The A-Team Team Description Paper 2026

M. Barulic, C. Clark, R. Osawa, J. Spall IV, W. Stuckey, N. Witten, M. Woodward

The A-Team

Abstract. "The A-Team" is a community team from the United States participating in our fourth year of RoboCup SSL. Our mission is to practice technical skills, inspire the communities in which we are present, and stay connected with friends around the world. This paper presents the work done since our 2025 TDP. Focus is placed on ongoing hardware upgrades, improvements to our motion control solutions, and learnings from RoboCup 2025.

1 Mechanical

1.1 Dribbler

Experimentation on dribbler designs have continued to improve ball stability and impact absorption. Work on the pressure-gradient–based dribbler was temporarily paused to prioritize a shorter-term, competition-ready solution. As an alternative, a more traditional, roller-based dribbler incorporating a dampening mechanism was investigated.

Results are still mostly qualitative, but preliminary tests show notable improvements over a static dribbler solution. Key benefits provided by adding a predominantly translational degree of freedom include:

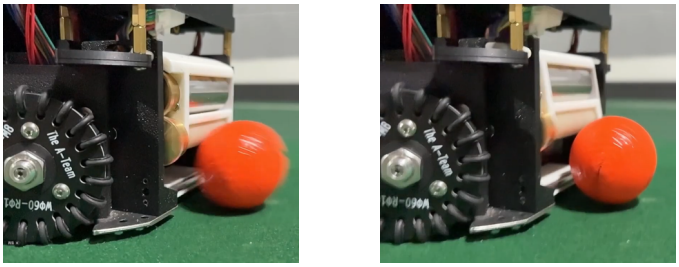
- Lower impulse due to the extended collision time
- Longer contact time enables the roller to apply sufficient backspin, ensuring that even after rebound, additional collision opportunities are created, which can further dissipate energy over multiple cycles
- Energy dissipation occurs in both the compressing and relaxing directions

With the current setup shown in Figure 1, friction within the system alone is capable of dissipating enough energy from a ball rolled off of a 1m tall ramp (velocity estimated around 4 m/s). Testing efforts will continue in order to better identify the relevant parameters and geometries for consistent ball control.

2 Electrical

2.1 Radio

The team has been using a 5 GHz Wi-Fi module, the ODIN-W262 from u-blox since our first event in 2023 [21]. Previous TDPs characterized radio latency and



(a) Extended State (b) Retracted State

Fig. 1: Dribbler Range of Motion

data rate, which allowed us to easily offload data for debugging and opened a pathway for advanced communication features without the presence of an operating system [1,2]. The team has generally been happy with the radio. However, this module has been designated "not for new designs" by the manufacturer, meaning it is nearing "End of Life" designation.

The team has a designed a new radio PCB using the NORA-W36 module from u-blox [22]. In many ways it is the next generation ODIN. The new module offers several key benefits over the ODIN, of particular interest to the community. Notable improvements are a much better SWaP, slightly higher effective data rate, the ability to force Wi-Fi channels and region regardless of environment and location, and a substantial cost reduction. This comparison is shown in Table 1. The main improvements seems to stem from the lack of an automotive and commercial export license (which require regional frequency detection and compliance). This doesn't matter for SSL teams. SWaP improvements can be seen in Figure 2.

There are two practical downsides with the new module: it requires an external antenna connector and it is more difficult to solder. Unlike the previous ODIN, a breakout PCB with impedance controlled layers is required for the antenna. Despite the higher manufacturing cost for the PCB, the cheaper radio module leads to a lower overall radio cost. While the ODIN module mostly uses castellated edge holes with some small grounding points underneath, the NORA-W36 is entirely land grid array. We successfully soldered several modules, all on the first try, with careful application of solder via an iron, and then hot air reflow with hobby level equipment.

(a) ODIN-W26X Assembled PCB (b) NORA-W36X Assembled PCB

Fig. 2: Radio Module Physical Comparison

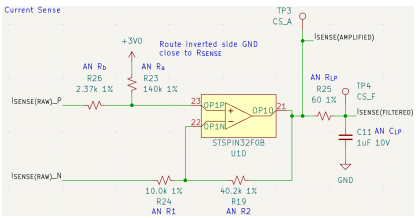
Table 1: Radio Module Properties

Metric	ODIN-W26X	NORA-W36
Size (mm)	22.30 x 14.80	14.30 x 10.40
Area (mm^2)	331	149
Weight (g)	2	1
Power (mW)	2640	2640
Cost (USD)	$49	$11
Transparent Data Rate (kb/s)	1000	2500
Non-Transparent Data Rate (kb/s)	450	900
Force Channel Region	No	Yes
External Antenna Option	No	Yes
Automotive Certification	Yes	No

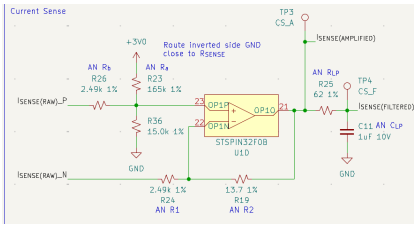
2.2 Current Sensing Updates

In order to do more precise control, current sensing improvements were required. Initial testing with last year's board appeared to have unexpected offsets and amplification error. This led to re-evaluating our existing filter circuit using guidance from STMicroelectronics [15].

