TIGERs Mannheim

(Team Interacting and Game Evolving Robots)

Extended Team Description for RoboCup 2026

Felix Weinmann, André Ryll, Nils Piecha and Michael Ratzel

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

Abstract. This paper presents the latest improvements in ball and robot modeling and calibration by TIGERs Mannheim, a Small Size League (SSL) team aiming to participate in RoboCup 2026 in South Korea. This year, the ETDP focuses on a ball motion model that explicitly represents phase dependent dynamics as time parameterized trajectories. We describe in detail how the ball model parameters for ground movement and chip behavior are calibrated and verified during the game. Furthermore, we present a vision based methodology for estimating translation velocity and acceleration limits of robots from historical and live data. Using these limits and the ball parameters maximizes our chances for successful passes and goal shots.

1 Importance of Calibrated Prediction Models in the SSL

To be able to achieve a wanted future outcome for current actions an accurate prediction of future states is necessary. For the RoboCup Small Size League (SSL) this requires an accurate model of the ball and robot movement capabilities. For a pass to succeed you need to know that the pass receiver can reach the target destination in time, ensure that the ball has a height and velocity acceptable for reception and minimize the chance of opponent robots to intercept the ball along the way [3]. Movement capabilities of our own robots are controlled variables and described in previous TDPs [13,7,8]. Uncontrolled variables include physical properties of the ball and field as well as the movement capabilities of opponent robots. These uncontrolled variables cannot be determined in advance and therefore have to be calibrated during tournaments. Furthermore, constant verification is beneficial in case robot capabilities or ball properties change over the course of a tournament.

Section 2 explains how we model the ball, calibrate this model and verify the ball model parameters during matches. Section 3 describes how we model robot movement capabilities and measure robot velocity and acceleration limits.

2 Ball Models and Calibration

In the RoboCup Small Size League, an orange golf ball (approx. 46 g, approx. 43 mm) is used as the game ball [11], so that golf-ball physics directly affects our ball model (flight, rolling phase, contact models).

This section outlines ball models for flat balls and chipped balls. Afterwards, a calibration procedure implemented in our central software is described to efficiently determine parameters which vary with the playing field. The final section shows how the calibrated parameters are supervised during matches.

2.1 Flat Ball Model

The model for kicked flat balls consists of two phases with different accelerations. It is only described here briefly as this topic is well covered in existing literature [12,2,5]. The two-dimensional velocity of the ball is given by (1).

$$\boldsymbol{v}(t) = \begin{cases} \boldsymbol{v}_{kick} + \frac{\boldsymbol{v}_{kick}}{\|\boldsymbol{v}_{kick}\|} \cdot a_{slide} \cdot t & 0 \leq t < t_{switch} \\ \boldsymbol{v}_{switch} + \frac{\boldsymbol{v}_{switch}}{\|\boldsymbol{v}_{switch}\|} \cdot a_{roll} \cdot (t - t_{switch}) & t_{switch} \leq t \leq t_{stop} \end{cases} \tag{1}$$

With:

$$\boldsymbol{v} = \begin{bmatrix} v_x & v_y \end{bmatrix}^\top \tag{2}$$

$$\boldsymbol{v}_{switch} = k_{switch} \cdot \boldsymbol{v}_{kick} \tag{3}$$

$$t_{switch} = \frac{\|\boldsymbol{v}_{kick} - \boldsymbol{v}_{switch}\|}{|a_{slide}|} \tag{4}$$

$$t_{stop} = t_{switch} + \frac{\|\boldsymbol{v}_{switch}\|}{|a_{roll}|} \tag{5}$$

A ball kicked with a velocity v_{kick} is initially sliding over the field. While it is sliding the ball has a non-zero velocity between its surface and the field. Sliding friction is modeled as Coulomb friction and hence independent of velocity. It is characterized by a constant acceleration a_{slide}. The ball gains angular momentum and looses forward momentum due to friction until its contact velocity reaches zero, at which point it is rolling with a_{roll}. The velocity at which this switch happens is denoted by v_{switch} and it is a fraction k_{switch} of the initial kick velocity as shown in (3).

