3D Ball Positioning

Wouter Kuijpers

Eindhoven University of Technology
Universidade de Aveiro

w.j.p.kuijpers@student.tue.nl

Supervision:
dr. ir. René van de Molengraft (TU/e)
Prof. António Neves (UA)

November 23, 2015

Content

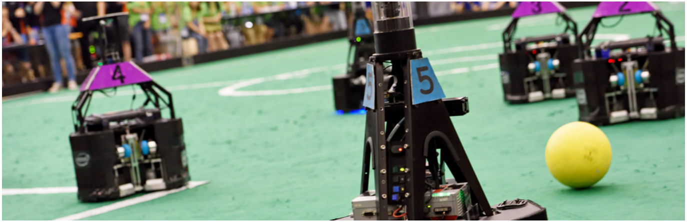
Part 1: 3D Ball Positioning

Part 2: Challenges

Part 3: Triangulation Algorithm

Part 4: Results

Part 5: Conclusions & Recommendations



3D Ball Positioning

Target: *develop 3D position detection for the ball, which can be applied in the Middle Size League (MSL)*

3D Ball Positioning

Target: *develop 3D position detection for the ball, which can be applied in the Middle Size League (MSL)*

- Several solutions/methods to this have been presented, for example adding a
  - perspective camera,
  - kinect camera,

  to the robot, or by using the bijection principle applied to projection model.

3D Ball Positioning

Target: *develop 3D position detection for the ball, which can be applied in the Middle Size League (MSL)*

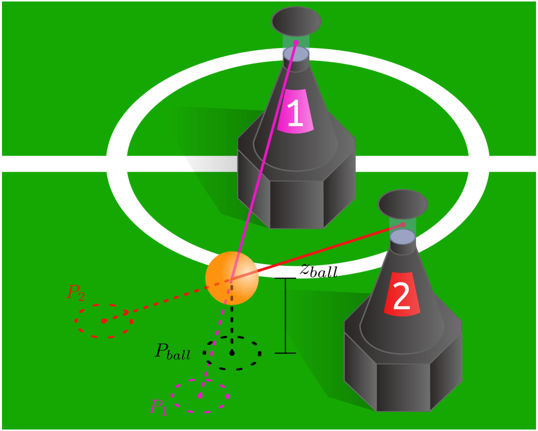
- Several solutions/methods to this have been presented, for example adding a
    - perspective camera,
    - kinect camera,

  to the robot, or by using the bijection principle applied to projection model.

- An interesting solution/method might also be found with cooperative sensing:

  Research and Verify whether cooperative sensing can be used for real-time 3D ball positioning using triangulation of omni-vision camera data.

Triangulation of Omni-Vision Camera Data

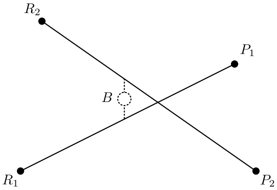


Applications:

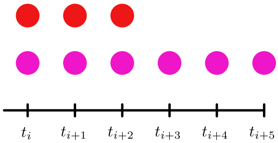
- reliable ball position projection on field: P_{ball}
- height of the ball z_{ball}, enables interception strategy (lob pass)

Challenges

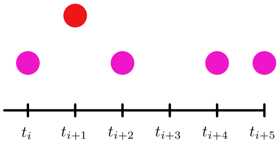
1. **Challenge 1:** Lines of Sight (LoS) might not cross due to noise in measurement.
   - **Possible Solution:** do not calculate intersection but minimum distance between LoS's. Check!
2. **Challenge 2:** communication delay in robot-robot communication.
   - **Possible Solution:** apply extrapolation to omni-vision data from other robots. Communication Delay + Noise → large inaccuracies.
3. **Challenge 3:** limited amount of data from other robots.



(a) Challenge 1

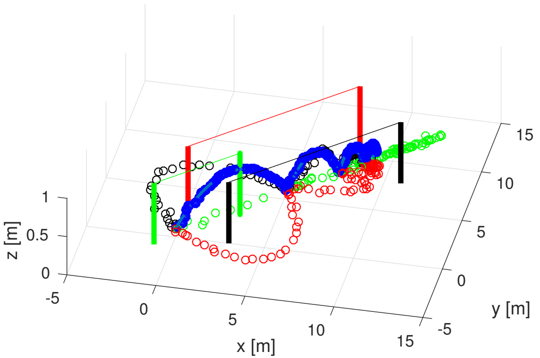


(b) Challenge 2



(c) Challenge 3

MATLAB Simulation Environment



Additions:

