KgpKubs Team Description Paper
Robocup SSL 2016

Abhinav Agarwalla, Kumar Abhinav, Arnav Jain, Kaustubh Mundhadha,
Dhananjay Yadav, Arpit Saxena, Soumyadeep Mukherjee, and Ashish Daga

Indian Institute of Technology, Kharagpur
West Bengal, India
`abhinaviitkgp1994@gmail.com`, `agarwallaabhinav@gmail.com`

Abstract. This paper describes the mechanical, electronic and software designs developed by Kharagpur RoboSoccer Students' Group (KRSSG) team in order to compete in RoboCup 2016. All designs are in agreement with the rules of Small Size League 2016. This is the first time we are participating in an international RoboCup event. This paper describes about Skills-Tactics-Play architecture implemented over Robot Operating System(ROS), dribbler/kicker design and embedded circuits.

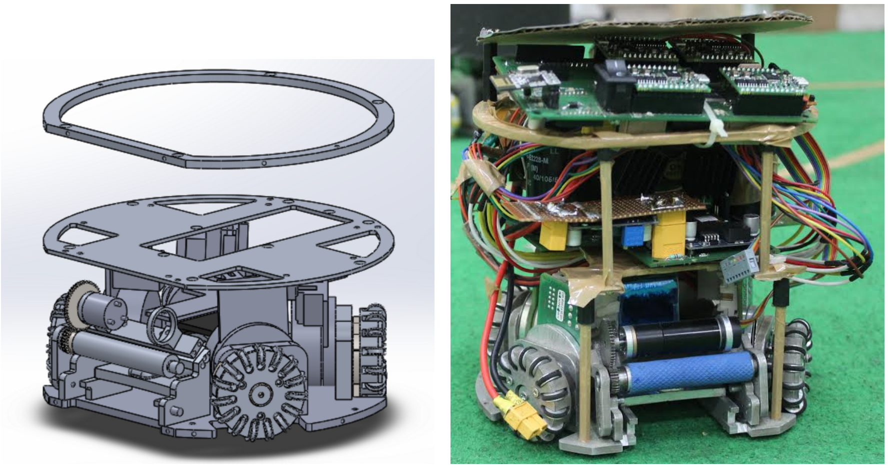
1 Introduction

KgpKubs is a small-size league soccer robot team from IIT Kharagpur, India. The aim of the research group is to make autonomous soccer playing robots. Students from all departments and years have been part of this including undergraduates and post-graduates. The principal investigator for the project is Prof. Jayanta Mukhopadhyay and it is also mentored by Prof. A.K. Deb, Prof. D.K. Pratihar and Prof. Sudeshna Sarkar. We have previously participated in FIRA RoboWorld Cup in the years 2013-2015 in the Mirosot League. In 2015, we secured Bronze position in the same. Thus, our team has the required experience in robot motion control, path planning, and behaviour design. However, we are a new entrant in the Robocup format and aim to emerge as a competent team in the upcoming Robocup 2016.

This work is organized as follows. The mechanical design of KRSSG robots is presented in section 2. Then the firmware and electrical project is presented in section 3. The software system is presented in section 4. Finally, discussion and future work are described in section 5.

2 Mechanical Design

This robot was designed and built using CAD (Computer Aided Design) and CAM (Computer Aided Manufacturing) softwares like Solidworks and Ansys. Moreover, extensive testing was done to validate the design.



(a) (b)

Fig. 1. Top view of 3D CAD model and the real bot.

2.1 Locomotion System

The locomotion system is a four wheel drive with back wheels inclined at 90 degrees and the front wheels inclined at 120 degrees to provide space for dribbler and kicker mechanism. The bot has a single centre of rotation and it comes well within the 18 cm diameter following the rules of Small Sized League Robocup competition (SSL).

An omni-wheel assembly is used which can move in any direction. There are 8 small sub wheels in every wheel which allows the wheel to move along the axis of rotation of the wheel. Spur gears are used between wheels and motors. The motor used for driving is Maxon EC 45.

2.2 Dribbling Mechanism

A circular dribbler made of Aluminium is used. Dribbler bar is directly driven by a brushless Maxon EC16 (16W) motor through gears with ratio of 4:1. To stop the ball from recoiling on impact, spring is used to dissipate its kinetic energy to ensure maximum ball handling. Rubber pad is used as dampener.

