Immortals 2013 Extended Team Description

Iran University of Science and Technology

Mohammad Reza Niknezhad[1], Ali Salehi[2], Ali GhaziMirSaeed[3], Sadegh Salehi[2], Mohammad Hossein Fazeli[4], Mohammad Tabasi[4], Mustafa Talaeezadeh[5]

1 Department of Computer Engineering of Iran University of Science and Technology
2 Department of Electrical Engineering of Sharif University
3 Department of Mechanical Engineering of Tehran University
4Department of Mechanical Engineering of Iran University of Science and Technology
5 Department of Computer Engineering of Shahid Beheshti University

Abstract. This paper presents some of the hardware and software improvements of the Immortals small size robotic team based in Tehran, Iran. The system is designed under the Robocup 2013 rules in order to participate in Robocup 2013 competitions in Eindhoven.

1 Introduction

Some achievements of Immortals team that seemed innovative and useful are explained in this paper. This paper is meant to improve these methods by sharing them with the community, and help new teams achieve the minimum requirements for paticipating in the competitions.

With some minor modifications and adjustments in hardware aspects, the main focus of the team is more sophisticated AI and improvements in the software side, that are described briefly in this document.

First some optimizations in electronics firmware are introduced, and then some software engineering concepts implemented in this year's robotic overall design are explained including an introduction to NOK-RRT and improved monitoring systems.

2 Hardware

This year's main focus is put on making robots more robust and reliable. It's achieved with an implementation of varieties of safeguards and monitoring systems, including odometers for different parts of hardware built in and hardcoded into the FPGA that is also used as the central processing unit for each robot. Any fault and error detected by this system, such as sensor reading errors will immediately be taken care of by a predefined series of action to avoid the propagation of the fault or error into other areas of work, and simultaneously leads to a report and log generation, sent onto monitoring PC, and also can be read through wireless connection to each robot after every match.

A Xilinx XC3S400 chip functions as the only processor on the main board and operates telecommunication, decodes IR sensors data, drives motors and executes PID controller. This chip was chosen because of its low power consumption and its huge logic gate numbers in comparison with other similar products.

A TSK-3000A soft processor is implemented in the FPGA and operates as the main processor. The following features are implemented inside the TSK3000 processor:

- An error detection system, that detects many errors, including low-voltage situations, voltage spikes, short circuits, over-temperature, daughter boards' malfunction, packet drops, sensor reading errors, motor stuck situations and much more. Then these errors are written to the flash memory for later diagnostics. But some of them are categorized as critical, and the processor halts the function of the malfunctioned part, and report the situation by making an error-specific sound, turning the STOP led on, and sending it over to the main computer. These errors should be cleared by using the diagnostics software, and the person erasing it should solve the problem before clearing it.
- Acting as an odometer, calculating the usage of each part, including BLDC motors, voltage booster unit, servo motor, rotary encoders and the wheels. These data then are used by the team, to know when to change or check each unit.
- Providing real-time diagnostics data over wireless link, such as each motor speed, voltages of some critical points, state of debugging switches, IMU data and voltage booster state.
- PID control loop with torque converter for each motor. The control loop frequency is 1.2 kHz, and delay time is 122 µS. The loop is so time-critical, and so is written in pure assembly.

Fig. 1. Main electronic board

The odometer designed within each robot helps improving the overall maintenance of robots and assures each robot that is sent to the field will work properly as it intended to.

3 Software

This year, the main target of the software development was developing more robust navigation methods. It lead to a joint Kino-dynamic planner called NOK-RRT.

3.1 Dynamic Safety Search

Dynamic Safety Search (DSS) is a multi-agent sampling-based dynamic path planning method. In contrast to positional planners (e.g. ERRT) it considers each agent kinematic parameters and current velocities in planning. DSS is proposed by J. R. Bruce [1], and is an improvement over the well-known Dynamic Window method. The main improvements are:

- Replacing grid-base sampling with random sampling
- Being multi-agent, meaning it can generate a safe path for any number of agents

The set of possible accelerations used in DSS calculation is also recalculated, using our actual robot model which is based on BLDC motor equation. The author proposed that acceleration space plot of the set is likely to be a partial ellipse, but based on our computations, it actually is a diamond (fig. 5).

