RoboCup 2023 SSL Champion TIGERs Mannheim - Improved Ball Interception Trajectories

Michael Ratzel, Mark Geiger, André Ryll

Department of Information Technology
Baden-Württemberg Cooperative State University,
Coblitzallee 1-9, 68163 Mannheim, Germany
info@tigers-mannheim.de
https://tigers-mannheim.de

Abstract. In 2023, TIGERs Mannheim won the RoboCup Small Size League competition with individual success in the division A tournament, and the chip pass technical challenge. Given only one conceded goal during the Robocup 2023, in addition to the focus of last year's champion's paper on the offense parts of the strategy, this year will focus on an important improvement for the goalkeeper: An improved trajectory generator for goal shot interceptions. The current time-optimal second-order BangBang trajectories include a complete stop at the intercept destination, wasting valuable time. This is overcome by generating virtual destinations for the robot, that overshoot the intercept point and avoid the preliminary braking, while ensuring the keeper will reach the intercept point at the same time as the ball.

1 Introduction

RoboCup Small Size League (SSL) games stand out with their highly dynamic games and a fast pace. With 11v11 fully autonomous and omnidirectionally moving robots, fast decision-making and efficiently trajectories are a key component for success. This paper will focus solely on an extension to the current system for generating 2D trajectories that improves the goalkeeper performance when catching shots on the goal, as our last champion papers focused on the hardware development of our robots [1] and the offensive part of our strategy [2]. The currently generated second-order trajectories are time-optimal with constant acceleration and include a full stop to reach zero velocity at the specified destination position. However, this full stop wastes valuable time when intercepting goal kicks, and as long as the goalie and the ball are at the interception point at the same time, it is irrelevant that the goalie comes to a full stop at the intersection point in a time-optimal manner.

Since the TIGERs robots are controlled only by specifying the final destination at which the robot should stop, we present an extension algorithm that can provide virtual destination positions such that the actual destination is reached at a given time by an actual trajectory created to the virtual destination. Since this problem is overdetermined, as the robot is not always able to reach the destination in the given time, a best possible virtual position is generated that brings the robot as close as possible to the destination in the given time.

2 Related Work

The SSL is a fast paced robot soccer league, and due to the high power of the motors compared to size and weight, the omnidirectional drive system of the robots is mainly limited by friction. Therefore, the Cornell Big Red team has presented an approach that generates second-order time-optimal trajectories with constant acceleration and a complete stop at a given target destination [3]. The current implementation is based on this approach and is discussed in more detail in section 3.

The work of Hove et al. [4] presents a problem similar to intercepting a goal shot: Catching a ball with a robotic arm. However, they must impose more stringent requirements on the trajectory of the arm's end effector as they attempt to match the position, velocity, and acceleration of the ball at the interception point to reduce the risk of the ball bouncing off the arm. For future offensive applications, such as receiving a pass with a moving robot, these requirements may also be imposed, but for simply intercepting a shot on the goal, whether the ball bounces off the goalkeeper is irrelevant, and trying to match velocity and acceleration wastes valuable interception time.

In the "Mousebuster" work, an attempt is made to catch a mouse with a robotic arm. Here, the end effector of the robot does not have to match the velocity of the mouse, so the problem is closer; instead, an attempt is made to catch the mouse by placing a cup over it on the floor. Therefore, the velocity of the robot at the point of contact must be 0 to avoid hitting the ground. As mentioned earlier, this full stop again wastes valuable time. Furthermore, the trajectory presented in the paper is a third-order jerk-limited trajectory. The TIGERs use only second-order acceleration-limited trajectories because the robots are built to withstand very high jerks in collisions with other robots, and the motors are powerful enough to generate the large acceleration jumps. More details on the TIGERs robots can be found in the team's latest publications [1, 5, 6].