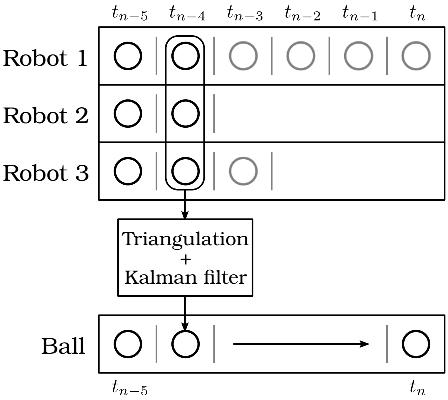
- Multiple (> 2) Robots
- variable Communication Delay, Clock Offset and Package Loss
- Hybrid Automaton ball model

Triangulation Algorithm

- Current Time: t_n

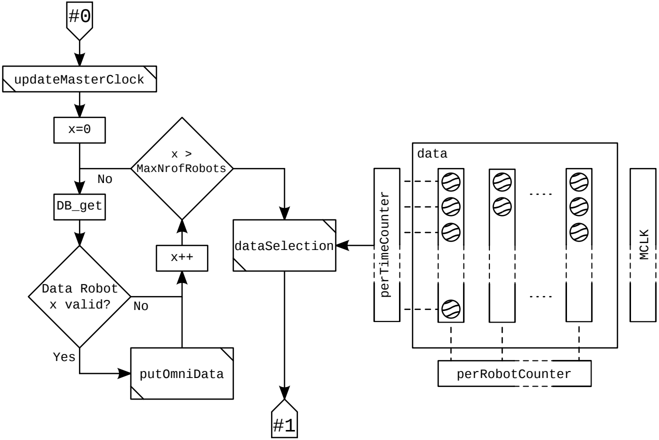
- *Solution 2:* triangulation will be performed on past data, which is stored in a data buffer.
- *Solution 3:* combine LoS with predicted position of Kalman filter.

- Algorithm consists of 4 main modules.



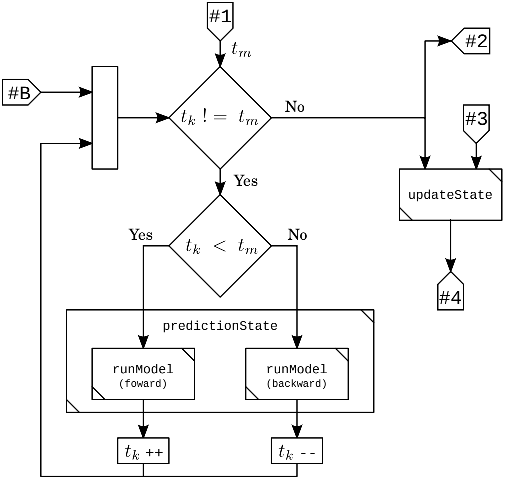
Data Buffer

Data buffer: retrieving, storing and selecting triangulation data.



Kalman filter

Kalman filter: filtering, by combining the (non-chronological) measurements with the Hybrid Automaton ball model.



#1

t_m

#2

#B

$t_k \mathrel{!=} t_m$

No

#3

Yes

updateState

Yes

$t_k < t_m$

No

#4

predictionState

runModel (foward)

runModel (backward)

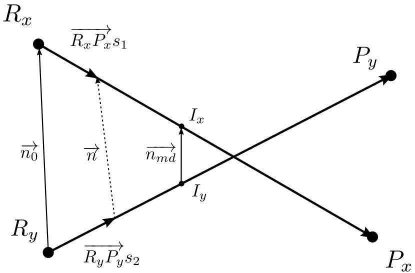
t_k ++

t_k --

Legend	
t_k	Kalman Filter Time
t_m	Timestamp

Minimum-Distance Algorithms & Model Update

Minimum-distance algorithms: combining triangulation data into 3D positions.

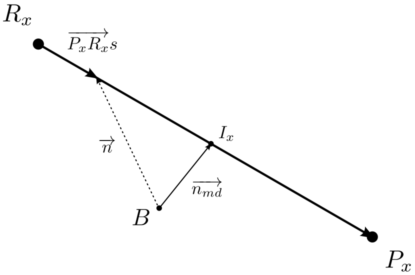

R_x
$\overrightarrow{R_xP_x}s_1$
P_y
I_x
$\overrightarrow{n_0}$
$\overrightarrow{n}$
$\overrightarrow{n_{md}}$
I_y
R_y
$\overrightarrow{R_yP_y}s_2$
P_x

(d) Minimum-distance between lines (pairwise).

Model update: updating the 3D position to the current time.

Minimum-Distance Algorithms & Model Update

Minimum-distance algorithms: combining triangulation data into 3D positions.



(e) Minimum-distance between line and point.

Model update: updating the 3D position to the current time.

