

Description of the 2023 Warthog Robotics SSL Project

Pedro A. G. Pizarro, Lucas F. Granela,
Leandro T. C. Andrade, Kenzo M. Sakiyama,
Rafael G. Lang, Ivan N. da Silva, and
Roseli A. F. Romero

Warthog Robotics
University of São Paulo at São Carlos
400 Trabalhador São-carlense Ave, São Carlos, São Paulo, Brazil
info@wr.sc.usp.br
<https://www.wr.sc.usp.br>

Abstract. This paper presents the Warthog Robotics Magic project and its main improvements made to this RoboCup SSL project during last year. During the last research and development cycle, a new implementation of the Kalman Filter was studied and it's currently under test in order to be used in 2023 RoboCup. Additionally, a new firmware architecture was developed as well as a fuzzy logic for ball possession determination.

Keywords: RoboCup, Firmware, Kalman Filter, Embedded Electronics, Warthog Robotics.

1 Introduction

The Warthog Robotics is a research and development Robotics group from the University of São Paulo at São Carlos. The group counts with over 100 members from several knowledge areas, such as Computer Science and Electrical, Mechatronic and Computer Engineering, and develops Robotics technologies in several areas, applying most of them at Robotics competitions. Due to budget limitations, the team was not able to participate in the last edition of the RoboCup, but is still competing in a local competition, with the following results in the Latin America RoboCup Open: the 2016 and 2018 first places, the 2017 and 2022 second places and 2019 third place.

The WR Magic is the RoboCup SSL robot of the Warthog Robotics group and its mechanical structure and electronic boards are the same from the last years, of which detailed information can be found in: [1], where a more detailed explanation about the mechanical locomotion and electronic system were given; [2], where the GEARSsystem and WRCoach software architectures were first presented; and [3], in which the Control algorithm was first documented, discussing the transition from the continuous PID implementation to its discrete version with a digital filter. Furthermore, the 2020s Warthog Robotics Team Description

Paper reviewed these topics with hardware diagrams and images, as well as a software architecture [4].

The next sections presents the newest improvements of the WR Magic project.

2 Improvements for 2023

2.1 Fuzzy Logic for Ball Possession Determination

Fuzzy Logic is a technique that makes it possible to represent computational models that have some degree of uncertainty or imprecision. This technique allows the developer to design software that represents algorithms closer to the way human reasoning works, with a formal methodology [5]. Basically it consists of 4 main modules: fuzzification and defuzzification blocks, where the inputs are transformed between the Fuzzy and time domain; rule-base, in which the knowledge is held in form of set of rules to control the system; inference mechanism, where the inference process occurs, based on the rule-base block and system inputs.

In a robot soccer game in the RoboCup Small Size League competition, there are several moments when it is difficult for the software system to identify the actual current game state. In a real game, many times the robots of the two teams will be facing each other with the ball in the center of the table, but cannot decide which of them has possession. For these moments, the Fuzzy Logic technique was developed by the Warthog Robotics team, since it can better abstract this state to a human reasoning that has a high degree of uncertainty.

For this it was created two Fuzzy Inference System (FIS) files, prepared in MATLAB that contain the system information, inputs, output and the rules. In one of them were placed the relative velocity information between the ball and the verified robot, where it is seen if the ball is moving away "a lot", "a little", or if it is getting closer. The other file contains only the relative distance information. The junction of these two fuzzy systems allows our code to have a better abstraction of what would be "ball possession" in real life and allows us to use this distinction in our strategies. With the fuzzy values trained by the group of data, it can be use into the strategy module without having to modify the structure of the code [6].

2.2 New Kalman Filter Implementation

The Kalman Filter is an efficient mathematical algorithm applied in many areas, such as, control systems, robotics, navigation and extensive areas related to accurate estimation of a system state. The technique, pioneered by Rudolf Kalman, was developed in 1960 and it is still used due to its efficiency, fast response and robustness to system disturbances. Rudolf Kalman created an algorithm that receives measurements and, combined with a model of the object's motion, it estimates the object's position and velocity. The Kalman filter works by update iteratively two sets of equations: the state equations, which describe the behavior

of the system, and the measurement equations, which describe how the measurements are related to the system state. Therefore, it produces an estimate of the current state of the system, along with an estimate of the uncertainty of the state estimate. The estimation is then used to update the state equations, which are utilized to predict the system state at the next iteration [7]. The process is then repeated, with the estimate being refined based on the latest measurements and predictions.

