Eindhoven University of Technology

Department of Mathematics and Computer Science

EngD Mechatronics System Design 2022 – 2024

System Architecture document

Autonomous Referee

Provided by Ahmet Demirel

Introduction

What the system should do:

To understand what systems should do, the needs of the stakeholders should be referred. The AutoRef project had been assigned to MSD EngD trainees for 7 years, therefore some of the stakeholders' needs were known beforehand and can be addressed directly. Then, the updated version of the needs (if there is any update) can be discussed further to understand the context and the requirements of the system.

According to the previous years' works, stakeholders demanded 5 minutes of match between teams of 2 robots, which is officiated by AutoRef. Aside from that final goal, it is stated that the system should be able to log and show evidence. It must be communicated with a human referee to help him/her and to get feedback about the decisions it made.

Furthermore, the project should be built on the earlier work and should have a structure to make it possible for future generations to work on. In order to achieve this, clear documentation including the earlier work process should be included.

To make a list of general idea about the expectations:

1- The AutoRef should be accurate about the decisions it made
2- It should be reliable, can work in different stadiums and independent of area of gameplay and time of the game.
3- A final delivery suitable to the team's resources should be decided.
4- Personal goals of the team members should be achieved during the process
5- A validation method should be presented and used.
6- The system should have logs of the match and can show evidence of violations when needed.
7- Proper documentation should be provided.
8- The project should build on the work of the previous years.
9- The system can communicate with the human referee, have feedback, and help during the match.
10- The project should allow the next generation to work on top of this year's results.

Understanding the context and requirements is an inter-dependent and iterative process. Therefore, having a broad knowledge about robot soccer and Tech United team is essential.

RoboCup MSL and Tech United Overview

In order to promote research and education, RoboCup, Robot Soccer tournaments have been held since 1997. There are various categories of in such robots can compete in a simulation environment or their shape can be restricted only to humanoid form. Tech United, Robot Soccer team in TU/e, competes in

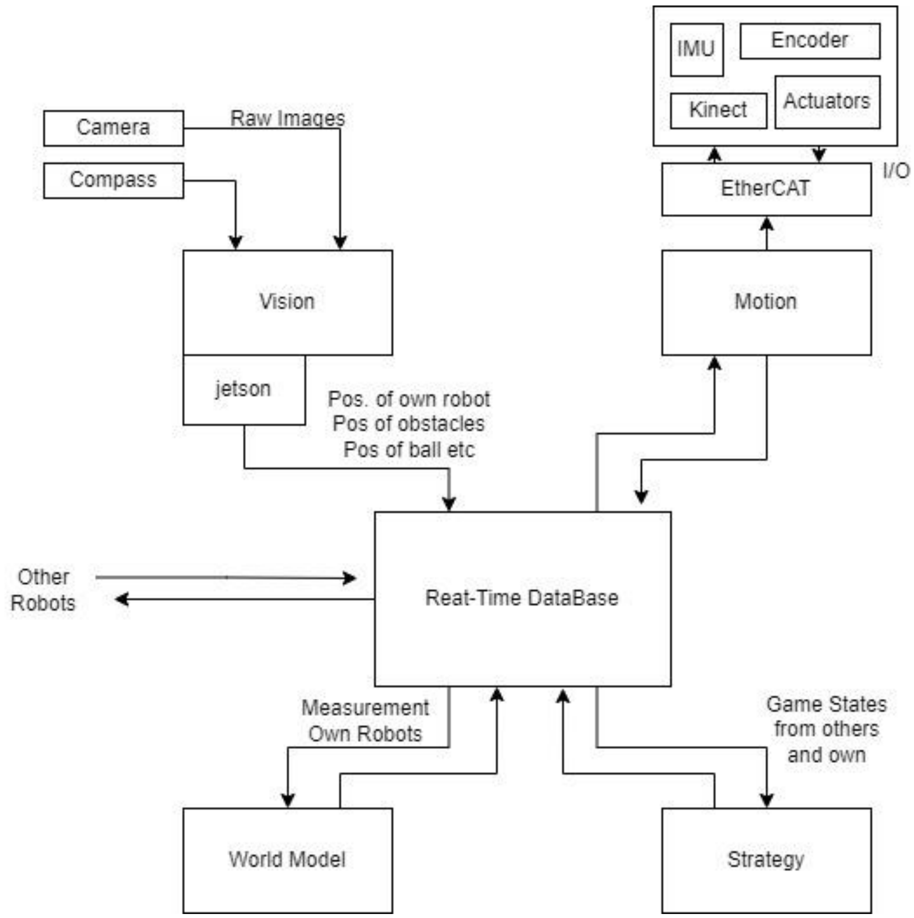
Middle Sized League (will be abbreviated as MSL) and in @home league with service robots. The team is composed of alumni, PhD students and university employees. The AutoRef project is related to the MSL competition.