It can be shown that k_{switch} only depends on the moment of inertia distribution β of the ball according to [12]. The moment of inertia of a sphere is defined in (6) and β ranges from $\frac{2}{5} = 0.4$ for a solid sphere to $\frac{2}{3} \cong 0.67$ for a hollow sphere.

$$I = \beta \cdot m \cdot r^2 \tag{6}$$

$$k_{switch} = \frac{1}{1+\beta} \tag{7}$$

It is interesting to note that k_{switch} must be in the range from 0.71 to 0.6 for the respective moment of inertia distributions. Yet, [12] empirically determined it to be 0.58 and [5] is at the lower end of 0.6. This would indicate that a golf ball is a hollow sphere with all mass concentrated at the outer shell which is definitely not the case. The reason for this discrepancy is a topic of future work.

Note that this model is only valid for kicked balls with no initial spin. Although a_{slide} and a_{roll} keep their validity, k_{switch} will vary and in case of spin in a direction not equal to the linear velocity will cause the ball to initially move along an arc path.

If calibrated the acceleration parameters are published by SSL-Vision in the `straight_two_phase` field of the `SSL_GeometryModels` message[1].

2.2 Chipped Ball Model

The model for chipped (or lobbed) balls uses three different coefficients of restitution during its bouncing phase and the same a_{roll} as a flat kicked ball during the final rolling phase. The coefficients of restitution are defined as follows:

e_{xy1} Ratio of planar (XY) velocity before and after the first impact event.
e_{xy} Ratio of planar (XY) velocity before and after all following impact events.
e_z Ratio of vertical (Z) velocity before and after every impact event.

The trajectory of a bouncing ball is discontinuous due to the discrete impact events and will thus be modeled as a piecewise function. The velocity during each hop is defined as:

$$\boldsymbol{v}(t) = \begin{pmatrix} v_{i,x} \\ v_{i,y} \\ v_{i,z} + g \cdot t \end{pmatrix} \quad \text{for} \quad 0 \leq t \leq t_f \tag{8}$$

With $\boldsymbol{v}_i$ being the initial ball velocity, gravity constant $g = -9.81$ m/s^2, and t_f the total hop time. The initial velocity of hop n+1 where $\boldsymbol{v}_{f,n}$ is the final velocity of hop n is:

$$\boldsymbol{v}_{i,n+1} = \begin{pmatrix} e_{xy} \cdot v_{f,n,x} \\ e_{xy} \cdot v_{f,n,y} \\ -e_z \cdot v_{f,n,z} \end{pmatrix} \tag{9}$$

Note that $e_{xy} = e_{xy1}$ for the first hop. Given these equations a chipped ball would never stop bouncing. Hence, we use the maximum ball height $p_{z,max}$ during a hop and switch to the rolling phase when this value is below a threshold $p_{z,roll}$.

$$p_{z,max} = \frac{v_{i,z}^2}{2g} \tag{10}$$

When the ball starts rolling the velocity equation is the same as for the rolling case in (1) and $v_z = 0$.

This model assumes that the ball is initially kicked with no spin. During the first impact event a part of the forward momentum is transferred to angular momentum. We observed that this effect is only significant during the first impact event. Hence e_{xy} is only split for the first and then all subsequent hops. In general $e_{xy1} < e_{xy}$ due to the momentum transfer. This does not apply to e_z as it is orthogonal to the impact plane and hence causes no change in angular momentum.

If the ball has initial spin this simplified model is not applicable anymore. Observed effects are a change in e_{xy}, depending on spin even for every hop. High initial backspin can actually cause the ball to have one or two forward hops followed by a backward hop and rolling. Furthermore, air drag and the Magnus effect are not part of the model. Air drag causes a velocity dependent decrease in velocity and the Magnus effect causes lift or downforce due to spin [1]. These effects would also cause the hop equation (8) to become non-linear. Initial data analysis of kicks performed with and without spin indicate no improved fitting with these effects taken into account. A more thorough analysis is part of a future work.

