Design and Development of an Omni Wheel Soccer Robot

Musharraf Ahmed Hanif*, Muhammad Sarwar Ehsan*, and Syed Atif Mehdi**
*Department of Electrical Engineering, University of Central Punjab, Lahore, Pakistan.
**Department of Computer Science, University of Central Punjab, Lahore, Pakistan.
musharraf.hanif@ucp.edu.pk, sarwar.ehsan@ucp.edu.pk, syed.atif@ucp.edu.pk

ABSTRACT

This paper presents the design of soccer playing robots developed for the RoboSprint 2015 national competition by the team from University of Central Punjab, Lahore, Pakistan. This competition was envisioned to be a stepping stone to encourage locally developed robots to take part in the international RoboCup matches. Some of the biggest challenges for competing in the international RoboCup was the necessary leap in design capability that was missing in most national competitions. Our main design criteria was to develop robots with good performance without the usual high costs associated with such designs. In this regard, we were able to design and build two robots that had about half the speed and acceleration for less than a quarter of the total cost of the typical designs employed by the leading teams in the RoboCup SSL (Small Size League). Another design choice normally not found in such robots is the implementation of an internal closed loop feedback control in addition to the external closed loop control incorporating the shared vision server and the team server. This internal feedback loop architecture was developed to reduce the responsibility on the team server, leaving it free to form strategy without having to be concerned with the micro-management of the two robots. Test measurements show that these performance goals are achieved with a low-costrobot.

Key Words: Omni Wheel Robot, Soccer Robot, RoboSprint.

1. INTRODUCTION

Robotics is gaining popularity among younger generation, which is quite evident by the tremendous increase in robotic competitions all over the world. Most of these challenges require indigenously developed robots that can perform a particular task in a very controlled environment [1]. Among these, one popular competition is where robots play in a field with a ball and score goals. There are different variants of this Robot Football. In some versions, humanoid robots are used to play against each other [2-3]. In other versions, groups are encouraged to develop their own wheeled robots and these robots play the game [4-5].

The RoboSprint was the most popular robot football competition in Pakistan. It was aimed to promote research in the field of autonomous robotics in Pakistan [6]. Therefore, the competition required teams to have their own indigenous robots to play the game. Its eventual aim was to evolve as a local version of the international SSL of the RoboCup competition. The main goal of the RoboCup is "By the middle of the 21st century, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup" [7]. The last RoboSprint was organized in 2015. The design presented in this paper participated in the 2015 competition and was later updated in anticipation of future events.

A high degree of mobility and controllability is required for wheeled robots to play a soccer game. In many cases it is not feasible that a wheeled robot may take full turn like a car when it is trying to fetch the ball. This will not only increase the turning radius but also cause un-necessary delay in picking the ball. Similarly, the design of the robot should be such that it may hold the ball when it is moving towards the goal. Eventually, when the robot reaches its target location, it may able to shoot the ball with some force so that it may go for a goal. These design, budget and performance constraints led to U-bot – an indigenous soccer robot. U-bot, Fig. 1, is a unique robot in its performance and design. Details of the robot and its capabilities are discussed in detail.

In this paper, we were able to design and build two robots that had about half the speed and acceleration for less than a quarter of the total cost of the typical designs employed by the leading teams in the RoboCup SSL. We also employed an internal closed loop feedback control in addition to the external closed loop control incorporating the shared vision server and the team server. This internal

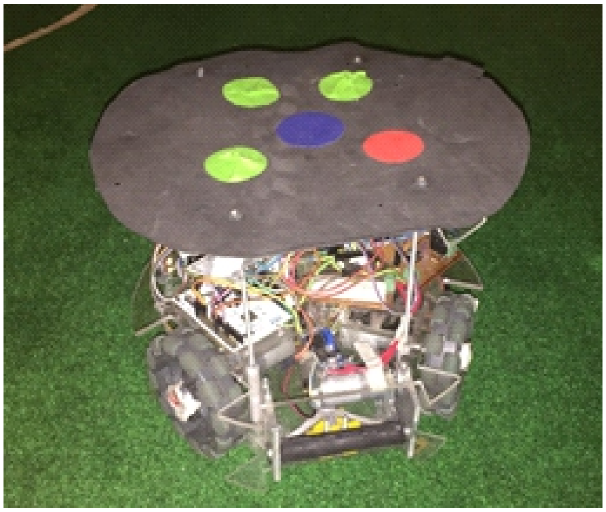

FIG. 1. ONE OF THE ASSEMBLED U-BOT ROBOTS WITHOUT ITS PROTECTIVE SIDEWALL

feedback loop architecture was developed to reduce the responsibility on the team server, leaving it free to form strategy without having to be concerned with the micro-management of the two robots.