Because the STSPIN has so few ADC pins that can be synced with sampling with the motor driver circuit, we used a single shared shunt resistor topology. Our circuit matches the "Amplification network schematic for bipolar current sensing" [15], but with a DC offset to only read positive current from 0-10 A, with direction gleaned from our hall sensors. After further simulation and measurement, we discovered a resistor missing on the positive op-amp channel which would have impact on both the bias. Additionally, R23 and R36 should have Thevenin equivalence to the feedback resistor R19 to reject the common mode gain, with the effect being dynamic gain error. With the missing resistor and values adjusted, much more expected values based on ground truths from fixed current loads indicated success. A comparison can be seen in Figure 3.



(a) Current Sensing v1.0 ECAD with bias and Thevenin error



(b) Current Sensing Corrected in v1.1 ECAD

Fig. 3: Current Sensing v1.0 vs v1.1 ECAD

Additional motor controller firmware improvements also improved the usability and accuracy of the measurements when actually driving the motors. This included replacing floating-point with fixed-point calculations during PID control and extracting motor currents, adding USB-OTG offloading for plotting, adding the ability to calibrate motor-specific current bias offsets, improving the 6-step commutation to enable more closely coupled current measurements during commutation, and adding the option to command target currents and voltages for the actual motor control. All of these firmware and hardware changes allowed us to more accurately measure and visualize the current to enable the usage in the Torque Control work below.

3 Motion Control

3.1 Torque Control

The team has previously used wheel-level velocity or gain-scheduled PID motor controllers. Compared to torque-based control, these approaches have a slower response time and higher error rate. Additionally, without torque limiting, wheel-level feedback control may command a wheel torque above the static friction between the wheel and carpet, leading to wheel slip. When this occurs, recovery becomes challenging and the local deviation from the desired trajectory is substantial. Tuning multiple gain stages and hysteresis also proved difficult to do quickly in new environments. Having precise torque (and therefore force) control will help mitigate these issues. A later section describes how wheel velocity and torque setpoints are produced. This section covers the approach to achieve the setpoint.

3.1.1 Architecture

Developing a motor current controller generally has the same challenges and tradeoffs as the design of any controller. Bandwidth (and response time) should be minimized while avoiding negative behaviors such as ringing, ripple, or outright instability in operating corners. A few key top level parameters and limits can give us preliminary bounds we may want to satisfy. The architecture, shown in Figure 4 starts from the widely established starting point of constructing a velocity PID loop above a current/torque PI controller. Now specific parameters need to be chosen.

Bandwidth Target We want to maximize bandwidth to maximize response performance. The size of packets transmitted between the control board and the motor controllers bounds the velocity loop rate around 1.2 kHz. The Nyquist rate limits the theoretical bandwidth to 600 Hz. This provides a rough limit on the desired current loop bandwidth. Most sources recommend 5-50x separation of loop rate or bandwidth between nested loops. This is impractical at the upper boundary. A 50x separation on 600 Hz velocity BW, 30 kHz current BW, would require a PWM frequency in the hundreds of kHz, which would create massive switching losses and is incompatible with our microcontroller. The low end is

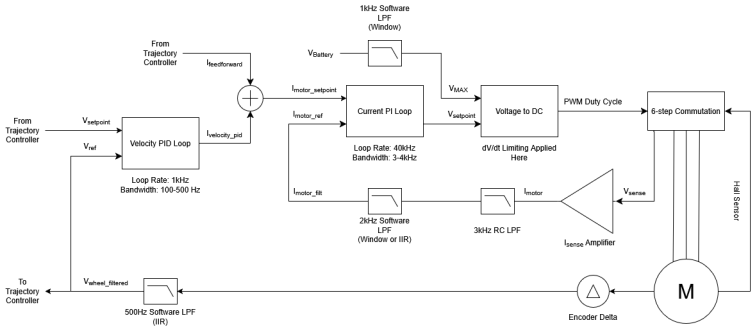


Fig. 4: Wheel Feedback Controller Architecture

feasible. A 3 kHz current controller bandwidth would require PWM frequency around 6 kHz. Unfortunately, practical implementations contain details that produce a more realistic theoretical bound of 1/6th to 1/10th based on sampling variance and noise [3,24]. Most sources practically suggest 1/20th, so there is additional margin in controller stability. This is advantageous since the electrical domain is more difficult to observe on a live system due to challenges in attaching instrumentation or offloading bulky data. With a maximum PWM frequency in the hardware of 40-50 kHz, we get a new maximum current bandwidth of 2.5 kHz, at 1/20th separation. This would also suggest that a velocity loop bandwidth of 600 Hz is borderline, although some of the same practical limits on current feedback (like filter delay) will impact the upper loop as well.

Filter Selection Filters help reduce ringing and ripple due to anomalous or noisy data. Motor current data is infamously noisy, so we'll want to intelligently select hardware and software filters to reduce negative effects without a impacting current loop bandwidth. It's important to recall that filters introduce lag, which erodes the control loop phase margin. If too much phase margin is lost, the control loop may become conditionally stable, or unstable. As such, one needs to select both in tandem. Given a previously established current controller target bandwidth of 2.5 kHz, we can safely select a hard RC low pass filter (LPF) with -3 dB cutoff of 3 kHz, which gives some wiggle room. This does not have computational overhead. A second discrete filter is implemented in firmware. This gives the designer some wiggle room to adjust the overall filter. The current prototype uses a 8 slot window averaging filter for the discrete software filter. The bandwidth is computed as $f_c = F_s/N$, or $f_c = 40000/8 = 5$kHz [7]. This can be made more aggressive by increasing the number of samples in the average to 16 or 32. Powers of two provide computational efficiency in division by bit-shifting.