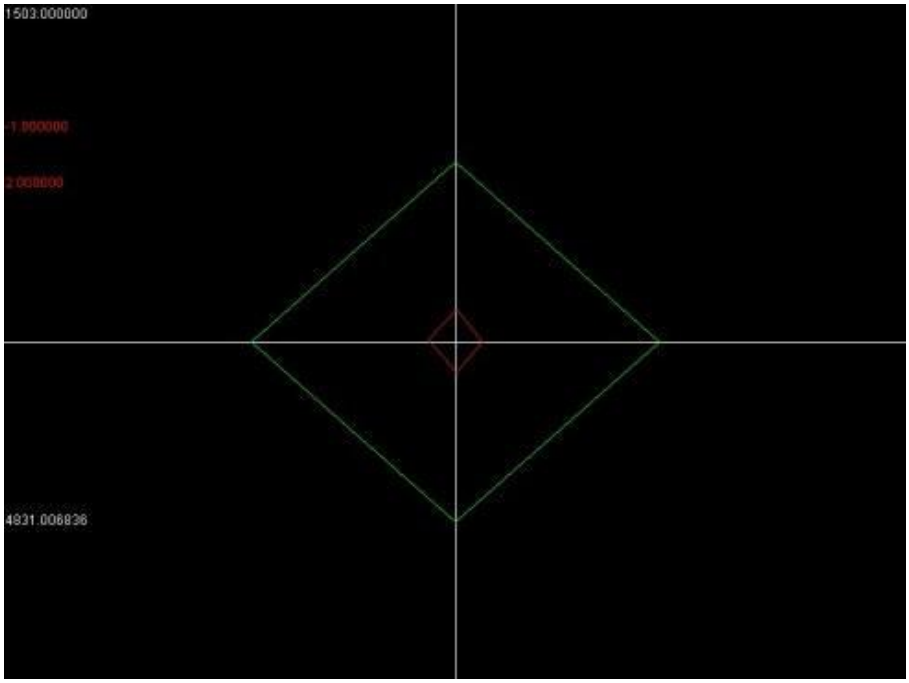

Fig. 2. Acceleration space plot of the set based on BLDC equations

In our AI software, the output of the ERRT planner is fed into the motion planner, and an acceleration vector is generated for the robot. Then this vector is passed to the DSS as the input, and a safe acceleration vector f is generated.

On the implementation side, the DSS algorithm generates the result for all robots simultaneously, so it cannot be parallelized. But because of its anytime basis, it can be terminated whenever required, and it will give the best found result. In our experiments, it almost always gives good enough results in the given 0.2 mS time.

3.2 Near-Optimal Kino-dynamic RRT

After introduction of SB-RRT in 2011[2] and it's improvements by ANN in 2012[3], in 2013 a new approach for path planning is tried and is still in its test phase. In this work, the capability of RRT in handling higher degrees of freedom is considered noticeably in order to achieve better maneuverability. Combined with SB-RRT, Near-Optimal Kino-dynamic RRT (abbreviated as NOK-RRT) is believed to gain maximum maneuverability and safety altogether for robots. NOK-RRT potentially can unify path, motion, and Kino-dynamic planning into one phase and process.

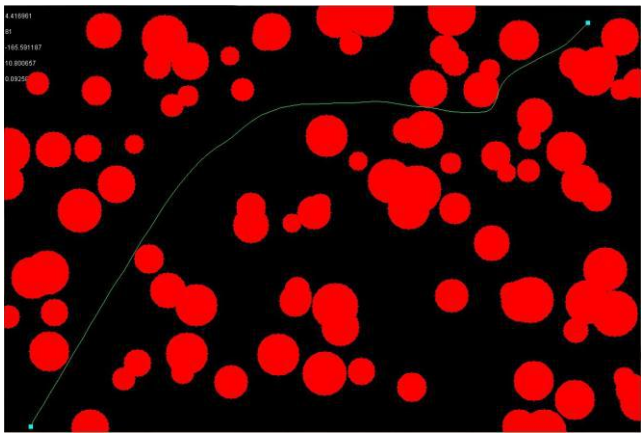

Fig. 3. NOK-RRT planning in a heavy occluded environment

In NOK-RRT, other constraint rather than obstacle and obstacle-free spaces are defined. These constraints are limits in differentiation of velocities and accelerations, forces and torques of actuators and as processed together, the modeled robot; hence NOK-RRT plans in a 4D state space. A benefit of being a joint planner is decreasing the probability of failure in highly dynamic and obstacle rich environments such as SSL.