If calibrated the coefficients of restitution are published by SSL-Vision in the `chip_fixed_loss` field of the `SSL_GeometryModels` message [1].

2.3 Ball Model Calibration

Calibration of both ball models consist of several steps:

1. Automated sample collection on the field with two robots doing flat kicks.
2. Semi-automated sample selection with user interaction.
3. Determination of flat model parameters.
4. Automated sample collection on the field with two robots doing chip kicks.
5. Semi-automated sample selection with user interaction.
6. Determination of chip model parameters.

Note that the flat ball model must be calibrated first because a_{roll} is used during chip model parameter estimation as a fixed parameter.

Sample Collection Two robots passing the ball to each other are used on an empty field to collect samples. We developed a configurable play which can use variable locations and kick speeds. Although varying the kick speed is not required to determine ball model parameters it also helps us to calibrate the kick strength of our robots. Furthermore, it gives a dataset which closer reflects real game conditions where different kick speeds are used.

[1] https://github.com/RoboCup-SSL/ssl-vision/blob/master/src/shared/proto/messages_robocup_ssl_geometry.proto

To collect flat kick samples it is best to kick diagonally over the whole field to maximize ball travel time. To collect chip kick samples it is best to do it along the half field line where the cameras on a Div A field overlap as this provides stereo-vision capabilities to better estimate ball Z position.

The ball positions and according timestamps as reported by SSL-Vision are recorded for each kick from the initial kick point until the ball either stops, crosses a field boundary, or touches a robot. This two-dimensional trajectory is then put into a non-linear solver to find the global optimum of exact kick location, velocity, and model parameters. We have chosen to use a readily available CMA-ES solver from the Apache Commons Math Library. It is a derivative-free solver for global function minimization. The known kick location from the play and commanded kick velocity can be used as an initial estimate. For each estimate of parameters a full trajectory according to the models in section 2.1 or 2.2 is computed and projected to ground as seen from the respective camera. The solver then attempts to find the global minimum of mean squared errors between projected points and recorded points. The final result of ball parameters is stored in a list afterwards for further selection.

Sample Selection User interaction is usually required to run a sanity check on the collected samples. Often samples have to be removed when the kick distance was very short or SSL-Vision did not detect the ball for some period (due to improper calibration). An automated function can also remove all samples where any parameter is outside two times the standard deviation for this parameter over all samples. Calling this function repeatedly removes further samples and thus is only semi-automated.

Determination of Model Parameters The final set of parameters for a flat or chipped ball model is simply the average of all remaining samples after selection.

2.4 Ball Model Verification During Matches

Dedicated field time and expertise is necessary for the semi automated calibration procedure described in section 2.3. Online calibration during live matches has been investigated due to field time constraints prohibiting a dedicated ball model calibration session early at tournaments. For this each ball kicked during matches is sampled as described in section 2.3. The estimated ball model parameters from multiple kicks are gathered in a sorted binary tree as interference is expected due to the live game environment. To get a stable ball model estimation during matches the median over the gathered samples is used. To guarantee a sufficiently stable estimate the median selection of the samples is only started when reaching 31 samples. To limit the amount of data and runtime during matches the smallest and biggest value are removed from tree set when the amount of samples becomes larger than 31.

This approach yields less reliable parameter estimations than the dedicated calibration procedure due to the uncontrolled nature of the live match environment: The ball rarely rolls to a complete stop, limiting the amount of ball samples available for parameter estimation. Due to vision shadow some kicks cannot be observed completely. Spin generated during dribbling introduces an angular momentum to the ball when kicked, which results in a diverging k_{switch} measurement. As spin is not observable with vision data this harms the overall k_{switch} estimate. The amount of chip kicks necessary for a stable chip kick parameter estimation is rarely reached within a single SSL match, limiting the ability to calibrate these parameters online.

