RRC Team Description paper for robocup asia-pacific 2017

Islamic Azad University Central Tehran Branch
Tehran – Iran
Dr. Shahram Javadi, Faramarz Daemi, Mehrnaz Payami, Niloofar Jahednami, Sajjad Ahangar, Ata vagheffar, Soheil Sepahyar, Mohammad Damirchi, Mohammadsadegh Bashirian, Amirmohammad Taghavi, Ali Taghavi, Amirreza Lahooti, Omid Mohammadi

@Work League

fdmxfarhan@yahoo.com, f_daemi@yahoo.com
http://fdmxfdmx.blog.ir

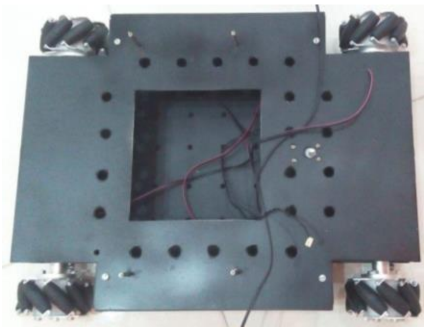
Abstract. In this paper we present and describe about RRC @work team for robocup asia-pacific 2017 in Thailand. This team has started its work from 2016 and now wants to participate in robocup asia-pacific 2017. This team made of some students who any of them has responsibility of one part of robot. This robot includes mechanical, electrical (hardware), software part and artificial intelligence.

Introduction

Last year, RRC team participated in Iran open 2016 computation and now it is going to participate in robocup asia-pacific 2017. Some requirements to reach this target are achieved by redesigning the electrical and mechanical mechanisms. Moreover, simple learning and optimization approaches are employed in the way of more dynamic play. This paper is organized as follows: First of all, the software architecture which includes our approaches in high level strategies, defense, The Electrical designs including AVR micro controller, and other accessories. [1]

Mechanical Design and platform

One of the main parts of our robot was mechanical systems. First the robots platform had been designed with computer and then it had cut on an aluminum plate. For designing the platform we used Solid Works software that helps us design every each parts of robot and then match them to each other. After platform designing completed every parts cut by a laser-cut machine (CNC) from an aluminum plate and eventually every parts will be matched to each other. Then other parts will be assembled on the main platform such as motors, arm, hardware part and etc.



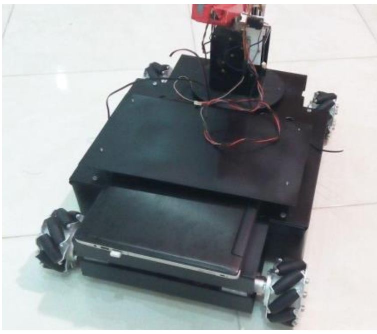


Figure 1 mechanic and platform

Its height and width are 80cm*50cm and its weight is about 20kg and it has the ability to carry 20kg itself. We should use a computer or a laptop for image processing part and the robot should have a place to put the laptop in it. So a place had been designed in the back of the robot for placing the laptop. The robots arm had designed by computer and then made like the platform part. This arm includes four free axes that can help the robot to catch the objects and carry them so easy. [2]The wheels that we used are a kind of mecanum wheels. The reason that why we used this kinds of wheels is that this wheels gives robot the ability two move to any direction with no rotation. Robots motors are Buhler kind. These motors with their own gearboxes velocity and power are good enough for this robot. For getting feedback from each motors and PID control we used external encoders for each motor.

Type	Nominal voltage	No load speed	Nominal speed	No load current	Starting current	Max. efficiency
BUHLER	24V	35 rpm	25 rpm	270 mA	5.7A	89%

Figure 2 motor and mecanum wheel

As you know, the linear velocity is quickly moving on a circle of radius r and rotates at a constant velocity v. The speed at any moment is perpendicular to the direction of motion. Angular velocity is the speed at time t1 and t2 which animates the circle and the angle is θ1 to θ2 period. Angular velocity and linear velocity obtained from the following relationships:

To calculate the actual speed of the engine must respect and apply them to our gearboxes. As you know, the task of slowing down the transmission of torque. The engine and gearbox are increasing. For other movements, as well as the outcome is speed.