3 Current Approach: Untimed Trajectories

As mentioned earlier, our current approach for time-optimal 2D BangBang trajectories is based on the approach presented by Cornell Big Red [7, 3]. It consists of two 1D trajectories for the two orthogonal axes x and y in the plane of the field. Each 1D trajectory consists of up to 3 phases with constant acceleration. An acceleration phase, an optional plateau phase with maximum velocity and an second acceleration phase. Each phase is described by the following equations of motion, with the position $s(t)$, the initial position of the phase s_0, the velocity

$v(t)$, the initial velocity of the phase v_0 and the constant acceleration a is either zero or the positive or negative limit $\pm a_{\max}$.

$$\begin{aligned} s(t) &= s_0 + v_0 t + \frac{1}{2} a t^2 \\ v(t) &= v_0 + at \\ a &= \text{const} \end{aligned} \tag{1}$$

A 2D trajectory is shown in fig. 1 that starts at $S_0 = (0\,\mathrm{m}, 0\,\mathrm{m})^T$ with an

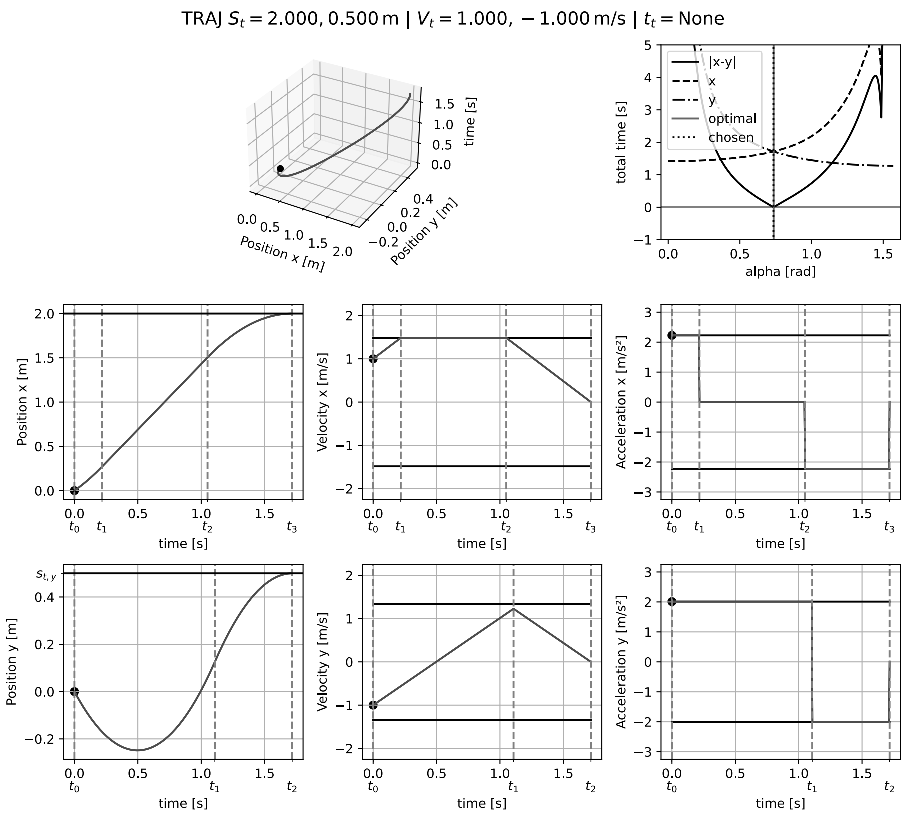


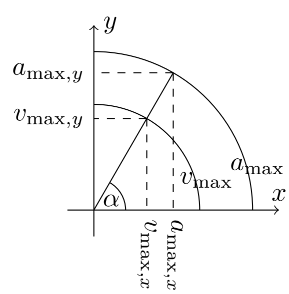
Fig. 1: 2D-Trajectory without Ball Interception Improvements