With our focus on reliability over peak performance the live calibration results are currently not directly utilized during games. Instead, the human operator is warned in case of a significant deviation of more than 20% being measured compared to the ball model parameters distributed by the vision. In the future this live calibration could be adapted for usage in the TIGERs Autoreferee, measuring over multiple consecutive matches to gather more samples and a more stable estimate suitable for distribution in the vision protocol.

3 Measuring Robot Movement Capabilities

The velocity limit and maximum acceleration are core parameters of our two-dimensional bang-bang trajectories [6,9]. These trajectories are used in a wide variety of tasks, like path planning [6,8], ball interception [10] and passing [3]. Trajectories are also generated for opponent robots for the evaluation of the pass interception risk. Therefore trajectory parameters corresponding to their performance need to be estimated as well.

Section 3.1 describes our movement model and its parameters. Section 3.2 shows our analysis into the observability of movement limits in vision data. Section 3.3 explains how we determine these movement limits during matches.

3.1 TIGERs Robot Movement Model

Achievable robot acceleration is under most circumstances the limiting factor for robot movement. Acceleration is limited by available motor torque and traction on the field carpet. The system of a direct current motor can be modeled as

$$\frac{L}{T} \cdot \dot{M} + \frac{R}{T} \cdot M = U - K \cdot \omega \tag{11}$$

with motor inductance L, motor resistance R, back-EMF constant K, motor torque M, motor torque constant T, speed ω and applied motor voltage U. According to eq. (11) increasing speed ω negatively affects the torque M and therefore acceleration, as soon as the voltage U can not be increased further to compensate $K \cdot \omega$, e.g. by reaching the battery voltage. Note that ω and M change drastically with a reduction gear ratio between the motor and wheels. The acceleration limit for asymmetric omnidirectional drives as typical in the

SSL is therefore primarily depending on the battery voltage, the drive train setup, robot velocity and the wheel locations relative to the direction of the desired translation [9]. The traction limit is dependent on the wheel locations as well, but also on the specific properties of the carpet used and as such variable between tournaments.

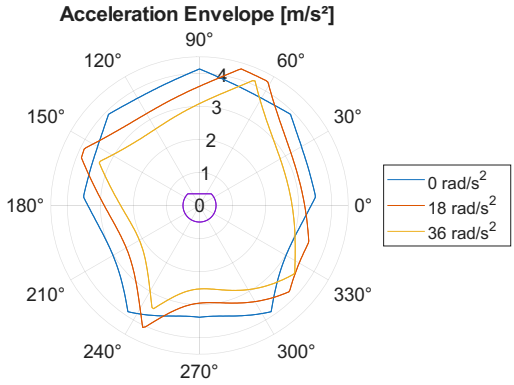


Fig. 1: Traction limit of the motion envelope for our robots with different rotation accelerations.

Figure 1 shows the directional traction limit of our robots for an arbitrary carpet. The maximum translation acceleration decreases in most directions with increasing rotational acceleration as the lines indicating different rotation accelerations show. We use a subset of the motion envelope modeling the maximum acceleration as uniform in all directions and for all velocities. This simplification enables us to utilize the computationally fast bang-bang trajectories and generate millions of them per second for all purposes while ensuring we can actually perform the wanted trajectories in the real world leading to a predictable behavior in time and space. To prevent the motor torque limit from interfering with the uniform acceleration model the velocity is limited to a certain maximum value. Rotation is considered as independent from the translation as maximum rotation velocity is seldom the limiting factor in the SSL.

The headroom for additional acceleration enables us to utilize a higher deceleration in the simplified scenario of breaking to a stop, resulting in a separate "breaking" acceleration limit.

This leads to the following parameters in our movement model:

- Translation acceleration limit in m/s^2
- Translation breaking limit in m/s^2
- Translation velocity limit in m/s
- Rotation acceleration limit in rad/s^2
- Rotation velocity limit in rad/s

3.2 Analyzing Historical Movement Data

For the development of the live movement limit determination analysis was done on historical data from the game log archive[2] based on vision data. As the latency and asynchronicity of robot movement increases the difficulty of mapping an intended trajectory to actual measured movement of the robot no additional internal data is used for measuring our own robots. Unfiltered vision data is utilized for the analysis as our vision filter resamples the vision frames in time domain and does not model acceleration due to the possibility of instantaneous changes.