Now the blocks of the architecture diagram in Figure 4 have details. A general summary of the tradeoffs are available in the table in Figure 2

3.1.2 Tuning

Theoretical and seemingly sane bounds were placed on the hardware and firmware controller design. The designer now has to tune the controller to achieve the

Table 2: Current Controller Tradeoffs

Parameter	**Effect**
Controller Bandwidth ⇑	Response ⇑, Stability ⇓, PWM Frequency ⇑
PWM Frequency ⇑	Maximum Bandwidth ⇑, PWM Resolution ⇓
Controller Loop Rate ⇑	Maximum Bandwidth ⇑, Computation ⇑
Filter Bandwidth ⇑	Stability ⇑, Noise ⇑, Response ⇓

desired bandwidth (while maintaining stability), tune the firmware filters, and characterize the performance.

A challenge with tuning controllers, particularly in RoboCup where holonomic drives can have complex responses, is finding a valid starting point and comparing it with a theoretical model. In the case of current loop tuning, this is actually fairly feasible. Controlling current in a motor is predominantly a circuits problem with the coils being modeled as RL circuits, which have a current response lag. This means a low order model of the circuit can be used to calculate the current transfer function[10]. Analysis of (1) yields K_p and K_i in Equations (2) and (3)[10].

$$I(s) = \frac{K_p s + K_i}{Ls^2 + (R + K_p)s + K_i} \tag{1}$$

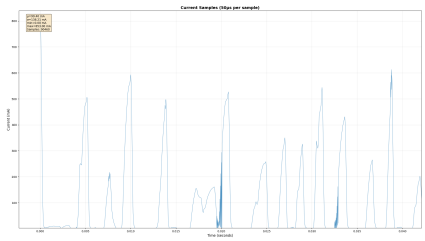
$$K_p = \omega_{-3db} L \tag{2}$$

$$K_i = \omega_{-3db} R \tag{3}$$

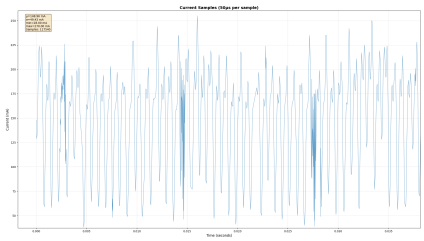
All of the information is now available to choose an initial tuning point. Our team uses the Nanotec DF45 50W Motor, whose datasheet will be needed to get the motor coil resistance $R = 0.700\ \Omega$, and inductance $L = 330\ \mu H$[8]. The goal now is to maximize ω_{-3db} while preserving stability. A hard-ish limit of 2.5 kHz was previously established. Some parameters are fixed for testing. Notably, the motor and wheel have no load (this choice will be analyzed later), and the input voltage is fixed at 25 V, near a 6S1P LiPo max voltage. Velocity loop is disabled. Current samples are captured at 20 kHz and offloaded to be analyzed and plotted. Analysis began with a 2 kHz bandwidth, seen in Figure 5a.

The steady state behavior at 2 kHz is very unstable. Oscillation is so profound the controller is clipping out of the valid current range into negative current range. A few issues could be causing the instability. Filter lag could be eroding stability, but we have provided a fair bit of margin for now. Noise could be getting though the filter and impacting stability. Double checking the noise in open loop control (no feedback for voltage setpoint) is fairly easy. This behavior is seen in Figure 5b.

The noise is fairly substantial. Easily ±50% of the mean, approximately 125 mA. One can also see periodicity of the signal around 1600 Hz. Where does the noise come from? The system reports a wheel velocity of 300 rad/s or 48 Hz. But, we have a 16 pole motor, so commutation would occur at 800 Hz. Almost an exact harmonic of 1600 Hz. The induced noise may be related to the



(a) Steady State Current Control: $\omega_{-3db} = 2000$Hz, f_{-3db}FW $= 5$kHz

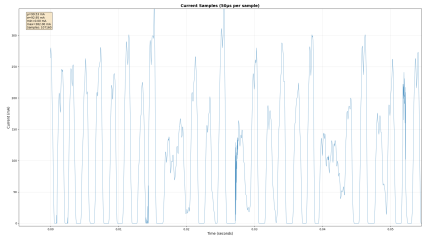


(b) 12500mV Open Loop Voltage Control - Current Capture

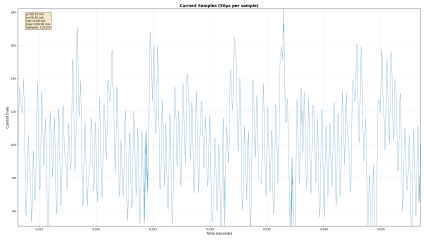
Fig. 5: Current Sense Initial Tuning and Noise

6-step commutation scheme we use, which is known for producing particularly bad current and torque ripple at low RPM [6,12]. Though the RPM is not very low, the jarring commutation may still be playing a factor. Additionally, the no load conditions mean voltage is high, but current is still low. The controller is producing high PWM swing to accommodate very small changes in current compared to the overall design range which accommodates up to 10 A of motor current (giving us some margin for ripple over the motor peak rating of 7 A). Next steps are to try to mitigate some of the noise with more filtering (which may push bandwidth down) or also to simply lower the bandwidth. For sanity purposes, an experiment was run with a PI controller target bandwidth of 2 kHz, and a firmware filter bandwidth of 1.25 kHz. The result is seen in Figure 6a.

