \alpha_{Cu}(R_{\text{th1}} + R_{\text{th2}}) \cdot R_{TA} \cdot I_{\text{RMS}}^2} \quad (9)$$

$$\Delta T_s = \frac{R_{\text{th2}}}{(R_{\text{th1}} + R_{\text{th2}})} \cdot \Delta T_W \quad (10)$$

$$T_s = T_{\text{ambient}} + \Delta T_s \quad (11)$$

$$T_W = T_{\text{ambient}} + \Delta T_W \quad (12)$$

Using these equations, the temperature evolution of the stator and windings over time was determined to assess compliance with motor specifications. The results, shown in Figure 11, indicate that the winding temperature of the current motors exceeded the specification of 155°C, reaching 162°C.

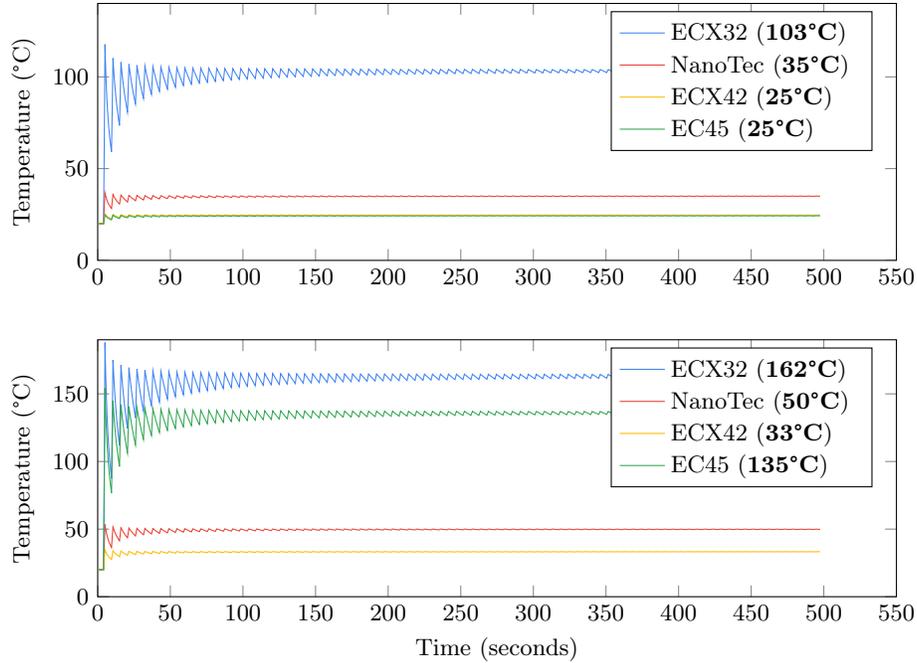


Fig. 11: Temperature of the Stator (top) and Windings (bottom) over Time

Given these findings, the team explored alternative motors better suited for the robot's application. Although the final selection is still under investigation,

the bottom assembly and motor mounts were redesigned to support modifications. The new motor mount accommodates both the Maxon ECX Flat 42M and the Nanotec DF45L024048-A2, while the bottom plate redesign allows for future integration of motors up to $\varnothing 43\text{mm}$.

Structural and Frame Modifications To accommodate the larger motors, several structural modifications were necessary:

- The kicker PCB was flipped to resolve interference with capacitors that no longer fit between the larger motors.
- The power board was relocated to fit between the capacitors.
- The top plate was adjusted accordingly to accommodate these changes.

Additionally, smaller refinements were implemented:

- The skirt was updated to address tolerancing issues with the new, larger wheels.
- Cable management systems were revised to improve accessibility and organization.
- Mounting strategies for the top PCB were altered aiming to reduce noise issues in IMU data.

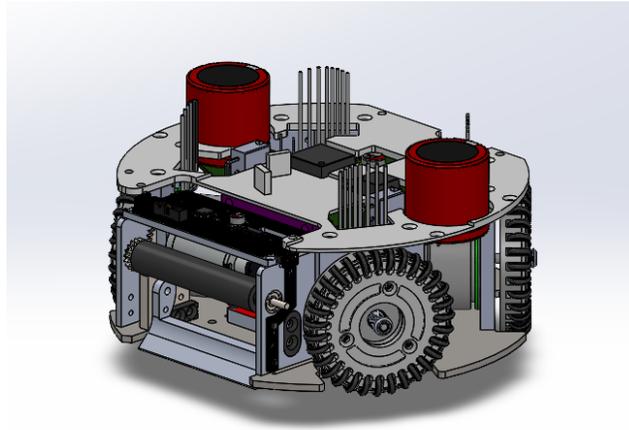


Fig. 12: View of the redesigned Bottom Assembly

Ongoing Research and Development Two research projects were initiated:

1. **Front Assembly Damping:** Investigating methods to quantify ball-handling improvements in linear motion. Current evaluations of damping systems rely on subjective observations, and this study aims to develop a standardized testing framework.

2. **Kicker Add-on Shape:** Initially explored but put on hold due to minor performance improvements and time constraints.

The damping research is entering its testing phase and is expected to yield results by RoboCup 2025.

References

1. Nicolai Ommer, A.R., Geiger, M.: Tigers mannheim extended team description for robocup 2019 (2019), <https://tigers-mannheim.de>
2. Berscheid, L., Kröger, T.: Jerk-limited real-time trajectory generation with arbitrary target states. arXiv preprint arXiv:2105.04830 (2021), <https://arxiv.org/abs/2105.04830v2>
3. Monat, C., Dankers, E., Skurule, K., Steenmeijer, L., Sijtsma, S., Diamantopoulos, S., Aggarwal, R., Smit, T., van Harten, A.: Roboteam twente extended team description paper for robocup 2022 (2022), https://ssl.robocup.org/wp-content/uploads/2022/04/2022_ETDP_RoboTeam-Twente.pdf
4. van der Kaaden, B., Krommendijk, C., Buzogany, C., Capitani, D., Ayad, G., Ward, G., Kroijenga, J.G., Willigers, J., de Jong, J., González Torres, J.J., Muchembled, J., Hoekstra, L., Winters, L., Petrea, M.R., Mahmoud, M., Srinivasa Kumar, P.K., Roetenberg, R., Tollardo, S.: Roboteam twente extended team description paper for robocup 2024, https://tdpsearch.com/#/tdp/soccer_smallsize__2024__RoboTeam_Twente__0?ref=list