initial velocity of $V_0 = (0\,\mathrm{m\,s^{-1}}, 1\,\mathrm{m\,s^{-1}})^T$, ends at the given destination $S_t = (1.5\,\mathrm{m}, 0.5\,\mathrm{m})^T$, and has a maximum velocity of $v_{\max} = 2\,\mathrm{m\,s^{-1}}$ and a maximum acceleration of $a_{\max} = 3\,\mathrm{m\,s^{-2}}$. The bottom two rows of the figure show the position, velocity, and acceleration of the 1D trajectories of the x and y axes, with the target destination marked as a horizontal solid black line in the position graph. The 1D graphs have time stamps marked with a vertical dashed line: the transition times of the phases $t_0, t_1, t_2[, t_3]$. These are used in the following to assign variables to specific time points or time intervals. A position, velocity, or acceleration with a single index specifies its value at the exact time stamp, e.g., $s_i = s(t_i)$ is the position exactly at t_i. Variables with two indices describe the difference, e.g., $s_{i,j} = s(t_j) - s(t_i)$, or $t_{i,j} = t_j - t_i$.

The velocity profile of the x-axis in the middle clearly shows all three possible phases: $t_{0,1}$ the acceleration phase one, $t_{1,2}$ constant velocity phase and $t_{2,3}$ acceleration phase two. Overall, the velocity resembles a trapezoidal shape. The y axis has only the two acceleration phases $t_{0,1}$ and $t_{1,2}$, which resembles a triangular velocity shape. The constant velocity phase is omitted for the triangular shape because the distance on the y-axis is not long enough to accelerate to $v_{\max}$ and decelerate back to 0.

In order to enforce the maximum velocity and acceleration that the robot's hardware can achieve, we need to solve the optimization problem of what fraction of the total maximum velocity and acceleration can be assigned to each 1D trajectory. We can formulate the problem as finding the optimal angle α and computing from it the respective velocity and acceleration maxima using the following equations:

$$\begin{aligned} v_{\max,x} &= v_{\max} \cos\alpha \\ a_{\max,x} &= a_{\max} \cos\alpha \\ v_{\max,y} &= v_{\max} \sin\alpha \\ a_{\max,y} &= a_{\max} \sin\alpha \end{aligned} \tag{2}$$



The maxima are used to compute the total time t_{total} of the 1D trajectories. In the upper right corner of fig. 1 we see the total times of the x (dashed) and y (dashdotted) trajectories in relation to α, and the absolute difference between these times in as a black solid line. For a time-optimal 2D trajectory, the difference must be 0. This is indicated by the gray vertical line. The dotted black line marks the α^* chosen by the implemented optimization strategy. Since the absolute time difference is mostly convex, we use a binary search approach. For a more detailed explanation why a binary search is sufficient in a mostly convex scenario, refer to this year's extended team description paper (ETDP) [8].

4 Overshooting Trajectories

The extension now adds an extra input for the trajectories, not only the 2D destination S_t, but also a wanted arrival time, the target time t_t is given. It is marked in the 1D graphs as a vertical solid line, while the 2D graph marks the timed target position $(s_{t,x}, s_{t,y}, t_t)^T$ with a cross, as shown in fig. 2.