r = linear velocity
w = angular velocity
A round rev=*rev v=rw*

$$\overline{w}=\frac{\Delta\theta}{\Delta t} \quad \frac{rad\ or\ rev}{s}$$

$$1\ \text{rev}=2\pi\ rad \rightarrow 1\frac{rev}{min}=2\pi\ \frac{rad}{min}$$

The above equation can be concluded that the average angular velocity is equal to:

(Rpm= round per minute)

$$w=\frac{rpm}{60}\times 2\pi$$

Nominal Speed=655.1 *rpm*

$$w=\frac{655.1}{60}\times 2\pi$$

$$\pi \approx 3.14 \qquad w=\frac{655.1}{60}\times 2\pi \quad =10.9183\times 2.8$$

$$v=\sqrt{(2v)^2+\ (2v)^2\ }=2v\sqrt{2}=2.88\ {}^{m}/_{s}\ =\ 68.5667\ \frac{m}{s}\quad r$$

$$=0.15\ cm \quad v=0.15\times 68.5667\ =10.2850$$

$$v=100.2850\ \frac{cm}{s} \qquad v=1.02\ {}^{m}/_{s}$$

First the robots arm designed by computer and after so many tests the main arm had been chose for making. The motors that had been used in the arm are kinds of servo motors. These motors has the ability to get order from robot and stop to the degree that robot tell them. The other motors that we could use instead of servo motors are dynamixle, step motors, and DC motors with some special gearboxes. The reason of not using step motors is that they need a high current to work and it is too much for the robot. About DC motors we don’t have any feedback from DC motors. The best choice was using servo motors. The griper for catching the object has the same motors as the arm. A servo motor open and close the griper. The griper can get an object in 10cm and move 5kg.

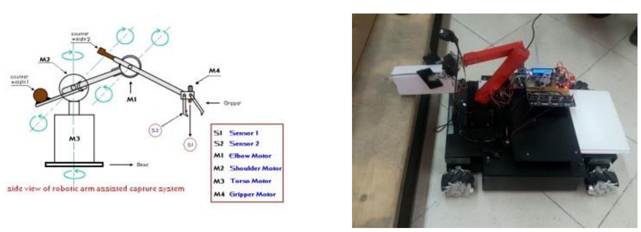


Figure 3 robots arm

Hardware/ Electrical Design

The electrical circuit designed by computer and then printed. After the PCB printed electrical pieces soldered on it such as connectors, drivers, LCD and etc. All the electrical parts of robot such as motors, encoders and sensors connect to the microcontrollers with connectors. The robots microcontroller is AT-mega2560 which is from AVR microcontroller kind. The main reason of using this kind of microcontroller is their easy working and its options. For using the motors the robot have a motor driver for any motors. These drivers are from L6203 kind. It had been chose after testing and comparing other drivers. The output current for this drivers is about 4A. With giving a pulse from microcontroller to the drivers it can control the motors speed.

Compass Module can show us deviate from the North Pole / South (magnetic field) as a number from 0 to 255. With some play on this value we made it a number from -128 rights to +128 left and 0 for center. CMPS11 was our compass sensor called a three-axial magnetic sensor, a three-axis gyroscope and a three-axis accelerometer is formed using a Kalman filter, tilt the board will be compensated compass automatically. The I2C module registers with different addresses to which we have access to different information. The compass microcontroller serial I2C the SCL and SDA by two base AT-Mega16 microcontroller is connected to two arbitrary bases. For detecting the distance of the walls and other things we used four SRF-05 ultrasonic sensors. This sensors can detect distance from 1 to 300_{cm}. actually we send a 10 micro second pulse to it and then it send us a pulse and with measure this pulse we can convert it to inch or centimeter.

We used a USB to serial convertor module for communication to the main program in visual studio software. The visual studio program is by C# programing language. All commands come from this program and then the other part do this commands. Our AVR micro controller has two serial ports but we used only one serial port for this communication.

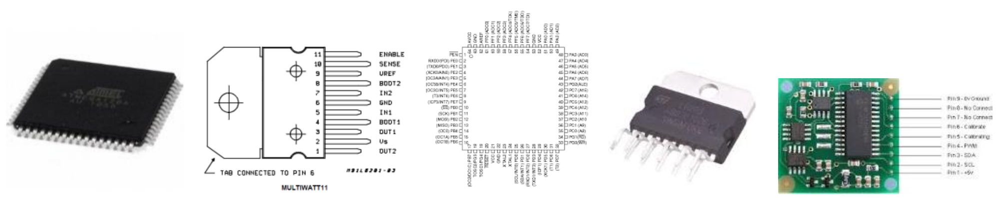


Figure 4 electrical parts (microcontroller, driver, compass...)