Whichever technical solution would be used for the AutoRef, it should resemble soccer playing robots. The data sources may differ, however, the AutoRef should be able to tell some or all the game states to be able to automatically judge the game. It is wise to have a general idea about how the Tech united robots sense the environment.

Turtle (The Tech United RoboCup Team: Limited Edition)

The soccer robots of Tech united team are called Turtle. Turtles use two different cameras. One of them is Omni Vision camera targets directly a convex mirror to see a field of vision around the Turtle, approximately 6 meters. The other camera is a Kinect that helps Turtle to gather depth information and recognize the ball when it flew above 80 centimeters. The physical details of the Turtles are not related directly but the detailed information can be found via this link.

The software can be divided into 4 main categories, namely, vision, World Model, Strategy and Motion. At the Vision module, the images are retrieved and analyzed. They are also saved to disk. In a subsection named jetson, the images are processed. The field lines, ball and obstacles are understood. However, if the obstacles are fees or foes, it is not distinguished here. In the World Model part, information about the obstacles is combined and a single view of them is created. Opponents and team members can be differentiated here. Also, their position and velocity can be estimated. However, this part is not responsible for sensor fusion of the robot’s state and is not in charge of detecting the ball.



Inside the strategy module, by using all measurements of the current state of the game, a strategy is chosen from the playbook. Furthermore, the corresponding roles of the Turtles are distributed. The motion module oversees operating robots regarding to the commands coming from the strategy module. There are also helper tools, such as remote control or simulation and functions for algebraic calculations. More detailed explanation can be reached via GitHub link.

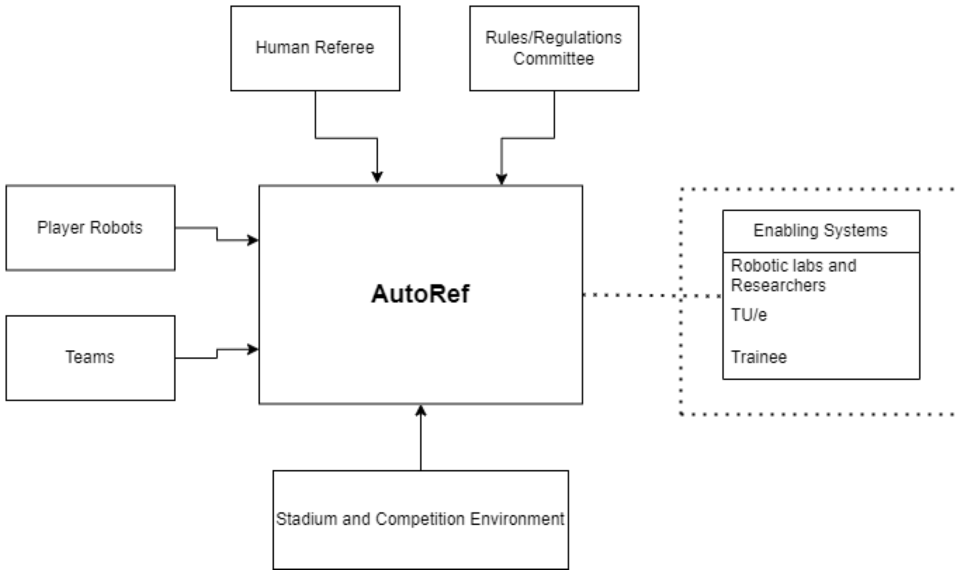
Physical Environment and Organizational Rules