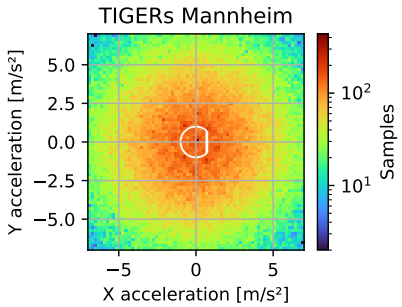
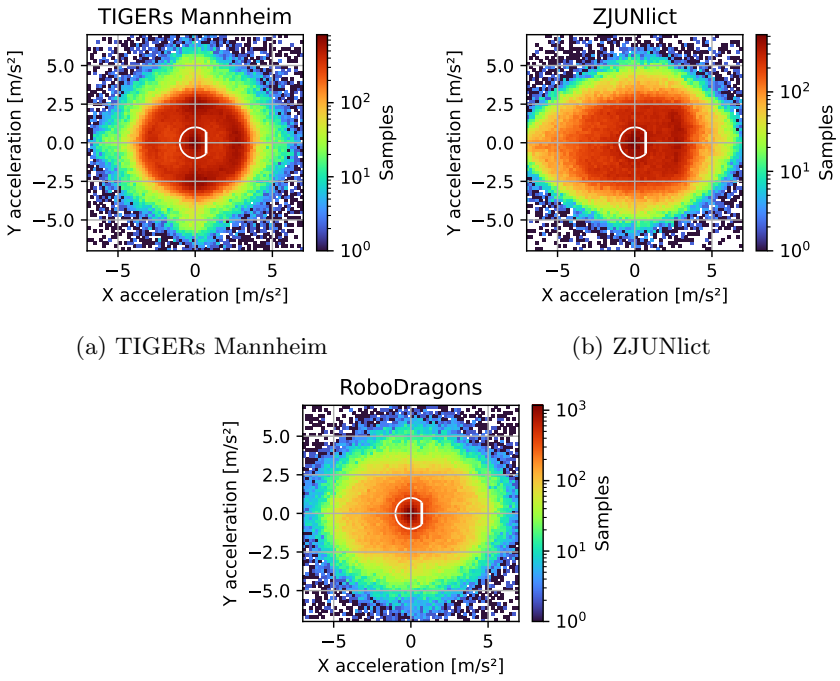


Fig. 2: 2D histogram of TIGERs Mannheim robot acceleration estimations using neighboring vision frames in HALT phases of the RoboCup 2025 grand final.

Figure 2 shows the distribution of a robot translation acceleration estimate based on three neighboring vision frames for TIGERs Mannheim during the HALT phases of the RoboCup grand final 2025. The acceleration directions have been normalized relative to the robot orientation indicated by the white robot outline. As no robot movement is permitted during HALT phases an acceleration estimation of $0\,\mathrm{m/s^2}$ would be expected. SSL-Vision has a standard deviation of 3.47 mm in robot positions reported at RoboCup [14]. When integrating over three neighboring frames at typical 70 frames per second for an acceleration estimation this standard deviation results in the large uncertainty visible in fig. 2. Therefore an integration over an longer time frame is necessary. As quicker movements get smoothed over when integrating over a longer time frame a careful balance has to be chosen for an acceleration estimation. For simplicity the integration time in this analysis was increased by using an offset of 5 frames for the neighbor search. Using a more accurate curve fitting approach is a potential future improvement.