Image processing/Vision System

In imaging science, **image processing** is processing of images using mathematical operations by using any form of signal processing for which the input is an image, a series of images, or a video, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image.[1] Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Images are also processed as three-dimensional signals with the third-dimension being time or the z-axis. Image processing usually refers to digital image processing, but optical and analog image processing also are possible. This article is about general techniques that apply to all of them. The *acquisition* of images (producing the input image in the first place) is referred to as imaging.[2] Closely related to image processing are computer graphics and computer vision. In computer graph-ices, images are manually *made* from physical models of objects, environments, and lighting, instead of being acquired (via imaging devices such as cameras) from *natural* scenes, as in most animated movies. Computer vision, on the other hand, is often considered *high-level* image processing out of which a machine/computer/ software intends to decipher the physical contents of an image or a sequence of images (e.g., videos or 3D full-body magnetic resonance scans). In modern sciences and technologies, images also gain much broader scopes due to the ever growing importance of scientific visualization (of often large-scale complex scientific/experimental data). Examples include microarray data in genetic research, or real-time multi-asset portfolio trading in finance.

The most important part of programing in this robot is image processing. To do this we used Open-CV library in C# language. The Open-CV library for C# is Emgu-CV. The commands are the same but the names have some differences. At first the robot should get to the object's table and then process on images that are getting from the webcam camera. After detect the object and find the place it is, it should catch it with its arm. For doing that it send's some commands to the AVR microprocessor with serial communication and the micro controller do them. After getting the objects it should carry them to somewhere else. Moving system is with distance detecting. We have some distance sensors around our robot that can detect the walls,

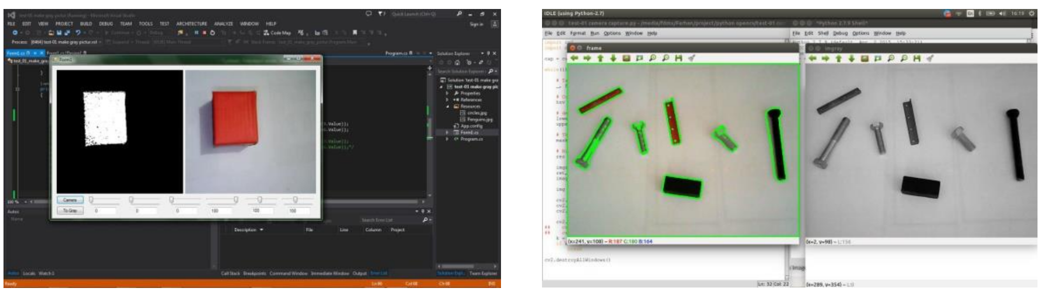


Figure 5 image processing

Navigation

The theories behind robot maze navigation are immense - so much that it would take several books just to cover the basics! So to keep it simple this tutorial will teach you one of the most basic but still powerful methods of intelligent robot navigation. We used a different camera under our robot that can detect the landmarks. To getting to the landmarks we have a laser scanner that gives us 2D plan of the land and the robot can move to the landmarks with process this plans. Most laser scanners use moveable mirrors to steer the laser beam. The steering of the beam can be *one-dimensional*, as inside a laser printer, or *two-dimensional*, as in a laser show system. Additionally, the mirrors can lead to a *periodic* motion - like the rotating *mirror polygons* in a barcode scanner or so-called *resonant galvanometer* scanners - or to a *freely addressable* motion, as in servo-controlled galvanometer scanners. One also uses the terms *raster* *scanning* and *vector scanning* to distinguish the two situations. To control the scanning motion, scanners need a rotary encoder and control electronics that provide, for a desired angle or phase, the suitable electric current to the motor or galvanometer. A software system usually controls the scanning motion and, if 3D scanning is implemented, also the collection of the measured data. In order to position a laser beam in *two dimensions*, it is possible either to rotate one mirror along two axes - used mainly for slow scanning systems - or to reflect the laser beam onto two closely spaced mirrors that are mounted on orthogonal axes. Each of the two flat or polygonal mirrors is then driven by a galvanometer or by an electric motor. Two-dimensional systems are essential for most applications in material processing, confocal microscopy, and medical science.