RoboCup tournament uses similar rules and concepts to FIFA laws. The stadiums are smaller and indoor type. Field markings are white while the ball is yellow to make it easy detect. Teams are composed of five people and one of them should be the goalkeeper. Aside from the laws, there are competition rules to define qualifications, point system etc. for different tournaments. To get more information MSL rulebook can be referred.

System Thinking Processes

External and Enabling Systems

External Systems are defined as the related parties with the AutoRef while it is working such as players and teams during a match. Enabling Systems are defined as the people or organizations that are indirectly related but have mutual interests with the system. That can be analyzed below figure.



From the figure, it can be understood that the competition environment and rules of the competition forms of the system constraints. Teams, robots, human referee, and committee members are related to the functional requirements and corresponding performance/validation criteria of those.

Mission Timeline

To define scope and requirements well, the system's mission should be presented. Below diagram can be examined to get better understanding.

Pre-mission	Mission	Post-Mission
System mounting to environment	Check the kick-off procedure	Ensure and handle the data related to the game is safe
Communication with related parties before match (Refbox etc.)	Set time and start the match	Shut Down
Come Online	Monitor the whole match and game states	
	Record the last position of ball in case of connection with and field line	

	Detect rule violation	
	Decision-making algorithm (award and punishment system such as penalty or yellow card)	
	Help human referee if needed	
	Log and show evidence for game calls	
	Stop the game when needed and check procedures before starting again.	
	End match	

Needs, Requirements and Deliverables

Constraints can be deduced from the above table and external systems diagram.

1- System needs to be mounted on any stadium/competition ground.
2- System needs to support a connection with Refbox and a competition PC.
3- System needs to be adaptable to rule/procedure changes
4- System needs to be able to allow data retrieval after the match

Functional and Behavioral expectations can be listed as follows.

1- System needs to be able to check game start/stop procedures and able to send signal (via Refbox.)
2- System needs to set and get time of the match and eventually start/stop the match.
3- System needs to monitor the match, acquire related data, and deduce relevant information.
4- System needs to judge positions during the play and decide award/punishment situations.
5- System needs to be able to log the game. The data can be used as evidence, so that the format should be proper or convertible.

By utilizing constraints and expectations from the system, the following functional requirements can be deduced.

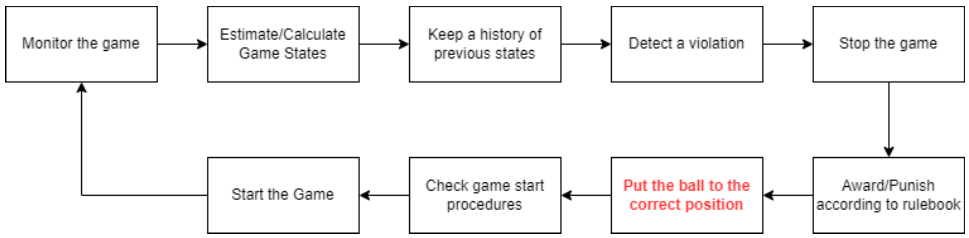
1- AutoRef shall be aware of the time and use this information to start and stop the match.
2- AutoRef shall comply with MSL Rulebook while deciding violations and following procedures