[2] https://ssl.robocup.org/collected-data/



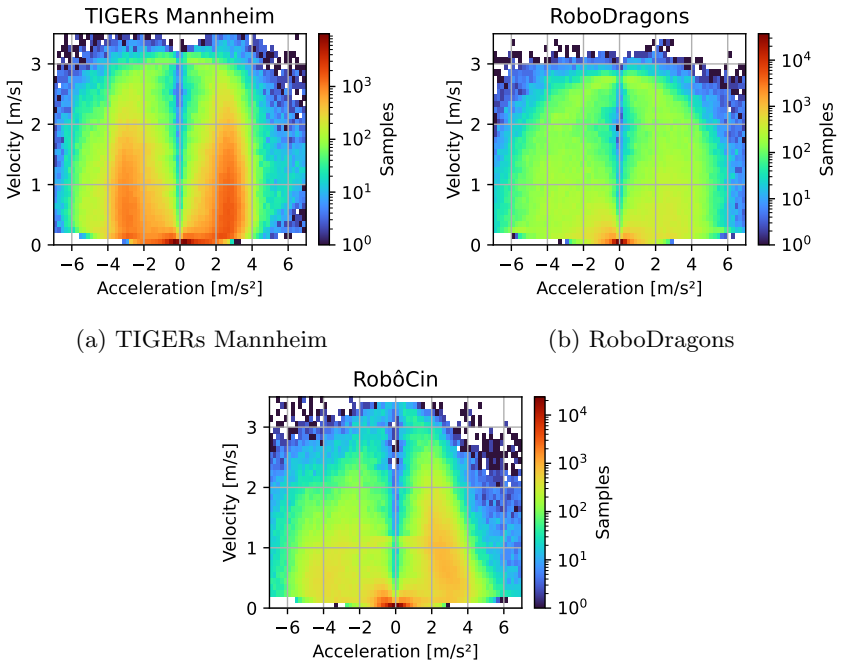
(a) TIGERs Mannheim

(b) ZJUNlict

(c) RoboDragons

Fig. 3: 2D histogram of robot accelerations relative to the robot orientation during RoboCup 2025.

Figure 3 shows a distribution of the directional translation robot acceleration and deceleration for a selected number of teams[3] during their respectively last game at RoboCup 2025 for all game states where the ball was in play. Robots slower than 0.1 m/s were filtered out for this analysis. The robot orientation has been outlined in white as visual reference. For us, TIGERs Mannheim, the $3\,\mathrm{m/s^2}$ uniform acceleration limit of our movement model is clearly visible in fig. 3a. The darker areas in the forward and sideway direction indicate a more frequent acceleration in these directions. This tendency is explainable by the typical orientation of our robots facing the ball. As such defenders and pass receivers predominantly move sideways, while the attacker typically accelerates towards the moving ball. According to fig. 3b ZJUNlict appears to utilize the asymmetric acceleration capabilities generated by their wheel geometry in their path planning which gives the best traction and acceleration backwards. RoboDragons does not show the clear acceleration boundary associated with least-time bang-bang trajectories in fig. 3c, likely linked to their usage of model predictive control [4].



(a) TIGERs Mannheim

(b) RoboDragons

(c) RobôCin

Fig. 4: 2D histogram of robot acceleration and speed during RoboCup 2025.

Figure 4 shows a histogram of the translation acceleration in relation to the current robot velocity for a selected number of teams[3] during their respectively last game at RoboCup 2025 for all game states where the ball was in play. An acceleration was determined as negative if the velocity vector magnitude has decreased. For TIGERs Mannheim the rectangular subset of the motion envelope is visible in fig. 4a, showing the acceleration limit of $3\,\mathrm{m/s^2}$, velocity limit of $3\,\mathrm{m/s}$ and occasional breaking with $6\,\mathrm{m/s^2}$. The lack of an identifiable acceleration targeted by RoboDragons in fig. 4b again shows the impact of their model predictive control while showing the capability to comparatively high accelerations. RobôCIns robots show a decreasing acceleration with increasing velocity in fig. 4c, possibly showcasing a torque limitation. The horizontal line at $1.2\,\mathrm{m/s}$ indicates some RobôCIn robots being limited by a maximum velocity restriction similar to the one imposed during stop phases even though the ball is in play.

A similar analysis of rotation velocity and acceleration yielded no usable information, likely due to the absence of large rotations in typical gameplay.