Results

Static Test

- Static test is used to:
  - validate the implementation of the algorithm
  - qualify the performance of the algorithm

Results

Static Test

- Static test is used to:
  - validate the implementation of the algorithm
  - qualify the performance of the algorithm

- The next sheet presents the algorithm applied to:
  - 2 robots with the algorithm running
  - 5 different ball positions; (x, y)
  - 2 different ball heights; $z = 0.11\ m \vee 0.42\ m$
  - 200 measurements (for each robot) are analysed (mean and standard deviation)

Results

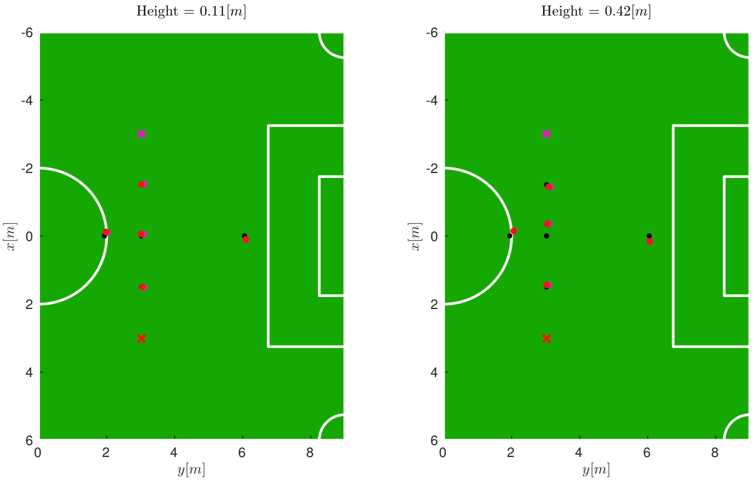
Static Test

- Static test is used to:
    - validate the implementation of the algorithm
    - qualify the performance of the algorithm

- The next sheet presents the algorithm applied to:
    - 2 robots with the algorithm running
    - 5 different ball positions; (x, y)
    - 2 different ball heights; $z = 0.11\ m \vee 0.42\ m$
    - 200 measurements (for each robot) are analysed (mean and standard deviation)

- Aside from accuracy of the position itself, asses difference between robots.

Results - Static Test



Overall: $\mu = 13.4\ cm$, $\sigma = 0.86\ cm$.

Conclusion

The Assignment:

Research and Verify whether cooperative sensing can be used for real-time 3D ball positioning using triangulation of omni-vision camera data.

Conclusion

The Assignment:

Research and Verify whether cooperative sensing can be used for real-time 3D ball positioning using triangulation of omni-vision camera data.

- A MATLAB simulation environment has been expanded, this has been used to improve an existing approach.
    - The new approach yields good results in simulation.
- The triangulation algorithm has been implemented on the robots.
    - The algorithms yields an accuracy of about 13.42 *cm* (10.6 *cm*).

Conclusion

The Assignment:

Research and Verify whether cooperative sensing can be used for real-time 3D ball positioning using triangulation of omni-vision camera data.

- A MATLAB simulation environment has been expanded, this has been used to improve an existing approach.
  - The new approach yields good results in simulation.
- The triangulation algorithm has been implemented on the robots.
  - The algorithms yields an accuracy of about 13.42 *cm* (10.6 *cm*).

- Dynamics tests still have to be performed.

Recommendations

Several recommendations are made, one of them is

- Optimal Selection of Time Instant based on Prediction Error Covariance of the Kalman filter.
    - Quantify the Measurement Error Covariance (R) for each measurement; based on e.g. robot-to-ball distance.

Recommendations

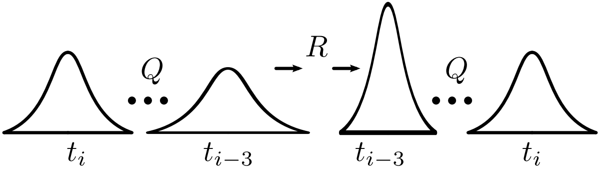
Several recommendations are made, one of them is

- Optimal Selection of Time Instant based on Prediction Error Covariance of the Kalman filter.
    - Quantify the Measurement Error Covariance (R) for each measurement; based on e.g. robot-to-ball distance.
    - Quantify the State Error Covariance (Q) for the model; probably through tuning.

Recommendations

Several recommendations are made, one of them is

- Optimal Selection of Time Instant based on Prediction Error Covariance of the Kalman filter.
  - Quantify the Measurement Error Covariance (R) for each measurement; based on e.g. robot-to-ball distance.
  - Quantify the State Error Covariance (Q) for the model; probably through tuning.
  - Allows to calculate the Prediction Error Covariance (P); calculate the time instant to minimize elements of P.



Demonstration!

Thank You!

Any:
Questions, Suggestions, Remarks, Answers, Replies, Opinions
Explanations, Words, Comments, Observations, Illustrations
Reflections, Unclarities

Or is there anything I:
Neglected, Ignored, Suppressed, Overlooked
Misunderstood, Did not say?