3- AutoRef shall decide an award or punishment regarding of the violations and procedures during the match
4- AutoRef shall gather data and can comprehend its reliability
5- AutoRef can estimate game states by using data gathered during the match
6- The game states can be listed as follows:
    a. The positions and velocities of soccer robots for each 0.2secs
    b. Ball engagement of each robot
    c. Robot and team id
    d. Robot's pose – angular orientation
    e. Positions of the lines and specific areas in the field.
    f. The field line positions the ball has touched last before leaving the play area.
    g. The time elapsed between the first and last touch of the robot during dribbling.
    h. The position and velocity of the ball.
    i. Time

Performance Requirements:

1- All hardware regarding the AutoRef shall be compatible with MSL organization environment.
2- AutoRef shall give permission to users to modify pre-defined game rule parameters and/or deploy some/all the rules.
3- AutoRef shall be connected to Refbox with maximum 500ms latency.
4- AutoRef shall be capable of sending/receiving digital ON/OFF signals to/from AutoRef.
5- AutoRef shall record/store game data as well as log history.
6- AutoRef shall allow users to use stored data and download it.
7- AutoRef shall upload game data automatically to a cloud deposit and ensure its safety during the process.
8- AutoRef shall acquire data with 20hertz and estimate game states max 10 cm for positions
9- AutoRef shall decide a violation within 1 second.

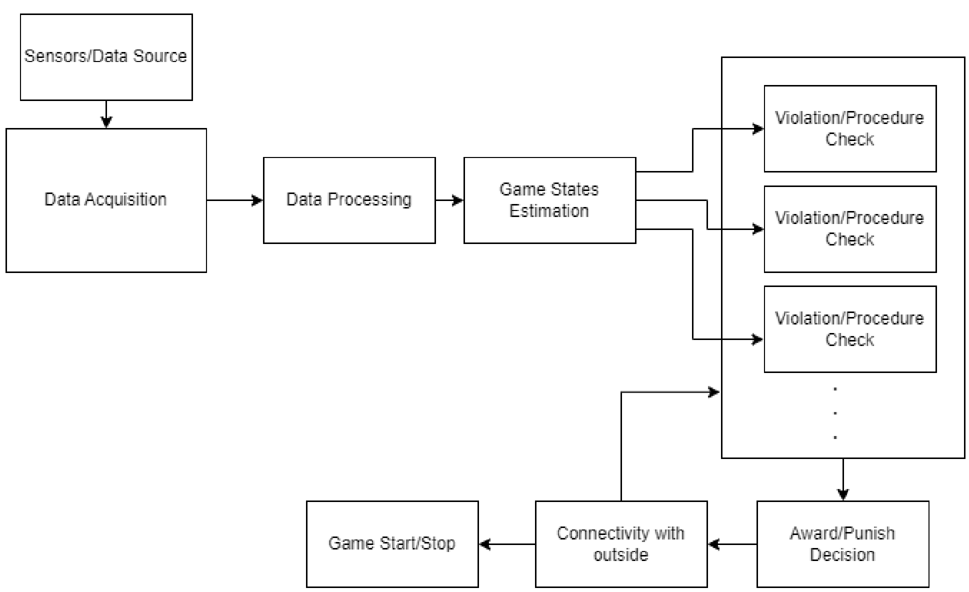
To understand the needs and constraints of the system, aside from meeting with the stakeholders, one needs to do research into the problem. By watching the real matches of Tech United during the tournaments (that can be found in YouTube), a flow diagram of the of the mission process can be presented such as below:



One can clearly see that the solution of the autonomous referee problem cannot be addressed only as software that relies on violation check by acquiring data. Therefore, a mechatronic solution should be added as a part of the system. However, the system can be decomposed into two groups and the ball handling system can be assumed as an external independent system. To make this project more achievable, the ball handling system will not be covered by this semester. However, it is advised that the next generation can address this part of the problem in more detail.

Into the Solution Domain

Before moving into the solution domain, a logical decomposition of the system can be shown as below and by using the diagram, the architecture focus can be decided.