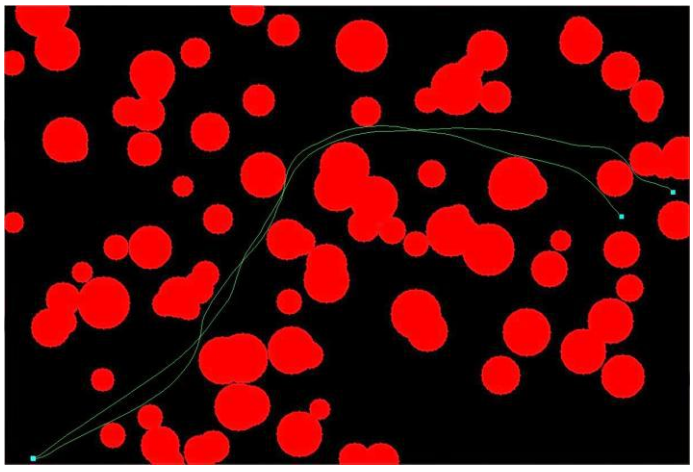

Fig. 4. Multiple iterations of NOK-RRT using GPGPU

It also takes advantages of heterogeneous parallel computations, introduced in 2010[4]. This means NOK-RRT plans many times in order to find the shortest path, in terms of path traversal time, not the positional length. Because of the dynamic limits each mobile robot has, the acceleration, deceleration, and cruising speed should be considered to compute the total traversal time. So the least traversal time, which seems to be the optimal solution, does not always correspond to the shortest path.

The multiple iterations are done on the GPU, using GP-GPU RRT framework described in 2010[4] which lead to more-than-sufficient planning iterations. Thus, the output path is near optimal in terms of both failure rate and traversal time.

3.3 Soccer Simulator

To avoid possible damages of running the actual robots to test new algorithms, such as new path planners, a soccer simulator is developed. The simulator uses internal physic-based robot model described in 3.1. The Newton Dynamics physics engine is used because of its deterministic calculations [5].

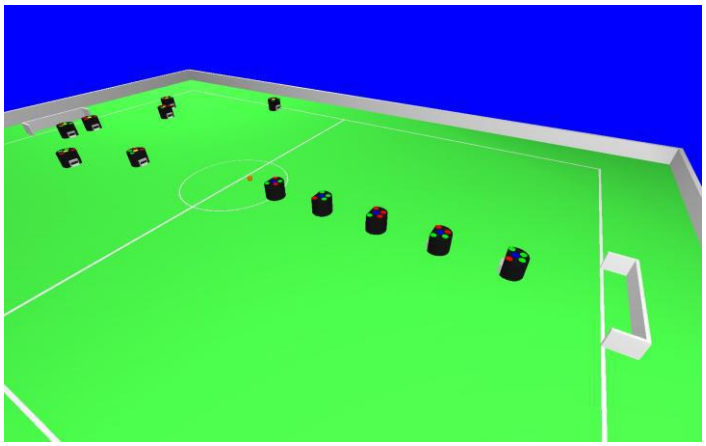

Fig. 5. An example shot taken from soccer simulator

The simulator outputs the data with the same protocol as the SSL-Vision, so the AI is unaware whether the incoming data is from the actual vision, or from the simulator. The input command data also uses the same protocol as our wireless interface board. For both vision data, and robot command data, the noise is simulated using white noise pattern, and the noise amount can be changed to test algorithms in different environmental situations. This way the same filtering and fusion can be used in both real robot and simulation cases.

References

1. Bruce, J.R.: *Real-Time Motion Planning and Safe Navigation in Dynamic Multi-Robot Environments.* PhD thesis, Carnegie Mellon University

2. Salehi, A., Niknezhad, M.R., Kamali, E., MirSaeed, A., Fazeli, M.H., Piran, Y., Salehi, S., Khuzani, M.: *Immortals 2011 Extended Team Description.* In: Proceedings of Robocup 2011.
3. Salehi, A., Niknezhad, M.R., Kamali, E., MirSaeed, A., Fazeli, M.H., Piran, Y., Salehi, S., Khuzani, M.: *Immortals 2012 Extended Team Description.* In: Proceedings of Robocup 2012.
4. Salehi, A., Niknezhad, M.R., Kamali, E., MirSaeed, A., Fazeli, M.H., Piran, Y., Salehi, S., Khuzani, M.: *Immortals 2010 Team Description.* In: Proceedings of Robocup 2010.
5. *Newton Game Dynamics*, newtondynamics.com.