The response is still very unstable. We'll lower the PI controller bandwidth target to increase stability. Let's try 1 kHz, seen in Figure 6b.



(a) Steady State Current Control: $\omega_{-3db} = 2000$Hz, f_{-3db}FW $= 1.25$kHz

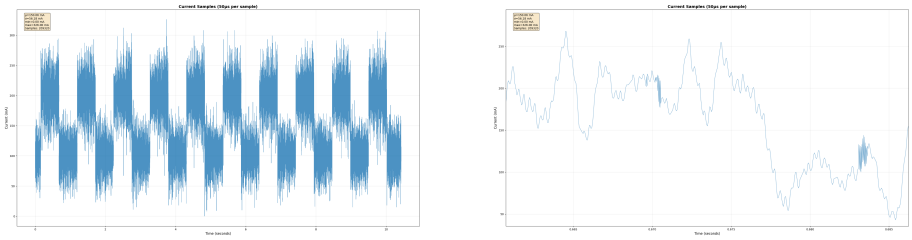


(b) Steady State Current Control: $\omega_{-3db} = 1000$Hz, f_{-3db}FW $= 1.25$kHz

Fig. 6: Secondary Tuning and Filtering Exploration

The response is much better and dramatic oscillation and clipping is gone. This could possibly be run, but the stability looks marginal or under-damped.

The decision was made to cut bandwidth again and test step response. A bandwidth of 500 Hz was chosen, and step response was confirmed to be stable at 3 voltage operating ranges, low: 1 V, mid: 12 V, and high: 20 V. An example of the macro view of the test is seen in Figure 7a. The graph was visually inspected for stability and response time. A zoomed in response of the 100-200 mA step test is seen in Figure 7b. The setpoint is achieved in about 2 ms, or 500 Hz, inline with the design bandwidth. Clipping and controller-induced oscillation are not observed. This manual inspection was done for each test. Additionally a 90% rise of ∼1 ms is seen. The theoretical rise time should be 700 μs, given by (4) which is within the margin of error or visual confirmation.



(a) Step Response 100-200 mA: $\omega_{-3db} = 500\text{Hz}$, $f_{-3db}\text{FW} = 1.25\text{kHz}$

(b) Step Response 100-200 mA Zoomed: $\omega_{-3db} = 500\text{Hz}$, $f_{-3db}\text{FW} = 1.25\text{kHz}$

Fig. 7: Final Tuning and Step Response

$$t_{rise} = \frac{0.35}{BW_{\text{Hz}}} \tag{4}$$

Next steps are to more rigorously process the data. The team will characterize current and control noise at varying setpoints and motor loads. This should include frequency and magnitude components, which facilitate the creating of a more specific control policy that addresses any noise that's non-uniform in frequency. Additionally, it's imprecise and laborious to plot and analyze response by eye. This is great for initial testing, but slow when trying to batch analyze data or analyze sweeps. It would be far more powerful to sweep bandwidth and filter parameters and plot summarized data. This would facilitate a more formal analysis of the performance of the system.

Long term, it will probably be beneficial for us to use Field Oriented Control (FOC) commutation for the reasons mentioned in [6] with respect to torque ripple. We will likely upgrade to the STSPIN32G4, so avoid having to use fixed point, challenges of which are described in the next section. It will also give the team an opportunity to improve current sampling precision and filtering with higher performance components and more computational overhead. This may allow us to unlock additional improvement in current/torque response. Notably,

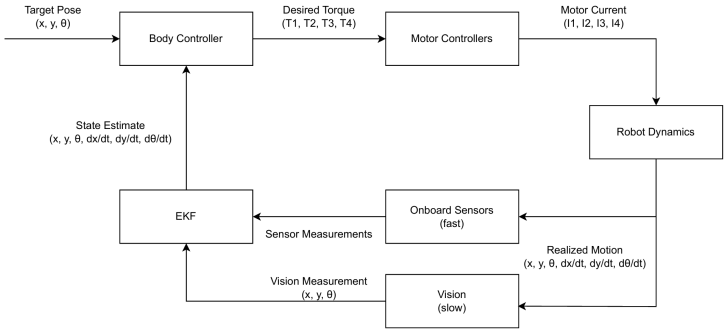


Fig. 8: On-Board Body Controller Architecture

Green Tea reported achieving a bandwidth of 2000 rad/s or 320 Hz [14]. This is slightly below our controller response, but their current noise looks substantially better. We'd like to thank Green Tea for a detailed write up and supporting data in their 2023 TDP.