The Kalman Filter is used in WREye, our vision data processing module used to predict ball and robots positions. This year, a new filter is expected to be used, with new methods to estimate robot and ball position.

The old implementation the filter was used to predict the position, velocity and acceleration of the robots, in addition, camera interference and packet loss were considered systematic errors in the system equations. Using this, it was realized that the system used was too sensitive to disturbances and so it was decided to change the approach. In the new system, however, the acceleration was considered part of the error, which despite making the system converge slower, allows more stability and, with a few corrections to fix the convergence, it was built a more stable and robust algorithm compared to the previous one.

The Multidimensional Kalman Filter that Warthog Robotics is expecting to finish implementing consists of predicting the robot and ball position on a XY plane. The following equations define the main loop of the filter:

– **The State Extrapolation Equation:**

$$\hat{x}_{t,t-1} = F\hat{x}_{t-1,t-1} + Gu_{t-1} + w_{t-1} \quad (1)$$

– **The Covariance Extrapolation Equation:**

$$P_{t,t-1} = FP_{t-1,t-1}F^T + Q \quad (2)$$

– **The Measurement Equation:**

$$z_t = Hx_{t-1} + v_{t-1} \quad (3)$$

– **The Kalman Gain:**

$$K_t = P_{t,t-1}H^T(HP_{t,t-1}H^T + R) \quad (4)$$

– **The State Update Equation:**

$$\hat{x}_{t,t} = \hat{x}_{t,t-1} + K_t(z_t - H\hat{x}_{t,t-1}) \quad (5)$$

– **The Covariance Update Equation:**

$$P_{t,t} = (I - K_tH)P_{t,t-1}(I - K_tH)^T + K_tRK_t^T \quad (6)$$

The output from the filter is the result of the state update (Equation 4) and state covariance update equations (Equation 6) and it is updated on every interaction. These provide the combined estimate from the model (prediction equations) and latest observation (measurements). The state matrix provides the mean value of the distribution for each state variable, and the covariance matrix provides the variances [8].

Therefore, using the parameters obtained by the steps above, a higher accuracy output compared with Warthog's current implementation is expected.

2.3 Firmware architecture

The RoboCup SSL environment is, as many other embedded system applications, a real-time environment, with response time deadline requirements. Therefore, a carefully crafted firmware architecture that complies with response times and determinism is needed. In this case, the group's new firmware implementation is a Real-Time Operating System (RTOS) based bare-metal approach that stays in between a commercial RTOS solution and a simple bare-metal software.

A RTOS consists of a kernel that is capable of guaranteeing response time deadlines, these being labeled sometimes as soft, firm or hard real time requirements, with its differences being based on the the application performance results if a deadline is missed [9]. Meanwhile, a bare-metal solution does not require a kernel with a scheduling algorithm and context switching. Thus, the only way the main loop code execution could be stopped is by a fired external to the code interrupt [10].

In the new implementation, however, there isn't a proper kernel with a task scheduling algorithm and context switching. Nonetheless, there are isolated modules that can be controlled by a pseudo-kernel that is capable of stopping and restarting each module independently. So, currently, the Warthog Robotics firmware is transitioning from a simple bare-metal supper loop approach to a full RTOS based architecture.

The development of this complete version is under development, with the aim to result in the deployment of a state-of-the-art generation RTOS firmware from an open source solution such as FreeRTOS or ChibiOS/RT. The most complex part will be to port theses modern RTOSes to a not so contemporary 16-bit digital signal controller, a dsPIC33FJ128MC804.