The rest of the paper is organized as follows: Section-II, describes an overview of already existing robotic systems that play soccer. Section-III, focuses on design on U-bot. Section-IV describes and discusses performance of U-bot in actual testing. Section-V, concludes the paper and gives an outlook to the future work.

2. LITERATURE REVIEW

RoboCup soccer started in 1997 to foster research in robotics, AI (Artificial Intelligence), Computer Science and Engineering [8].

A very comprehensive view about RoboCup has been [9]. The authors described rules for SSL and team formation. It also described the playing area that is monitored by a roof mounted camera providing whole view of the field to the server. This information is communicated to the robots playing the game and can be used for developing strategies or to intercept the ball in the field.

A unique mechanism for dribbling the soccer ball for small size robot was developed by Ruizand Weitzenfeld [10]. Their catch and dribble strategy provided more control on the ball during movement of the robot. There is still room for improvement and further experiments to make the mechanism more effective.

A mechatronic design for small sized soccer robot has been presented in [11]. They have used omni-directional wheels to move the robot in the environment. These wheels gave advantage to move the robot in any direction without rotation of the robot itself. They have also developed a kicking mechanism alongside dribbling to pass ball or shoot it towards the goal.

The robot developed in [12], focuses on distributed control architecture and a mechanism for communication between robots in the team was developed. It also includes PI (Proportional Integral) controller to control speed of the robot when it is approaching the ball or another robot.

Recognizing team player in the field of SSL is a challenging task. Bruce and Veloso [13], a fast and accurate method based on vision has been described to detect pattern. This pattern can be placed on top of the robots and top mounted camera in SSL field will transmit the images to the server. From the images, the image processing algorithm can be used to identify exact location and orientation of the robot in the field.

A unique omni-directional navigation system, omni-vision system and omni-kick mechanism has been described in [14]. The robots developed in their paper are based on no-head direction (i.e. their direction of movement and the orientation on the field are independent of each other) and respond more quickly in the MSL (Medium Size League) arena. Their robots are capable of more sophisticated behaviors like ball passing or goal keeping.

The RoboSprint 2015 rules were published to enable potential participants to build their system according to common specifications. These rules dictated that each team have two robots in the field. Fig. 2 shows both teams would get access to the locations of the robots and the ball on the field from RoboCup SSLs shared vision system. The rules also specified the dimensions and weights of the robot [15].

Four team description papers [16-19], provide a basis of components and techniques used by some of the leading teams in the SSL competition.

3. ROBOT DESIGN

System Overview: Fig. 3 shows the block diagram of the developed robots. Each robot has three brushed DC motors with gear boxes driving omni wheels. The motors also have integrated quadrature encoders that allow direct measurement of wheel velocities and displacements.

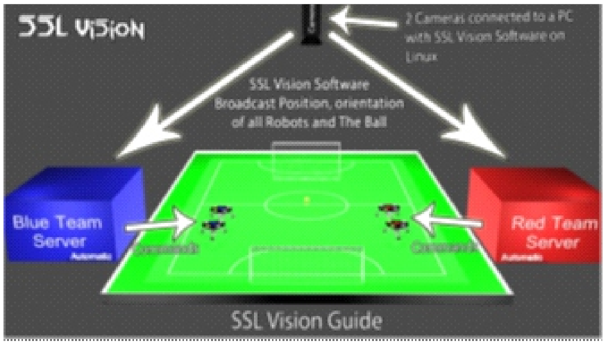

FIG. 2. OVERVIEW OF THE SHARED VISION SYSTEM USED BY THE ROBOCUP SMALL SIZE LEAGUE [15]

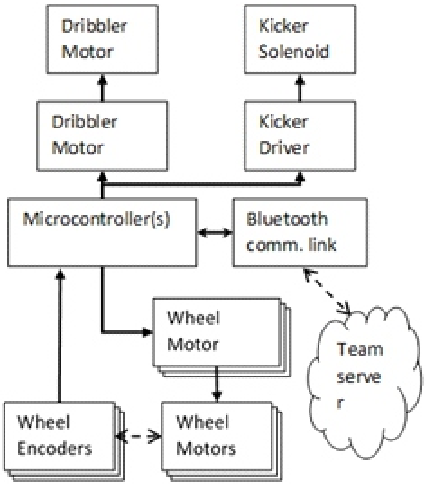

FIG. 3. ROBOT BLOCK DIAGRAM

The kicking mechanism uses a solenoid with a 25 mm travel pusher. The kicker driver allows control of the actuation power applied to the solenoid, which in turn controls the amount of force exerted on the ball.

A Bluetooth transceiver is used to provide a bidirectional link to the team server. This link is used to both receive commands as well as to send the robot's telemetry and status report back to the team server. The link can also be used to tune the gains of the PID (Proportional Integral Derivate) controllers using serial commands and getting the results of the step commands to the system.