3.1.3 Practicum

There are a few practical hurdles to tuning motor controllers. Fundamentally, motor current can't be observed easily like many mechanical responses. The designer cannot see oscillation, instability, or performance without instrumentation. On dynamic mobile robots, attaching instrumentation may impair movement. As such, we are often required to build custom tools to offload and process data from low level systems. This is often fairly straightforward in terms of embedded systems development, but it is time consuming. The time investment is worth it and in the long run helps accelerate system understanding. Our team's code is open source, including the tools to offload and plot real-time data from the motor controller [18].

Motor control designers also often face performance constraints. Running a PI controller and filtering at 10-40 kHz easily consumes all available computational power. In this system, it wasn't possible to use floating point numbers because the controller doesn't have a FPU, forcing the use of soft-float is 10-200x slower than integer math. The fixed point performance at -O0 was about 15 kHz. The fixed point performance at -O3 met the 40 kHz execution target. This effectively rules out using soft-float because fixed-point is so near the computation limit. A few resources were very helpful. TIGERs has a fixed point PI controller implementation in their repository [20] that is a good reference implementation. We also made frequent use of of the RF Wireless World fixed point conversion tool [13].

3.2 On-Board Body Controller

In past competitions, the A-Team has employed body-level velocity control of the robot. The Control Board firmware accepted x, y, and θ velocity commands from

the Coach Computer. It regulated the body velocity by running a PID controller with individual wheel velocity output and wheel velocity encoder measurements as feedback terms. This year, we have redesigned the body-level control approach with a bang-bang pose controller.

After reviewing the strategies described in the 2019 TIGERs TDP [9] and 2020 ER-Force TDP [23], we decided to re-haul our body-level motion control architecture to employ the near-time optimal bang-bang control described by [5,11].

Compared to previous years, the reformulation of state estimation and control effort output results in improved motion performance and accuracy. The control loop is capable of running at a faster 1 kHz frequency, fusing on-board sensor measurements and delivering wheel commands every millisecond. Moving these computations to the on-board firmware reduces lag and jitter as the high frequency telemetry and command channels no longer need to be communicated over radio.

3.2.1 Extended Kalman Filter State Estimation

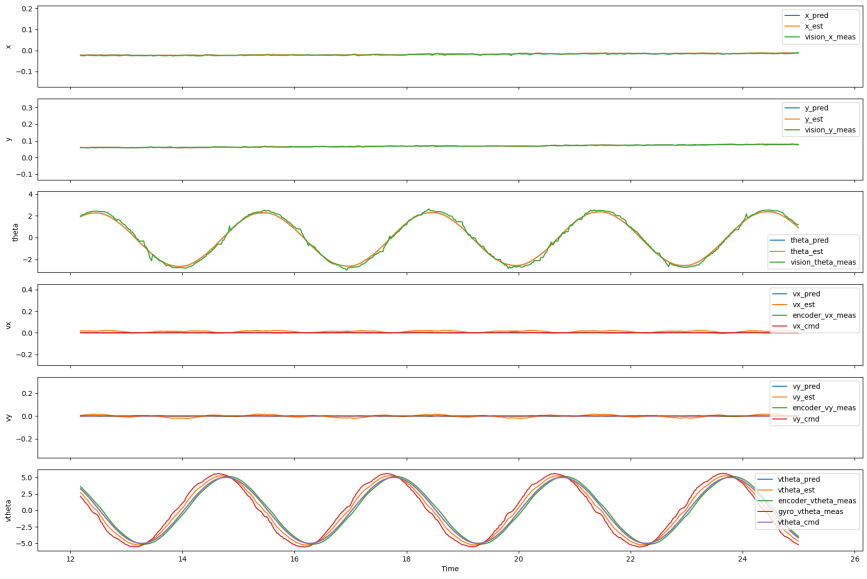


Fig. 9: Extended Kalman filter Tuning with Oscillating Theta

A strong motivation for moving global positioning to the firmware level is to achieve a better state estimate. By running an Extended Kalman filter on the robot control board, a higher frequency update can be achieved. The Kalman filter prediction step's variance is lowered proportionally to the square of the

Kalman filter update loop period and more sensor measurements are fused into the state estimate. An improved state estimate is critical for the performance of the body-level controller, as it calculates the desired trajectory from the current state.

The body controller formulates the state, control input, and sensor measurements as the following.

$$\begin{aligned} \text{State: } \mathbf{x} &= \begin{bmatrix} x\ y\ \theta\ \dot{x}\ \dot{y}\ \dot{\theta} \end{bmatrix}^T \\ \text{Control Input: } \mathbf{u} &= \begin{bmatrix} \ddot{x}\ \ddot{y}\ \ddot{\theta} \end{bmatrix}^T \\ \text{Measurement: } \mathbf{z} &= [vision_x, vision_y, vision_\theta, encoder_{fl}, encoder_{bl}, \\ &\qquad encoder_{br}, encoder_{fr}, gyro_{\dot{\theta}}]^T \end{aligned} \tag{5}$$

The vision measurements provide low-frequency absolute positioning, which mitigates the state from drifting. The state transition model simply integrates the time derivatives of the three dimensions. The non-linearity emerges from the encoder wheel velocity sensor measurements, which must be transformed from the local robot frame to the global frame using the current state estimate's θ to provide velocity state information.