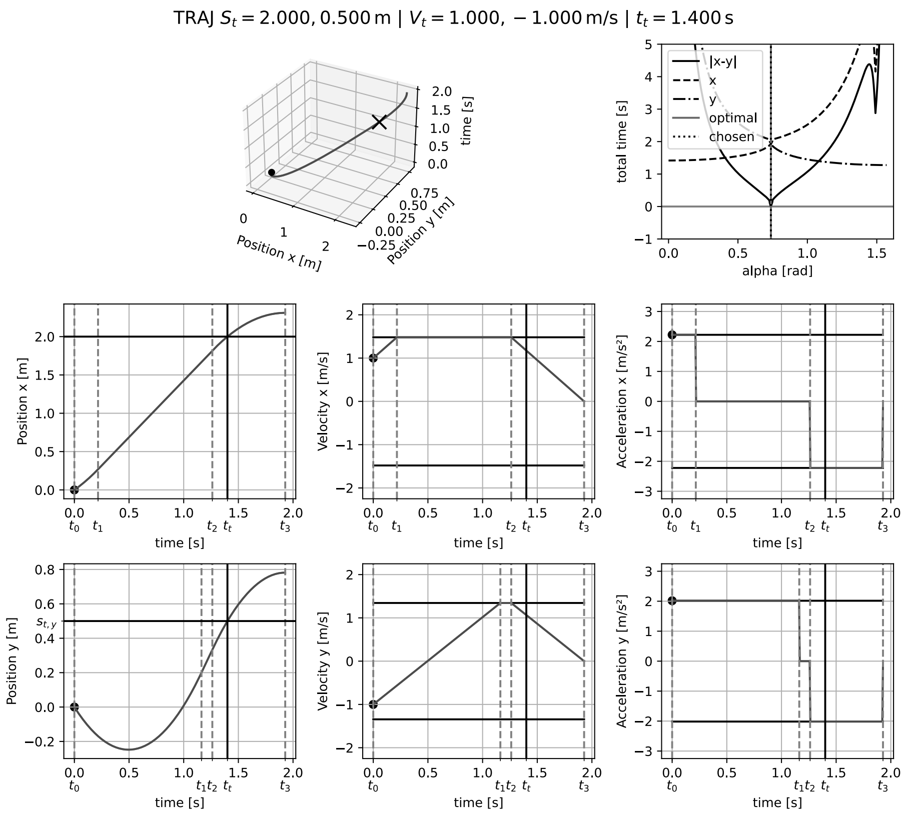


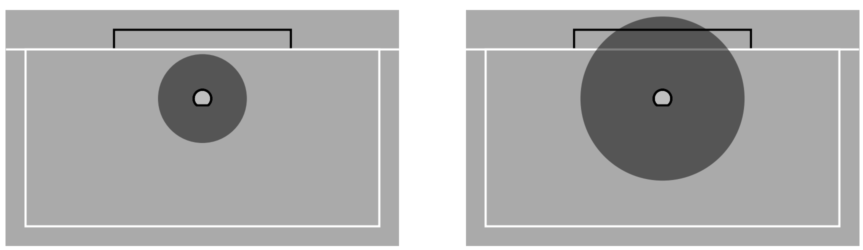
Fig. 2: 2D-Trajectory with Ball Interception Improvements

The new output of the extension are virtual destinations S_t^*, so s_3 of the x and y graphs in fig. 2. This virtual position $S_t^* = (s_{3,x}, s_{3,y})^T = (s_{t,x}^*, s_{t,y}^*)^T$ is then sent to the robot as a normal target destination, and the robot will drive there with a trajectory generated by the current approach, as the virtual position is created such that the robot passes the target destination at the target time. The virtual destinations can be placed behind the actual destination S_t to avoid the unnecessary breaking before reaching S_t, or exactly at S_t if t_t is large enough to allow for a full stop. We call trajectories where the robot drives further than the actually wanted destination overshooting trajectories, and they can be either a forced overshoot: The robot is too fast and passes the destination before the target time, so it overshoots and has to recover. Or they can overshoot deliberately, so that the robot heads for a virtual destination in order to reach the actual destination just in time.

For the generation of 2D virtual positions, the extension reuses the same α optimization strategy used in the existing approach, but replaces how 1D trajectories are generated. The new 1D trajectories to the virtual destination are generated, such that they get as close as possible to the target destination at the target time point. The total time t^*_{total} and final virtual position s^*_t of the trajectory are then used in the α optimization generation to combine two 1D trajectories to one 2D one. If the approach is successful and t_t is high enough to allow the robot to hit the target destination in time, the 1D position graph will intersect with both the target time t_t and the target position s_t at the same point, and the 2D position graph intersects the red cross.

A detailed description of how the proposed 1D algorithm works is presented in this year's ETDP [8].