Our dribbler holds up to 20 precent of the ball when receiving a pass, and less when the ball is at rest or during normal dribbling. The ball also sticks entirely to the robot during the movement. This complies with the rules of SSL 2016.

2.3 Kicker System

The available solenoids are flat and wide, they have a fairly slow movement, so they are not suitable for shooting mechanisms. The solenoids for shooting should

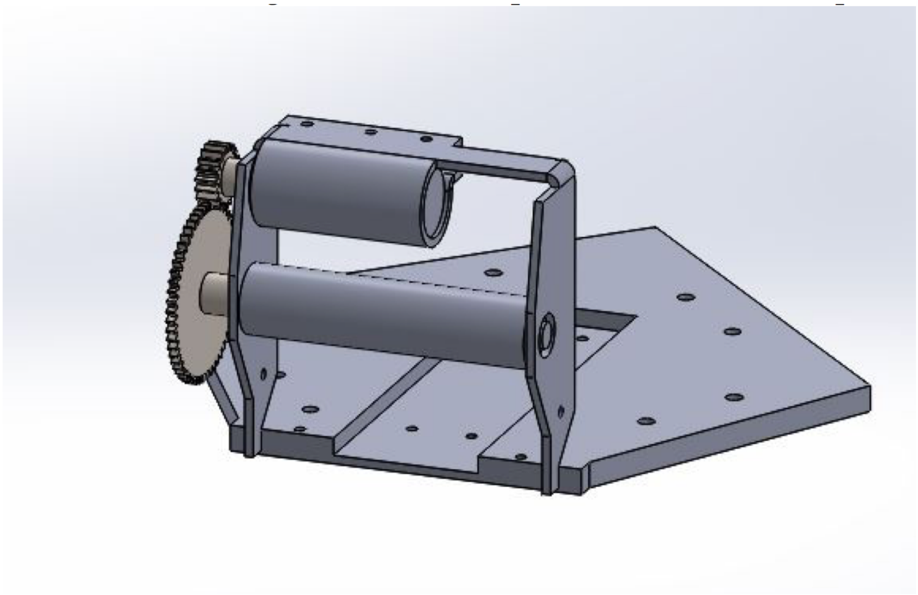


Fig. 2. Mechanical Design of dribbler.

be fast and should require a small space. So cylindrical solenoids are used which provides more power and cover much smaller area.

Straight Kicker

Straight kicker consists of solenoid and plunger.

- **Solenoid** The frame of solenoid is made of 6061-Aluminium alloy and is cylindrical in shape with an inner diameter of 11.4mm and an outer diameter of 12.5mm. Its length is 44mm and thickness is 0.55mm. The wires wound around it are 24 AWG wires and the number of turns is approximately 400. We assume that a constant current of 40 A flows in it throughout its application time.
- **Plunger** Straight kicker consists of a custom made plunger with magnetic material (pure iron) in the middle and aluminium upon the remaining length. The length of the iron part is 45mm.The diameter of plunger is 11mm. The dynamic analysis of a solenoid fed by constant current was done using MATLAB 2013a, FEMM 4.2 and ADAMS software.

Chip Kicker

2 plunger designs were used. One with a thickness of 3.52 mm and length of 64mm (Plunger 1) and the other with a thickness of 2.67 mm and length of 70.75 mm (Plunger 2). Both plungers are 33.8 mm wide. Plunger 1 was observed to provide more energy(4.387 J) than Plunger 2(3.053 J), and hence was finally selected.