The largest challenge in achieving this reformulation of on-board state estimation is streaming the vision measurements to the robot. Unlike the other sensors, vision measurements arrive at a lower frequency than the Kalman update rate. The measurements are only applied once and are masked in the Kalman filter observation model for the update steps that do not have a fresh vision measurement.

Figure 9 shows the Kalman filter's performance as the robot oscillates it's θ dimension. The model prediction, transformed measurements, and final state estimate are plotted together. Commanding the robot to open-loop oscillate the three dimensions x, y, and θ allows us to tune and test the filter via sensor and model statistical variances.

3.3 Trajectory Generation

Using the TIGERs firmware [20] as an implementation reference, the body-level bang-bang control strategy described in [5,11] was integrated into the A-Team firmware infrastructure. This control strategy simplifies the vehicle's nonlinear dynamics by mapping motion from the state space to the physical actuator space via an inverse dynamics model. By restricting the allowable control effort to an orientation-independent admissible set, the three-dimensional trajectory problem is reduced to independent one-dimensional bang-bang profiles. These profiles are then synchronized using an iterative search to optimize for minimal total travel time. The resulting framework is computationally efficient, enabling real-time execution where the generated trajectory is continuously transformed into instantaneous motor commands.

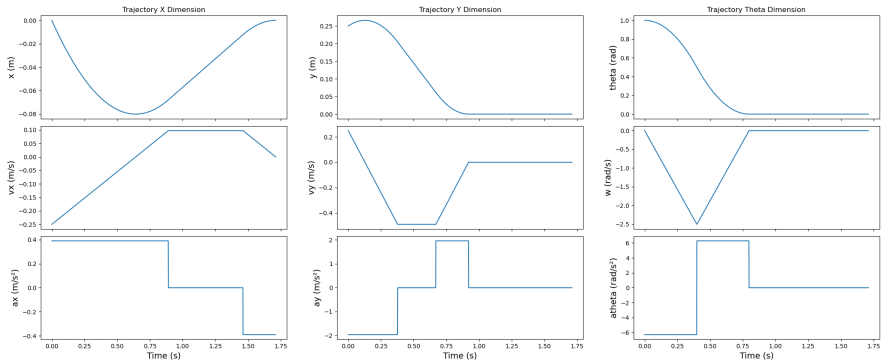
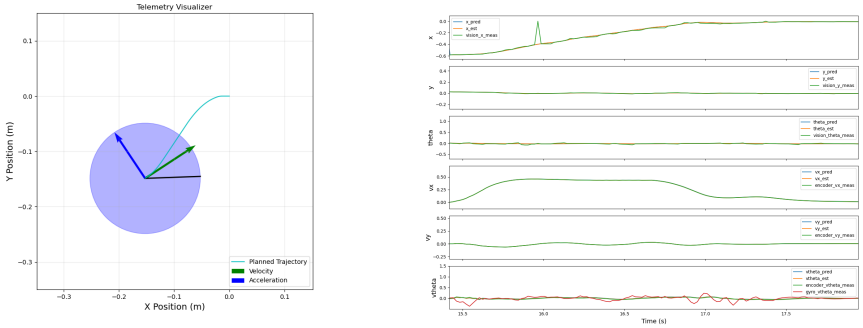


Fig. 10: Decomposed 3-Dimensional Bang-Bang Ideal Trajectory



(a) In-Motion Planned Trajectory (b) Realized Bang-Bang Trajectory

Fig. 11: Trajectory Planning Running on Robot Hardware

Figure 10 shows a near-time optimal bang-bang trajectory to the zero state, computed at an instant in time when the robot has a non-zero position and velocity. Note the bang-coast-bang accelerations which are then transformed into wheel torque commands according to robot geometry. These wheel commands are then immediately applied to the wheel motor controllers.

Figure 11a shows a planned trajectory to the zero state captured from a robot in motion. The instantaneous body acceleration command is shown in blue. The theta dimension is omitted for viewing.

The computational efficiency of the algorithm enables the recomputation of the trajectory from the current state estimate in the 1 kHz control loop. Figure 11b shows the results of the architecture when traveling from $x = -0.6$ m to $x = 0$ m. The realized bang-coast-bang profile can be seen in the x dimension. There are still improvements to be made to achieve a sharper response from the robot that more closely follow the ideal computed trajectories. This trajectory was captured from a robot utilizing wheel velocity control. The integration of motor torque control is expected to greatly improve the robot's dynamic response.

4 Software

Our software work in the past year has focused on continuing the spatial evaluation work described in our 2025 TDP and an architectural overhaul to our coach computer motion control software.

4.1 GPU-Enabled Pass Placement

In our 2025 TDP, we considered multiple approaches to pass receiver placement, including the "heatmap" approach which evaluates suitability of an even grid of positions across the entire field. Since then, we have rewritten the heatmap approach to make use of GPU parallel processing and were able to test the GPU-enabled implementation at RoboCup 2025. While we saw some success with this approach, we have ultimately decided not to pursue it further for reasons presented below.

4.1.1 Approach

Our GPU-enabled spatial processing module builds multiple heatmaps of the full field. These heatmaps are updated in every frame of our gameplay cycle (100 Hz) and queried by our plays to identify useful field positions by searching for extrema in one of these maps. Each map is built by mixing multiple layers. A layer is a simple, reusable spatial calculation, such as "distance from the ball." While our system architecture supports multiple maps targeting different use cases, our 2025 implementation features a single map: `ReceiverPositionQuality`.