Mechanical Layout and Corresponding Orientation Transforms: The developed robot uses three omni wheels arranged 120° apart in a circular pattern as shown in Fig. 4. The front of the robot also houses the ball handling mechanism which comprises of a dribbler and a flat kicker (indicated in Fig. 4 with a hashed rectangle).

With the used omni wheel arrangement, the transform matrices for converting between the robot frame velocities (v_f, v_n and ω_r) and the wheel velocities (v_1, v_2 and v_3) are given by Equations (1-2). These Equations (1-2) allow us to compute the lateral velocities from the measured wheel velocities as well as to compute desired wheel velocities for a known set of translational velocities.

$$\begin{bmatrix} v_f \\ v_n \\ \omega_r \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{3}{3}} & -\sqrt{\frac{3}{3}} & 0 \\ \frac{1}{3} & \frac{1}{3} & -\frac{2}{3} \\ \frac{1}{3R} & \frac{1}{3R} & \frac{1}{3R} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \qquad (1)$$

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{3}{2}} & \frac{1}{2} & R \\ -\sqrt{\frac{3}{2}} & \frac{1}{2} & R \\ 0 & -1 & R \end{bmatrix} \begin{bmatrix} v_f \\ v_n \\ \omega_r \end{bmatrix} \qquad (2)$$

where v_1, v_2 and v_3 correspond to the velocities of the three wheels, v_f is the forward velocity of the robot, v_n is the lateral velocity of the robot and ω_r is the rotational velocity of the robot frame and R is the distance from the centre of the robot to the wheels.

Similarly, transform matrices are required for converting between the absolute frame of reference (i.e. the match area) and robot frame of reference (i.e. robot chassis). Fig. 5 shows a robot at an angle θ_r to the x axis and with its center at coordinates x_r and y_r. Equations (3-4) provide the relationship for converting between the world frame coordinates and velocities and the robot frame velocities.

$$\begin{bmatrix} v_x \\ v_y \\ \omega_w \end{bmatrix} = \begin{bmatrix} \cos\theta_r & -\sin\theta_r & 0 \\ \sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_f \\ v_n \\ \omega_r \end{bmatrix} \qquad (3)$$

$$\begin{bmatrix} v_f \\ v_n \\ \omega_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & \sin\theta_r & 0 \\ -\sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_w \end{bmatrix} \qquad (4)$$

where v_x and v_y are the components of the robot velocity along the x and y axis of the absolute frame, ω_w correspond to the rotational velocity of the robot in the context of the world frame, x_r and y_r are the coordinates of the location of the robot on the absolute frame and θ_r is angle of the robot chases to the absolute frame.

Position Determination: SSL Vision and Deadreckoning: The SSL vision system calculates the position of all the robots and the ball on the field with an error of less than 5mm. Special markers placed on top the robots allows the system to identify the different robots on the field and their orientations. These are then passed to both the team servers using UDP (User Datagram Protocol) multicast packets. In our testing in the lab, it is

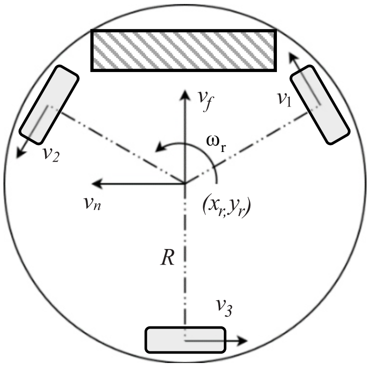

FIG. 4. LAYOUT OF THE ROBOT PLATFORM, THE VELOCITY VECTORS FOR THE THREE WHEELS AND THE ROBOT FRAME VELOCITIES

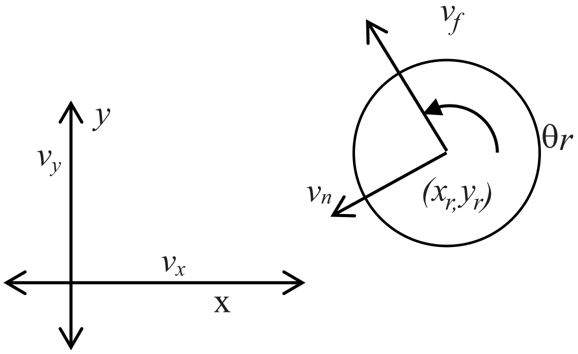

FIG. 5. ABSOLUTEFRAME AND ROBOT FRAME VELOCITIES

observed that due to some combination of the specified low-cost cameras and various other factors that we were unable to determine conclusively, there is a delay of up to 1s in positions received from the server and the actual positions on the field.

The world frame velocities can also be calculated from measured wheel velocities using Equations (1-2). These world frame velocities can then be used to calculate the change in the position and orientation of the robot on the playing field independently of the shared vision system. This approach provides a rapidly updated output. However, due to limitations of the sensors, gearbox backlash and slight wheel slip during hard accelerations, the error in the calculated position tends to compound quickly (an error of up to 100 mm can be reached within 10 seconds if aggressive maneuvering is being performed).