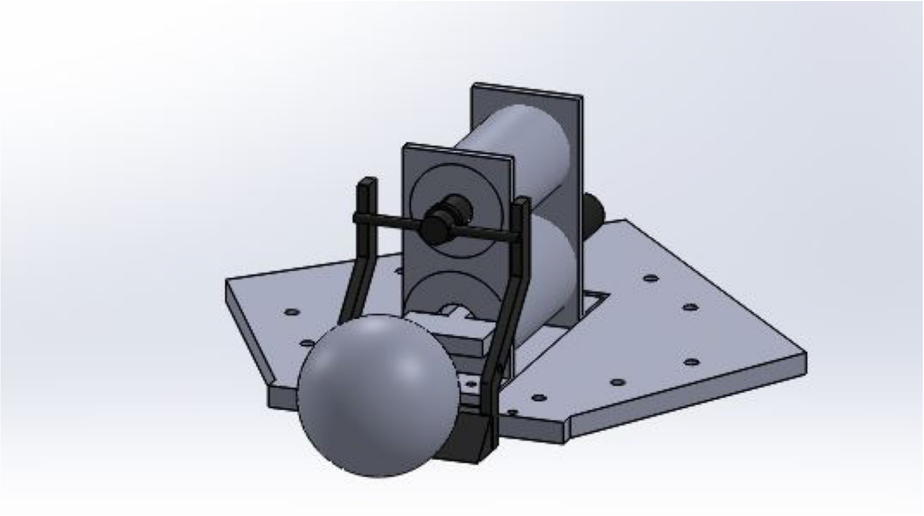


Fig. 3. Kicker Assembly.

3 Embedded System

3.1 Main controller board

The embedded architecture of the bot comprises of two circuits. One is mother circuit and other is the kicker circuit. The mother circuit, which is a 4-layered circuit, comprises of STM32f407vg microcontroller to control the motor controllers, inertial measurement unit (MPU 6050), communication with the master computer, proximity sensor and controlling the kicker circuit. For communication with the master computer, we have used nrf24L01+ (and XBee as a standby option). The proximity sensor has been used to sense the position of the ball when in front of the dribbler. The mother circuit sends a triggering signal to the kicker circuit in an external interrupt. We have used dedicated Escon 24-2 motor controllers for motor control. Each motor controller has its own microcontroller which has inbuilt tuned PID controller.

3.2 Power Circuit

We have used one 5-cell lipo 2200 mAh for power supply. The same battery has been used to power the whole robot. The battery operates at 18V. LM2596 has been used for regulating and supplying 12V to the motor controller and 5V to the rest of the circuit. To supply 3.3 volts to the main board, LM1117 voltage regulator has been used.

3.3 Motor Controller

Escon 24-2 motor controller module is used to control the BLDC motors used in our robots.The module has on chip auto-tuned PID controller to have precise speed control of the BLDC motors. It has a 12-bit resolution for PWM signal. It can successfully control and run the BLDC motor upto 10,000 rpm. The same motor controller has been used to control both the wheel and dribbler motors.

3.4 Communication Module

We have chosen an ultra low power 2.4 Ghz RF module, nrf24L01+, for communication with the master computer.The module can send maximum of 32 bytes of data.The first byte is used for team-id. Rest 30 bytes are used to send 4 wheel velocities, dribbler velocity and kicking command of 6 robots.The corresponding velocities are extracted at the robot side using bot id which is set using a switch.