The key ideas contributing to the performance of this system are persisting the heatmaps in GPU memory across multiple gameplay frames and leveraging GPU parallelization both in map generation and in queries. Full heatmap data

is only copied from the GPU to the CPU for visualization purposes, which is disabled during normal operation.

A single three-dimensional kernel updates all heatmaps at the same time at the start of the gameplay frame. Then, other parts of our gameplay code can query the contents of the heatmap. This generally takes the form of two tasks: querying the value of a map at a specific location, or finding the location of the min/max value on a map. Extrema searching was accomplished with a one-dimensional kernel to allow us to keep the heatmap in the GPUs memory.

The full source code for our GPU-enabled implementation is available in the *ateam_spatial* package at the *comp_2025* tag of our software GitHub repository [19].

4.1.2 Results at RoboCup 2025

The GPU-enabled spatial evaluation system was effective at generating high-resolution full-field heatmaps. The update kernel is able to generate a 2 cm resolution heatmap for receiver position quality in 1.3 ms on an NVIDIA RTX 4070 Mobile GPU. However, the system suffered from a few key deficiencies that ultimately hurt our competition performance at RoboCup 2025.

First, the GPU-enabled code was very brittle to other changes on our coach computer. Seemingly innocuous changes to computer settings dramatically impacted the performance of the GPU code. And moments before our last match, an unattended software upgrade broke our GPU driver installation rendering the device unusable and forcing us to fall back to older non-GPU dependent code.

The second major issue relates more to strategy than technical implementation. None of our heatmap layers accounted for locality to one of our robots. We were searching for the optimal place to pass the ball to regardless of how feasible it would be for one of our robots to reach that spot. This resulted in often selecting pass targets far from our robots. By the time our robots reached that location and executed the pass, it was no longer a very useful place for the ball to be.

After discussing this topic with members of the TIGERs Mannheim team at RoboCup 2025, we have realized that most of the full field heatmap for pass targeting is not worth considering as there are many portions of the field where no robot can get there in time to receive a quick pass. This key insight informed the pass strategy they describe in their 2025 ETDP [4], which we are adapting for our system this year.

4.2 Motion Planning Rearchitecture

Since RoboCup 2025, we have made substantial architecture changes to the motion pipeline of our coach computer software. This was motivated by two needs: making it easier for plays to express the intent of what they want robots to do, and making it possible to coordinate motion across the entire team in a more intelligent manner.

In the old system, a play was responsible for returning an array of command velocities for the team. All path planning and closed-loop control were left up to each play to handle. To help with this, various utilities were created to let plays share common implementations of frequently needed abilities. This resulted in many copies of the same utility objects being created to meet the needs of all skills, tactics, and plays that might run during runtime. Unfortunately, these instances had no way to coordinate with each other, so a lot of additional state checking had to be implemented to ensure no motion invariants were violated during cases such as switching from one play to another while robots were moving.

Ultimately, the weaknesses of this system directly contributed to the loss of our first knockout match of RoboCup 2025. Because the motion utilities tracked so much internal state, skills that wanted to change motion settings in different circumstances had to be careful about when they reset defaults and when settings changes carried over into other circumstances. Last minute changes to our goalie code accidentally introduced a settings reset to its motion controller specifically when the ball was moving toward our goal at high speed. This reset caused the goalie to start avoiding our default obstacle set, which includes both teams' defender boxes since most robots are not allowed to enter this area. This caused our goalie to try to exit the offending obstacle (our defender box) as quickly as it could, effectively running away from the goal exactly when it was supposed to be blocking an incoming shot.

The new architecture introduces the concept of a "motion intent," which describes what the robot should be doing at a higher level. Our system includes intents such as "move to this pose" and "move at this velocity". Linear and angular motion intents are represented independently. Plays now return motion intents for each robot and allow a central motion system to handle any path planning, closed loop control, and other motion needs. This dramatically simplifies the interface used in skills, tactics, and plays. Instead of managing their own copies of common motion components, they create and return simple structs of settings and targets that more directly express the desired motion.

This change reduces the likelihood of human errors when making changes to plays as there is no cross-frame state that needs to be considered at the play level. It also improves computational efficiency. Less memory is consumed as each STP object no longer needs to maintain its own copy of motion-related objects, and fewer operations are performed per gameplay frame as this eliminates throw-away motion work (such as running a path planner and later deciding to override the robot velocity directly). The final key capability this change unlocks is more intelligent coordinated motion between our robots. In the old system, there was no way for our robots to account for each other's motion when planning their paths. With the new, centralized motion system, the path planning step can be run for all robots at the same time, negotiating crossing paths while avoiding collisions with our own team.

While this change in approach may seem obvious to some teams, the old architecture was a natural result of our fast-paced, minimum-viable-product approach

to development in the early days of our gameplay software. The weaknesses were only fully realized now after several years of effective use. Given how impactful these weaknesses ended up being at RoboCup 2025, it feels pertinent to highlight this change for other newer teams who may be considering an approach similar to our old system.

5 Open Source

Mechanical, electrical, firmware, software, and control and circuit models are published on the team's GitHub page and licensed for broad use [16,17,18]. Competition versions are documented with known issues and production artifacts for full replication. Software currently supports the latest Ubuntu/ROS LTS. Mechanical, electrical, firmware, and controls currently support most modern operating systems.