5 Increase of the Effective Keeper Range



(a) No Overshoot $r_2 = 0.45\,\text{m}$ (b) With Overshoot $r_2 = 0.84\,\text{m}$

Fig. 3: Reachable Interception Points at $t = 0.75\,\text{s}$

To compare the effect of the overshoot, the situation of a goal shot with $6.5\,\text{m}\,\text{s}^{-1}$ and a distance of $4\,\text{m}$ is constructed. Resulting in travel times for the ball of roughly $0.75\,\text{s}$ depending on the carpet and ball model. The distances the keeper can travel within those $0.75\,\text{s}$ are drawn around a keeper in front of a Division A goal in fig. 3. Within the $0.75\,\text{s}$ the keeper travels without overshooting $0.452\,\text{m}$ and with $0.835\,\text{m}$, such that the keeper with overshooting can block all goal shots, that are further away than $4\,\text{m}$. The keeper without overshooting will need $1.055\,\text{s}$ to reach the same distance of $0.835\,\text{m}$ as the overshooting keeper, which translates to a kick distance of roughly $5.3\,\text{m}$. As mentioned above, these numbers are certainly not completely accurate, as they vary depending on the carpet and also goal shots cannot be detected immediately by the image processing system, but as a rough estimate, over-shooting significantly improves the performance of the goalkeeper.

6 Optimal Ball Interception Point Selection

The selection of the optimal ball interception point is not trivial, the simplest estimate with the foot of the perpendicular of the robot's position and the ball flight line can be improved. If the goalkeeper is standing still and the interception point is moved slightly towards the goal, the distance the goalkeeper has to travel increases only slightly, while the distance the ball travels increases significantly. Knowing how fast the ball and the robot will be at the interception point, it is possible to calculate how far the interception point should be optimally shifted towards the goal. However, the TIGERs keeper rarely stands still during a match and constantly updates its blocking position. Therefore, the calculation becomes more complex as the initial velocity of the robot strongly influences the optimal interception point. We decided to use a sampling approach because the entire trajectory generation process is very performant and can be executed many thousands of times per second. Every ten millimeters along the ball flight line and within the penalty area, a position is sampled. At each position the time it takes for the ball to reach the position is calculated and with this information a trajectory to a virtual destination is generated.

Trajectories are then selected via the following criteria: first distance to the intersection point, velocity at the intersection point, distance to the goal line, and finally time remaining. Thus, if the goalkeeper cannot reach the destination within the remaining time at any interception point, the trajectory with the smallest distance to the ball at its intersection time is selected. If there are multiple locations where the ball can be reached, the one where the goalie is slowest is selected, as this increases both accuracy and the margin for adjustments to the destination in the next frame. If there are multiple positions where the goalie can come to a stop in time, we prefer positions farther from the goal line. However, only within the first 0.27 m (3 bot radii), since we consider any interception point farther away to be safe. If there are multiple positions where the ball can be intercepted with a full stop that are further than 0.27 m from the goal line, the final decision criterion is to maximize the time between the goalkeeper's arrival and the ball to maximize the margin for future adjustments.

7 Results during the RoboCup

With section 5 highlighting the theoretical benefits of the presented extension, the practical proof is still missing that it is usable and works in a real tournament environment together with the interception point selection presented in section 6. The extension was added to the goalkeeper prior to the 2022 RoboCup, but in 2022 the goalkeeper had to block only two goal shots, of which none required any movement and successively no overshooting as the keeper was already positioned correctly before the shots were fired. Therefore, this section will focus only on the four goal shots during the 2023 RoboCup. Table 1 gives an overview about the shots and information to judge the interception. The first columns provide information to find the actual situation in the official game logs [1]. The keeper velocity mentioned is the absolute-measured velocity of the goalkeeper at the interception point, and the intercept constellation diagrams show the situation of the intercept. The flattened circle shows the outer shape of the keeper, and the point is the ball. The arrows pointing away from the shapes are their velocity directions at the interception point.