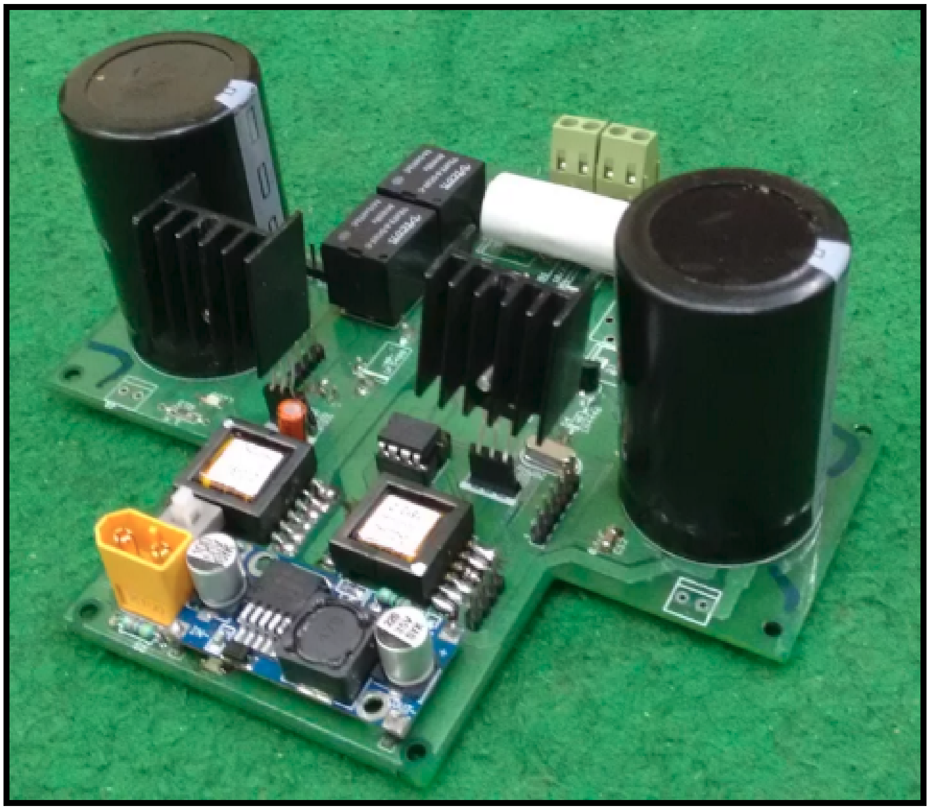


Fig. 4. Final fabricated Kicker circuit

3.5 Kicker Board

The kicker board mainly consists of charging and discharging circuits. The kicker circuit has a dedicated AtMega 328 controller. The kicker board communicates with the main board using external interrupt trigger. Currently, our kicker supports 3 levels of kick namely slow, medium and fast. The kicker can kick the ball at speeds upto 10m/s.

- **Charging Circuit** The charging circuit is based on SMPS (Switch Mode Power Supply) and use DC-DC boost converters. DC-to-DC techniques use transformers that work at much higher frequencies.Power MOSFETs used for switching are controlled by a PWM signal from Atmega with a frequency of 100 Khz. The PWM duty cycle is optimized for fast charging of capacitors.The capacitors can charge upto 150V in 6-7 seconds. The continuous feedback of charged capacitors is monitored by Atmega controller using ADC peripheral. 2 capacitors are connected in series with specifications of 2200uF each with a rating of 250V.
- **Discharging Circuit** In the discharging circuit,the solenoid is connected in parallel to capacitors in series with relays for switching the discharging circuit.The relays are also controlled with atmega controller. Variable discharging of capacitors is also implemented using relays. Two relays have been used in parallel for handling the high discharging current.

4 Software

An overview of Skills, Tactics and Plays (STP) architecture developed on Robot Operating System is presented in subsection 1. Subsection 2 discusses path planning, namely the RRT algorithm used. It also discusses trajectory generation, velocity profiling, and trajectory tracking.

4.1 Software Architecture

Almost all teams building robot soccer software rely on some sort of multi layer framework, that separates lower level logic of the robot (path planning, locomotion) from the higher level logic (goalkeeping, passing, plays). We utilise the Skills, Tactics and Plays (STP) framework developed by the CMU team CM-Dragons.

Our existing STP implementation had several issues, such as:

- **Single project, single executable**: For testing a single part of the machinery, the whole project must be compiled and executed. No scope for unit testing.
- **Deadlocks**: Multithreading has been employed for parallelization. Even though semaphores have been used, deadlocks do occur and executable needs to be restarted.

We have rehauled our architecture and moved over to ROS(Robot Operating System). Earlier, our software was a single package written in C++. It has now been divided into several modules called ROS packages. Each ROS package can be individually downloaded and built, and provides several advantages over the previous implementation.

Modules:

- **skills**: Contains definitions of all skills.
- **tactics**: Contains definitions of all tactics, and definition of a tactic factory.
- **plays**: Contains definitions of all plays, and definition of a play factory.
- **play exec**: Describes node that executes a play and publishes tactic id and params to each ssl robot instance.
- **ssl robot**: Describes node that manages a single robot. Subscribes to belief state and play exec, publishes to comm data
- **robot comm**: Describes node that communicates with either grSim (the simulator) or the real robots, depending on execution parameters. Subscribes to topic comm data and communicates over RF (real robots) or using protobuf (grSim)
- **vision comm**: Describes node that publishes vision info to the topic vision. Connects to either SSL Vision instance (real robots) or grSim instance (simulator) depending on execution parameters.
- **kgpkubs launch**: Contains .launch files and other scripts for launching all nodes on a machine.
- **navigation**: Contains definitions of all planners and controllers used by Skills.
- **belief state**: Describes node that subscribes to vision and publishes BeliefState information on belief state. This information includes robot positions and velocities, ball position and velocity, field parameters, predicates, etc.
- **ssl common**: Contains miscellaneous libraries that are required by several other modules, such as config files, network libraries, serializer/deserializer etc.
- **ssl msgs**: Contains all ROS message definitions required by any other module. Communication on all topics must be through one of these message types..