References

1. Avidano, C., Barnette, S., Barulic, M., Medrano, L., Neiger, J., Osawa, R., Peterson, E., Spall IV, J., Spalten, J., Stuckey, W., Woodward, M.: The A-Team technical description paper 2023. RoboCup SSL TDP Archive (2023), https://ssl.robocup.org/wp-content/uploads/2023/02/2023_TDP_The_A_Team.pdf
2. Avidano, C., Barulic, M., Clark, C., Osawa, R., Spall IV, J., Stuckey, W., Woodward, M.: The A-Team technical description paper 2024. RoboCup SSL TDP Archive (2024), https://ssl.robocup.org/wp-content/uploads/2024/04/2024_TDP_The_A_Team.pdf
3. Dixon, L.: Control loop design. Unitrode Power Supply Design Seminar Manual (1996), https://www.ti.com/lit/ml/slup098/slup098.pdf
4. Geiger, M., Ommer, N., Ryll, A., Ratzel, M.: TIGERs Mannheim extended description for RoboCup 2025 (2025), https://ssl.robocup.org/wp-content/uploads/2025/04/2025_ETDP_TIGERs-Mannheim.pdf
5. Kalmár-Nagy, T., D'Andrea, R., Ganguly, P.: Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle. Robotics and Autonomous Systems (2002), https://seas.ucla.edu/coopcontrol/papers/02cn05.pdf
6. Lee, S., Lemley, T., Keohane, G.: A comparison study of the commutation methods for the three-phase permanent magnet brushless DC motor. Electrical Manufacturing Technical Conference 2009 (2009), https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/902/34.-A-comparison-study-of-the-commutation-methods-for-the-three-phase-permanent-magnet-brushless-dc-motor.pdf
7. Mathuranathan: Equivalent noise bandwidth (ENBW) of window functions. GaussianWaves - Signal Processing for Communication Systems (2020), https://www.gaussianwaves.com/2020/09/equivalent-noise-bandwidth-enbw-of-window-functions/
8. Nanotec: DF45 - brushless DC flat motor (2022), https://us.nanotec.com/products/1544-df45-brushless-dc-flat-motor
9. Ommer, N., Ryll, A., Geiger, M.: TIGERs Mannheim extended team description for RoboCup 2019 (2019), https://ssl.robocup.org/wp-content/uploads/2019/03/2019_ETDP_TIGERs_Mannheim.pdf
10. Pieper, J.: Auto-tuning current control loops. mjbots blog (2021), https://blog.mjbots.com/2021/04/13/auto-tuning-current-control-loops/
11. Purwin, O., D'Andrea, R.: Trajectory generation for four wheeled omnidirectional vehicles. Proceedings of the 2005, American Control Conference, 2005 (2005), https://ieeexplore.ieee.org/document/1470795
12. Rao, C.N.N., Sukumar, G.: Design and analysis of torque ripple reduction in brushless DC motor using SPWM and SVPWM with PI control. European Journal of Electrical Engineering (2018), https://share.google/jj5pMhEaOslkf5Utu
13. RFWirelessWorld: Floating point to fixed point converter (2026), https://www.rfwireless-world.com/calculators/floating-point-to-fixed-point-converter
14. Sato, H., Okamoto, N., Ito, A., Kayalo, S., Sugishita, N., Nakaaki, S., Nakao, T., Nishimura, Y., Hara, Y., Fujita, K., Yuri, R.: GreenTea 2023 team description paper (2023), https://ssl.robocup.org/wp-content/uploads/2023/04/2023_TDP_GreenTea2.pdf
15. ST Microelectronics: Current sensing in motion control applications (2019), https://www.st.com/resource/en/application_note/an5397-current-sensing-in-motion-control-applications-stmicroelectronics.pdf

16. The A-Team: 2024 Robot Design (2024), https://cad.onshape.com/documents/75f8983195cd6d66fab4fcd9
17. The A-Team: 2025 Robot Design (2025), https://cad.onshape.com/documents/2eeb704b914dc4eea021eab8
18. The A-Team: SSL-A-Team GitHub Organization (2025), https://github.com/SSL-A-Team
19. The A-Team: Software (2026), https://github.com/SSL-A-Team/software
20. TIGERs Mannheim: Firmware (2026), https://github.com/TIGERs-Mannheim/Firmware
21. u-blox: ODIN-W2 series (u-connect) - stand-alone IoT gateway modules with Wi-Fi and Bluetooth (2022), https://www.u-blox.com/en/product/odin-w2-series-u-connect
22. u-blox: NORA-W30 series - stand-alone dual-band Wi-Fi modules with Bluetooth low energy (2025), https://www.u-blox.com/en/product/nora-w36-series
23. Wendler, A., Heineken, T.: ER-Force 2020 extended team description paper (2020), https://ssl.robocup.org/wp-content/uploads/2020/03/2020_ETDP_ERForce.pdf
24. Zhang, H.: Understand power supply loop stability and loop compensation - part 1: Basic concepts and tools. Analog Devices Technical Articles (2022), https://www.analog.com/en/resources/technical-articles/power-supply-loop-stability-loop-compensation.html