The main algorithm to get to a place or a table has four main steps.

1. **Create a Discretized Map** all the tables and walls will be saved in a matrix as a map.
2. **Add in target and Robot Locations** the robots and its target will be saved in the created map.
3. **Fill in Wavefront** find the way with using wavefront algorithm.
4. **Direct Robot to Count Down**

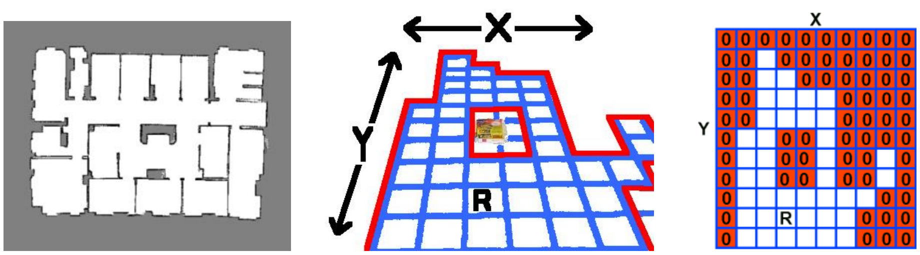


Figure 6 maping

Figure 7 navigation

References

[1] P.de Kok,N. Girardi, A. Gudi, C. Kooijman, G. Methenitis, S. Negrijn, N. Steenbergen, D. ten Velthuis, C. Verschoor, A. Wiggers, and A.Visser,"Teamdescription for RoboCup 2013 in Eindhoven, the Netherlands," Dutch Nao Team, Universiteit van Amsterdam & TU Delft, May 2013.

[2] N. Dijkshoorn, H.Flynn,O.Formsma, S. van Noort, C. van Weelden, C. Bastiaan, N. Out, O. Zwennes, S. S. Ot´arola, J. de Hoog, S. Cameron, and A. Visser, "Amsterdam oxford joint rescue forces - team description paper -virtual robot competition - rescue simulation league - RoboCup 2011," Universiteit van Amsterdam & Oxford University, July 2011.

[3] C. R. Verschoor, A. J. Wiggers, H. R. Oosterhuis, and A. Visser, "Maneki-neko - team description for iran open uav 2013 - teheran, iran," Intelligent Robotics Lab, Universiteit van Amsterdam, February 2013.

[4] S. Negrijn, J. Haber, S. van Schaik, and A. Visser, "UvA@Work customer agriculture order - mid-term report," Intelligent Robotics Lab, Universiteit van Amsterdam, Science Park 904 1098 XH Amsterdam, January 2014.

[5] J. van Enk, "Navigating youbot through a rose field with A*,"Project Report,Universiteit van Amsterdam, Science Park 904 1098 XH Amsterdam, August 2013.

[6] S. Koenig and M. Likhachev, "D* lite." in AAAI/IAAI, 2002, pp. 476–483

[7] L. Dorst, I. Mandhyan, and K. Trovato, "The geometrical representation of path planningproblems,"Roboticsand utonomous Systems, vol. 7, no. 2, pp. 181–195, 1991.

[8] M. Phillips, A. Dornbush, S. Chitta, and M. Likhachev, "Anytime incremental planning with e-graphs," in Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013, pp. 2444–2451.

[9] E. Corten and E. Rondema, "Team des-cription of the windmill wanderers," in Proceedings on the second RoboCup Workshop, 1998, pp. 347–352.

[10] P. Jonker, B. van Driel, J. Kuznetsov, and B. Terwijn, "Algorithmic foundation of the clockwork orange robot soccer team," in Algorithmic Foundations of Robotics VI. Springer, 2005, pp. 17–26.

[11] A. Visser, J. Sturm, P. van Rossum, J. Westra, and T. Bink, "Dutchaiboteam: Technicalreportrobocup2006,"December 2006.

[12] V. Spirin, S. Soffia Ot´arola, and A. Visser, "Amsterdam oxford joint rescue forces - team description paper - virtual robot competition - rescue simulation league - robocup 2014, jo˜ao pessoa - brazil," 2014.