One can immediately see that the algorithm for rule violation can contain one or multiple different sub-algorithms depending on the solution. Since there are not only rules but also procedures to be checked, there is a feedback loop between the outside connection and decision-making modules. That can be explained by checking flow diagram above, for each violation happens, the game needs to stop and before starting it a procedure needs to be checked. The design may have parallel processing or sequence-like patterns. Those are both architectural concerns that need to be addressed while searching for a solution.

Furthermore, data acquisition sub-system is one of the critical points of the possible solution, because it will affect the software architecture and data processing methods directly. Also, the data acquisition method will highly be related to the accuracy and reliability of the system whose importance was discussed in the stakeholder's meetings. As a result of these ideas, the logical strategy is to focus on the data acquisition method from an architectural point of view.

Secondly, the rule violation algorithm should be defined such that it allows adaptability of the system to possible rule changes in future. A parametric approach should be used for simple changes. Also, the decision-making module should work in parallel with different rule-checker code-snippets, which make it easy to implement new rules when needed.

From the high-level abstraction, it was stated that the data acquisition system should be the focus. In order to dive solution domain, a different technology should be researched. There are some technologies that can be used such as;

1- Vision systems
2- Sound
3- Laser
4- Temperature
5- Inertial sensing

For their advantages and disadvantages one can refer to feasibility reports.

All the above technologies are used in different competitions to support human judgement or replace it automatedly. The more detailed solution needed, the more combination of different sensing systems is merged. The most common assistive systems are VAR in football and hawk eye in tennis. Also, many sensing technologies are used to collect data and analyze statistics to improve performance. Inertial sensing is particularly used to understand muscle/bone movement and interactions during sporting events not only to develop a strategy but also to understand health concerns during competitions.

More importantly, the technological readiness level of those solutions is already minimum TRL7 or TRL8. Technology readiness levels (TRLs) estimate the maturity of technologies during a project's acquisition phase. Therefore, the hardware implementation or mounting of the sensing systems become more important. The complexity of the technical work as well as the costs involved are other major concerns. It can be observed that the RoboCup tournament areas are not standardized yet and different stadiums can be used. The infrastructure may differ from tournament to tournament. Also, the robot designs of the teams can differ for each tournament. This leads to us, a technology which needs less calibration, easy to use and mount, adaptable to environmental changes and inexpensive should be chosen as a main solution direction. Regarding the project needs, the MSD trainees also need to choose deliverables that are compatible with their personal achievements and to set reachable goals. Aside from the general solution architecture, a starting point should be selected carefully, that will enable the current generation team as well as help future generation to build on.

Therefore, data acquisition strategy is selected as using multiple cameras around the stadium. For the starting point to set an integration delivery, a corner kick game start procedure will be used. The reason for choosing this rule can be summarized as follows;

1- The procedure is simple
2- Similar with other game starting procedures.
3- Camera technology is less complex, and hardware is reachable.
4- Implementation has a direct effect on the game, since the human referee does not measure distances during a real match.
5- The procedure is not directly related to any rule violations that makes it easy to integrate with future/past implementations.

Implementation Strategy

Since the data sources have been selected, the proper techniques should be identified as a part of the solution. There are two options, the first one is to use different layers of filters to hand craft object features and use them to detect lines and robots. The filters should be tuned one by one, which is time and energy consuming. The second option is to use a machine learning algorithm to detect robots and merge it with a homographic transformation to detect distances between robots. This approach can be found in the self-driving cars and robots' literature, however, we also checked research about surveillance cameras, due to COVID brings our lives social distance concept, which overlaps with RoboCup's game starting procedures. The following papers have been considered as a starting point. Below questions should be answered properly and viable options should be searched thoroughly, before starting implementation.