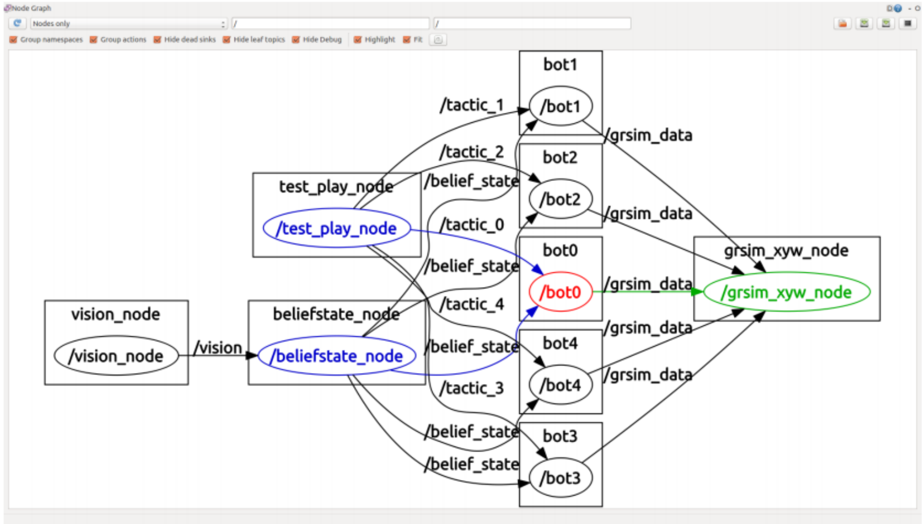
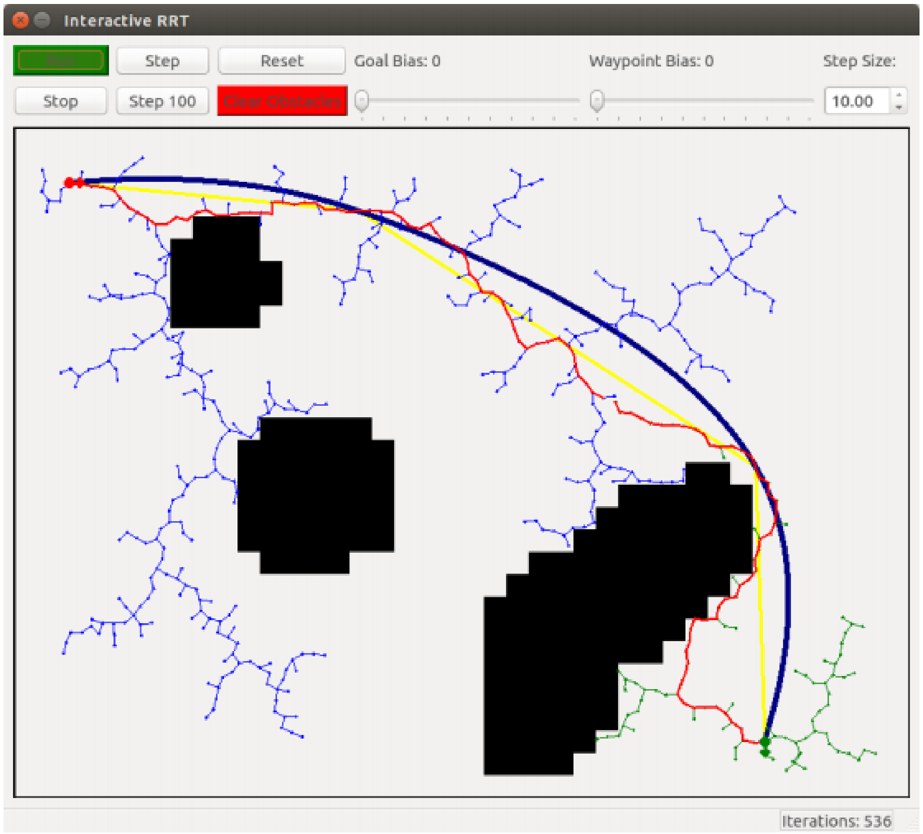


Fig. 5. ROS Nodes

4.2 Path Planning and Trajectory Control

Almost all teams use Rapidly generated Random Trees (RRT) for finding obstacle free paths. In our case, we have adapted an RRT algorithm from team Robojackets of Georgia Tech University to tackle the challenges of obstacle avoidance, time predictability, and adherence to robot kinodynamics. It provides correct, robust trajectories within 3 send-and-receive cycles. The new features introduced by us include getting velocity values at each vertex, minimum time based neighbour search and cubic interpolation between planned points.