Another complex topic that arises from the usage of a RTOS firmware, and shall be studied in future TDPs, is the jitter, know as a non-constant reaction time of the system. Due to the fact that there is a overhead added to the system as a result of context switching and concurrency abstract data types such as semaphores, the response times may varies subject to the current code section is running and with system inputs. This directly impacts the real time requirements of the application, possibly degrading the result performance of the robot, if ignored. An example of this behavior is found on the article "Real Time Systems Concepts" in ChibiOS page, and is included here for more clarification.

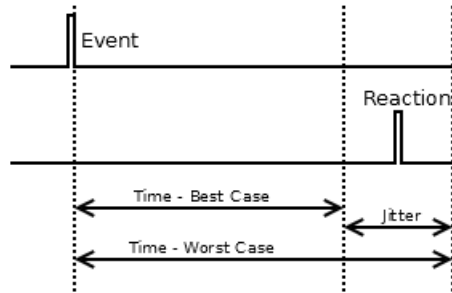


Fig. 1. System non-constant reaction time, known as jitter. Source: [11].

3 Conclusion and Future Work

Following the last years budget directives of the group, the development made in 2021-2023 focused in software and firmware. Minor improvements and bug fixes were made to the WRCoach software during last years, including during and after last years Latin America RoboCup Open competitions.

The ball possession system improved the systems responses during the game, and the new Kalman filter under development is expected to enhance in-game passes and kicks accuracy.

The current robot is robust, reliable and provides an excellent platform to the strategy systems. The implemented firmware architecture will provide a new program environment for further refinement, for example on the telemetry system.

Future works shall continue focusing in software improvements, including the finalization of the 2020s TDP opponent path prediction system, and a new WRCoach architecture.

4 Acknowledgments

The authors would like to thank all Warthog Robotics team for the hard work and friendship; the University of São Paulo for the facilities and financial support; SolidWorks for the technical sponsorship; and all others that helped us during the project.

References

1. Lang, R.G., Bernardo, A.M., Oliveira, G.C., Menezes, H.B.B., Ramos, L.C., Roque, L.G.S., Silva, I.N., Romero, R.A.F.: Description of the Warthog Robotics SSL 2015 Project. In: 2015 RoboCup.
2. Lang, R.G., Oliveira, G.C., Menezes, H.B.B., Rosa, N.S., Correa, R.A., Gomes, V.H., Silva, I.N., Romero, R.A.F.: Description of the Warthog Robotics SSL 2016 Project. In: 2016 RoboCup.

3. Lang, R.G., Oliveira, G.C., Barbosa, H.J., Rosa, N.S., Cepeda, B.H., Siqueira, A.H., Silva, I.N., Romero, R.A.F.: Description of the Warthog Robotics SSL 2017 Project. In: 2017 RoboCup.
4. Lang, R.G., Oliveira, G.C., Barbosa, H.J., Pinto, A.H.M., Pizarro, P.A.G., Silva, I.N., Romero, R.A.F.: Description of the 2020 Warthog Robotics SSL project. In: 2020 RoboCup.
5. Passino K.M., Yurkovich S., Fuzzy Control. p. 10-13 In: 1998.
6. Abiyev R.H., Günsel I., Akkaya N., Aytaca E., Çagman A., Abizada S.: Robot soccer control using behaviour trees and fuzzy logic. In: 2016.
7. F. Farahi, H.S. Yazdi: Probabilistic Kalman Filter for Moving Object Tracking. In: 2020.
8. J.-M. Jeong, T.-S. Yoon, J.-B. Park: Kalman Filter Based Multiple Objects Detection-Tracking Algorithm Robust to Occlusion. In: 2014.
9. Laplante P.A., Ovaska, S.J.: Real-Time Systems Design and Analysis: Tools for the Practitioner. p. 5-8. In: 2012.
10. Amos B., Hands-On RTOS with Microcontrollers: Building real-time embedded Systems using FreeRTOS, STM32 MCUs, and SEGGER debug tools. p. 13-16. 2020.
11. Real Time Systems Concepts - ChibiOS Article, available at <https://www.chibios.org/dokuwiki/doku.php?id=chibios:documentation:books:rtfonde:concepts>. Accessed at 21/03/2023.