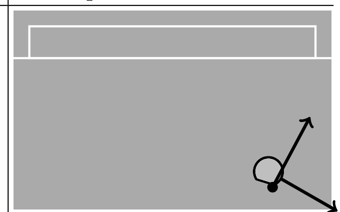
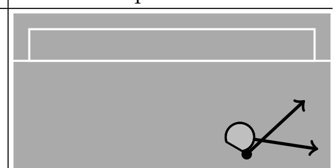
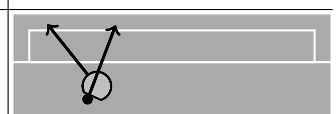
#	Opponent	Game	Timestamp	Keeper Velocity	Intercept Constellation
1	Immortals	Upper 4	1st 00:02	$1.8\,\mathrm{m\,s^{-1}}$	
2	Immortals	Upper 4	1st -00:36	-	Not Intercepted
3	ZJUNlict	Upper Final	2nd 00:19	$1.2\,\mathrm{m\,s^{-1}}$	
4	ZJUNlict	Grand Final	1st 03:16	$0.8\,\mathrm{m\,s^{-1}}$	

Table 1: Goal Shots at TIGERs Goal durign RoboCup 2023

To analyze the shots more deeply, the first one is shot number 2, which was a long shot over half of the field. It was deflected by our own robots close to our penalty area, which mislead the keeper and the keeper was too slow no matter the overshoot. For the other three shots, the keeper used the overshoot extension, as the keeper velocity was not zero or close to zero at the interception point. But especially for shot 1 and 3 the intercept constellation shows, that the ball was not intercepted optimally with the center of the keeper, and it were rather close calls. This is caused due to inaccuracies and latencies within the whole control setup, which complicates the task of ensuring that the robots follow exactly the calculated path. Between the Upper Final and Grand Final this system was tweaked, which can be seen in the interception constellation, as the keeper hits the ball more central.

[1] https://ssl.robocup.org/game-logs/

8 Conclusion

This paper describes an extension to the existing system for generating 2D trajectories. Unlike the existing approach, this extension does not guarantee that the robot will reach a destination point in an optimal time manner, including a full stop. This additional freedom allows the robot to reach the actual destination faster by removing the full stop constraint, while the extended algorithm ensures that the robot reaches the destination at a desired time, such as when intercepting a goal kick.

During the 2023 RoboCup, this extension was successfully used to defend 3 goal shots, and a deeper analysis showed, that the TIGERs system to control the robots is sufficiently accurate to follow the calculated trajectories and intercept goal shots.

References

1. Ryll, A., Ommer, N., Geiger, M.: RoboCup 2021 SSL Champion TIGERs Mannheim - A Decade of Open-Source Robot Evolution. In: Alami, R., Biswas, J., Cakmak, M., Obst, O. (eds.) RoboCup 2021: Robot World Cup XXIV. pp. 241–257. Springer International Publishing (2022)
2. Geiger, M., Ommer, N., Ryll, A.: RoboCup 2022 SSL Champion TIGERs Mannheim - Ball-Centric Dynamic Pass-and-Score Patterns. In: Eguchi, A., Lau, N., Paetzel-Prüsmann, M., Wanichanon, T. (eds.) RoboCup 2022: Robot World Cup XXV. pp. 276–286. Springer International Publishing (2023)
3. Purwin, O., D'Andrea, R.: Trajectory generation and control for four wheeled omnidirectional vehicles. Robotics and Autonomous Systems **54**(1), 13 – 22 (2006). https://doi.org/10.1016/j.robot.2005.10.002
4. Hove, B., Slotine, J.J.E.: Experiments in robotic catching. In: 1991 American Control Conference. pp. 380–386 (1991). https://doi.org/10.23919/ACC.1991.4791395
5. Ryll, A., Jut, S.: TIGERs Mannheim - Extended Team Description for RoboCup 2020 (2020)
6. Ommer, N., Ryll, A., Geiger, M.: TIGERs Mannheim - Extended Team Description for RoboCup 2022 (2022)
7. Kalmár-Nagy, T., D'Andrea, R., Ganguly, P.: Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle. Robotics and Autonomous Systems **46**(1), 47–64 (2004)
8. Ratzel, M., Geiger, M., Ryll, A.: TIGERs Mannheim - Extended Team Description for RoboCup 2023 (2023)