1. Decide which camera and which angle to be used
2. How to Calculate/Tune camera depth and pixel relation parameters

3. Find a proper ML technique to object detection
4. Prepare a dataset for training/testing purposes
5. What kind of images needed to be used in dataset

There are three different options about the camera decision. The first one is using security cameras. There are four different cameras around the stadium that all have different angles and perspectives. One should look through them one by one to decide the best option. The other choice is to use object detection cameras around the stadium. Due to special permission that needs to be given, we have not considered this option, due to the limitations of using them during project timeline. There are also two GoPro cameras that can be borrowed from Tech United, however, they need to be mounted around the stadium. A fixture needs to be found or designed, to be able to mount them properly. Also, the angle of the GoPro model was narrow, which makes it difficult to work in a PoC stage. For those above reasons, we decided go work with one of the security cameras, having a clear view of one of the corners.

The solution for the defined scope can be summarized as follow:

By using one of the security cameras, the corner kick procedure will be officiated. In detail, an object detection algorithm will be run in a proper video recording to detect the ball and players from different teams. Then a bird's eye transformation will be applied to calculate distances between them. By using the distances, the corner kick procedure will be checked and a signal depending on the players positions will be sent to the referee. A dataset containing robot images from different teams and ball images with and without a player around it should be collected in the first place. It has been decided to use YOLO to detect objects. YOLO is a well-known deep learning architecture. It is not only easy to find several resources to implement but also to train.

The explanation of the deep learning models for object detection is out of scope of this report. However, this can be addressed shortly. YOLO models are trained by different datasets for years and more accurate versions are developed. By using predefined software templates and also pretrained models, a high accuracy model can be achieved. The libraries are also optimized to automate using best parameters and strategies, they are fool-proof. The dataset is also expected to be small, compared to a standard dataset that needs to be used in a computer vision model. Therefore using an already implemented YOLO model would be beneficial for performance.

The bird's eye vision transform is another to calculate distances between objects. A simple Euclidian distance calculation can be used, however a proper coordinate axis should be set before starting calculations. The simplest approach would be to get a top view of the image and a proper scaling should be done. By that way, one can convert pixel distances in to real-world measurements. In order to transform the auxiliary view in to top view, a linear transformation have to be done. This transformation matrix can be calculated by matching different points taken from image in terms of pixels and real-world measurements. Luckily, the field lines are perpendicular to each other, and they can be used to calculate transformation matrix.

After collecting a video recording via one of the security cameras around the stadium, 300 frames are stored to be used as data set. The frames are chosen such that the different poses and positions tried to be kept. Especially the ball should be captured both near a player and isolated, since the model needs to differentiate the ball near a player or rolling/standing itself.

Since the robots can move sidewards also, their pose will change during the match even if we stable the camera angle. Therefore, the dataset needs to be enriched. While creating the dataset, augmentation of the photos was done. The photos were tilted up to 5 degrees to both sides and 1%pixel noise was added. It is expected that the dataset would be more generalized against poor image quality and various positions of the robots.

As for the training of the model, it is advised to train the ML model in the local environment and call for predictions. However, due to time restrictions our model is trained at RoboFlow service online. RoboFlow is a community that helps and enhances the use of Yolo models with their products. By using their website, it is possible to create dataset, augment the photos and train the model without dealing with complications. The trained model, which is Yolov5, is then used via api-key of the service. It can be directly call from Jupyter notebook and predictions can be gathered as a dictionary. Although it is easy to train and use the model via this service, the output structure it provides is difficult to work with. In other ML libraries, it can be observed that a tensor form is used as an output data type. Therefore, proper data structure preparation needed to be done, before starting to implement the violation/procedure check algorithm.

After prediction data was gathered, we implemented the corner kick procedure in the algorithm. The details of the implementation can be followed in the code via project GitHub. In order to access security cameras in real-time, another permission needed to be taken. Therefore, an offline method was implemented. An output was queried from the RoboFlow servers, which took considerable time. However, it was a reasonable choice with no real-time implementation option.