Goal Bias: 0

Waypoint Bias: 0

Step Size: 10.00

Iterations: 536

Fig. 6. RRT Applet Screenshot

4.3 Trajectory Generation and Tracking

There exists a variety of motion planning approaches that use splines to represent trajectories. They all concentrate on several aspects of the problem but none accounts for all requirements that we consider important: a path that can be exactly followed by the robot (i.e., curvature continuity), explicit planning for velocities with consideration of kinodynamics, anytime capability, an optimization not prone to local optima, and treatment of unmapped obstacles.

Trajectory Generation We have developed a trajectory generation algorithm over the RRT planned path. In essence, we solve the cubic bezier (for both position and velocities) for the waypoints returned by the RRT generator. We then discretize the bezier by taking only a sample of points in each curve (currently set to 10), so that the result is again a set of time parameterized waypoints, however now much smoother.

Tracking We have simply used a PID tracker independently for each dimension of the configuration space: x, y, and theta. Using a PID tracker over a more complicated algorithm has several advantages:

- Modeling errors (of the linear kind) are inherently handled by PID control
- Oscillations and overshoots can be directly tuned through the Kd and Ki coefficients.

The disadvantage of this algorithm is that all calibration is manual, without the use of any modeling parameters. In the future, we may try to use trajectory linearization trackers which are claimed to perform better.

5 Discussion and Future Works

As future work, it is imperative to explore more dynamics that affect the robot behavior. On the embedded side, we aim to incorporate variable discharging for controlling the speed with which ball is hit; develop fuzzy-PID controller based on encoder readings; integrate IMU for better state estimation. On the software side, we would improve upon the tracking algorithm and upgrade inter-tactic strategies such as pass-recieve.

Acknowledgements

The team also acknowledges the mentorship and guidance of our professors Prof. Jayanta Mukhopadhyay, Prof. Sudeshna Sarkar, Prof. Alok Kanti Deb and Prof. Dilip Kumar Pratihar. This research is supported by Sponsored Research and Industrial Consultancy(SRIC), IIT Kharagpur. We also thank our former team members who made all of this possible.

References

1. Brett Browning, James Bruce, Michael Bowling, and Manuela Veloso. Stp: Skills, tactics, and plays for multi-robot control in adversarial environments. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 219(1):33–52, 2005.
2. Lindsey Langstaff Ryan Strat Robert Woodworth Justin Buchanan, Sean Csukas. Team description paper, robojackets gt, 2015.
3. Tamas Kalmar-Nagy, Raffaello D'Andrea, and Pritam Ganguly. Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle. Robotics and Autonomous Systems, 46(1):47–64, 2004.
4. Michele Aicardi, Giuseppe Casalino, Antonio Bicchi, and Aldo Balestrino. Closed loop steering of unicycle like vehicles via lyapunov techniques. Robotics and Automation Magazine, IEEE, 2(1):27–35, 1995.
5. Werner Dirk Jan Dierssen. Motion planning in a robot soccer system. A Master's Thesis, Language, Knowledge and Interaction Group Department of Computer Science University of Twente The Netherlands, 2003.

6. Marko Lepetic, Gregor Klancar, Igor Skrjanc, Drago Matko, and Bostjan Potocnik. Time optimal path planning considering acceleration limits. Robotics and Autonomous Systems, 45(3):199–210, 2003.
7. Christoph Sprunk and Boris Lau. Planning motion trajectories for mobile robots using splines. University of Freiburg, 2008.
8. Gregor Klancar, Drago Matko, and Saso Blazic. Mobile robot control on a reference path. In Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation, pages 1343–1348. IEEE, 2005.
9. Michael Bowling and Manuela Veloso. Motion control in dynamic multi-robot environments. In Computational Intelligence in Robotics and Automation, 1999. CIRA'99. Proceedings. 1999 IEEE International Symposium on, pages 168–173. IEEE, 1999.
10. Alessandro De Luca, Giuseppe Oriolo, and Marilena Vendittelli. Control of wheeled mobile robots: An experimental overview. In Ramsete, pages 181–226. Springer, 2001.
11. Oussama Khatib. Realtime obstacle avoidance for manipulators and mobile robots. The international journal of robotics research, 5(1):90–98, 1986.