[3] The analysis script used and plots for all other teams at RoboCup 2025 are available at https://github.com/TIGERs-Mannheim/MovementAnalysis

Due to the rotation speed of the robot rarely being a limiting factor no further analysis on rotation velocity and acceleration was done.

3.3 Movement Limit Determination

The velocity and acceleration estimates generated from raw vision data can be utilized to determine the movement limits for our movement model for both teams during matches. Although section 3.2 shows that not all teams behavior can be represented by our movement model, it can give an approximation sufficient for e.g. estimating an interception probability during passes.

To get a stable estimate despite the observation uncertainty a percentile based approach is used to determine the parameters. Three histograms are used to store collected velocity, acceleration and deceleration samples to keep the amount of collected data and runtime constant. To account for potential modifications collected data is reset during breaks or timeouts. To guarantee the stability of the estimation parameter determination is started when at least 2000 samples have been gathered. The 0.995th percentile of measured velocity samples is used as velocity limit, the 0.75th percentile of measured acceleration samples is used as acceleration limit and the 0.95th percentile of measured deceleration samples is used as breaking limit. These percentiles have been chosen empirically using historical game log data to match our values configured at that time.

This approach is currently not usable for robot control as every systematic error in the estimation would result in a feedback loop. As our movement model does only allow for a subset of the possible motion envelope an misestimation of the robot capabilities is likely to occur. Therefore the online determined movement limits are not used automatically for our own robots.

4 Publication

Our team publishes all their resources, including software, electronics/schematics and mechanical drawings, after each RoboCup. They can be found on our website[4]. The website also contains several publications with reference to the RoboCup, though some are only available in German.

References

1. K. Aoki, M. Muto, and H. Okanaga. Aerodynamic characteristics and flow pattern of a golf ball with rotation. *Procedia Engineering*, 2(2):2431–2436, 2010.
2. R. Cross. Transition from sliding to rolling in billiards and golf. *Physics Education*, 56(4):045003, 2021.
3. M. Geiger, N. Ommer, A. Ryll, and M. Ratzel. TIGERs Mannheim - Extended Team Description for RoboCup 2025, 2025.

[4] Open source and hardware: https://tigers-mannheim.de/publications

4. M. Ito, H. Kusakabe, Y. Adachi, R. Suzuki, J. Du, Y. Ando, Y. Izawa, S. Isokawa, T. Kato, and T. Naruse. RoboDragons 2018 Extended Team Description, 2018.
5. C. Lobmeiner, P. Blank, J. Buehlmeyer, D. Burk, M. Eischer, A. Hauck, M. Hoffmann, S. Kronberger, M. Lieret, and M. Eskofier. ER-Force - Extended Team Description for RoboCup 2016, 2016.
6. N. Ommer, A. Ryll, and M. Geiger. TIGERs Mannheim - Extended Team Description for RoboCup 2019, 2019.
7. N. Ommer, A. Ryll, and M. Geiger. TIGERs Mannheim - Extended Team Description for RoboCup 2022, 2022.
8. N. Ommer, A. Ryll, M. Ratzel, and M. Geiger. TIGERs Mannheim - Extended Team Description for RoboCup 2024, 2024.
9. O. Purwin and R. D'Andrea. Trajectory generation and control for four wheeled omnidirectional vehicles. *Robotics and Autonomous Systems*, 54(1):13 – 22, 2006.
10. M. Ratzel, M. Geiger, and A. Ryll. TIGERs Mannheim - Extended Team Description for RoboCup 2023, 2023.
11. RoboCup SSL Technical Committee. Rules of the robocup small size league. PDF, 2024.
12. R. Rojas and M. Simon. Like a rolling ball. PDF, 2006.
13. A. Ryll and S. Jut. TIGERs Mannheim - Extended Team Description for RoboCup 2020, 2020.
14. F. Weinmann. Vision based Understanding of the RoboCup Small Size League field, 2024. https://download.tigers-mannheim.de/papers/2024-Vision-based-Understanding-of-the-RoboCup-Small-Size-League-field-Weinmann.pdf.