In order to fuse the position data from the two sources, a new algorithm has been devised and implemented. It overcomes the latency of the precise location and the poor accuracy of the responsive mechanism by maintaining a log of all the computed positions for the last 1 second. When an updated precise position is received, it calculates the correction factor for that moment in time (using past values) and applies it to all reading from that time forward. This technique offers much better performance than what is possible by either of the techniques acting alone.

Control Hierarchy: The overall control system comprises of a total of six PID controllers shown in Fig. 6. The velocity of each wheel is controlled with a PID controller. The output of the controller is used to control the direction and speed of each wheel. For the feedback of wheel speed, the angular displacement and speed of each wheel is calculated from the input of the corresponding wheel encoder. The gains of these controllers are set using online tuning through the use of Bluetooth telemetry. This online tuning mode makes it possible to change controller gains and then command step responses without having to recompile and reprogram the controller.

The reference inputs for the lower level wheel speed controllers are obtained from three higher level controllers, each of which is responsible for one of the three parameters (i.e. the position in x and y coordinates and the orientation angle). The outputs of these position controllers are then transformed to desired robot wheel velocities. The feedback for the position controller is taken from the position fusion algorithm described above.

Telemetry: Command and Status Packets: The robot expects a fixed length command packet from the team server. The minimum time between two commands is limited by only the transmission limit of the Bluetooth communication channel. In practice, team server could send as many as 60 commands per second. In the absence of a new command, the robot completes the execution of the last command and then holds the last commanded position. Any new command received before the completion of the previous command result in the command being performed being overwritten. This is intentional as the situation can be rapidly evolving during a match and might require a change in the game plan where completing the previous commands might no longer be beneficial.

Table 1 show the format of the command packets. There were three major parts of the packet. The first part is reserved for the commanded position and orientation. The second part is reserved for actual position and

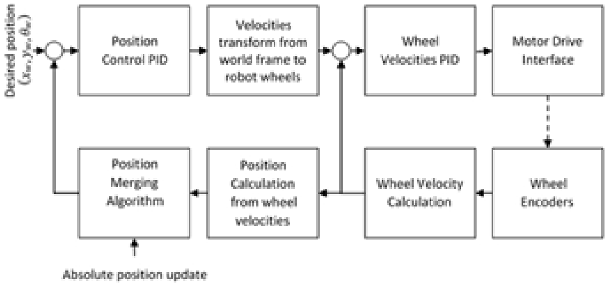

FIG. 6. COMPLETE CONTROL LOOP

TABLE 1. FORMAT OF THE COMMAND PACKETS RECEIVED FROM TEAM SERVER

Bytes	Contents	Description of Contents
1	'$'	Header
5	x	Commanded position and orientation in mm and degrees (with sign)
5	y	
4	θ	
5	x	Observed position and orientation from the vision system in mm and degrees. *t* is the age of position in this segment represented in ms.
5	y	
4	θ	
3	t	
3	Commands, force and speed	Kicker force, kicker commands and dribbler speed command
1	Status	Packet status (indicates which fields are valid)
1	';'	Terminator

orientation received from the shared vision system and also includes an age of the information based on the timestamps on the UDP packets. The third part is reserved for the ball handling commands and covers the kick strategy, kick force as well as the speed for the ball dribbler. Finally, the status indicates which of these three fields contain valid data. It is possible to send out commands where only one, two or all three fields contain valid data.

Table 2 shows the format of the status packets sent to the team server. Like the command packets above, these also had three fields. These are used to report back the last valid command received, the internally calculated position and to acknowledge the commands to the ball included consisted of the feedback from the ball detection sensor. There is no packet status as all fields always contained valid information regardless of the configuration of the last received command. The robot sends status packets every 20 milliseconds.

Overall System Integration: The overall system integration is achieved by implementing cooperative tasks and by using a cooperative time triggered task scheduler. Fig. 7 shows the complete data flow diagram of the robot. Following is a brief description of the various processes or tasks.

The Wheel Speed Measurement task periodically calculates the individual wheel speeds from the wheel encoders. These wheel speeds are then used by the Position Dead reckoning task to get the responsive but error prone position and orientation of the robot. This data along with the desired wheel speed data is also used by the

TABLE 2. FORMAT OF THE STATUS PACKETS SENT TO TEAM SERVER

Bytes	Contents	Description of Contents
1	‘#’	Header
5	x	Last commanded position and orientation in mm and degrees (with sign)
5	y	
4	θ	
5	x	Current internally calculated position and orientation from the vision system in mm and degrees.
5	y	
4	θ	
3	Commands, force, speed and possession	Last commanded kicker force, kicker commands and dribbler speed command
1	‘;’	Terminator

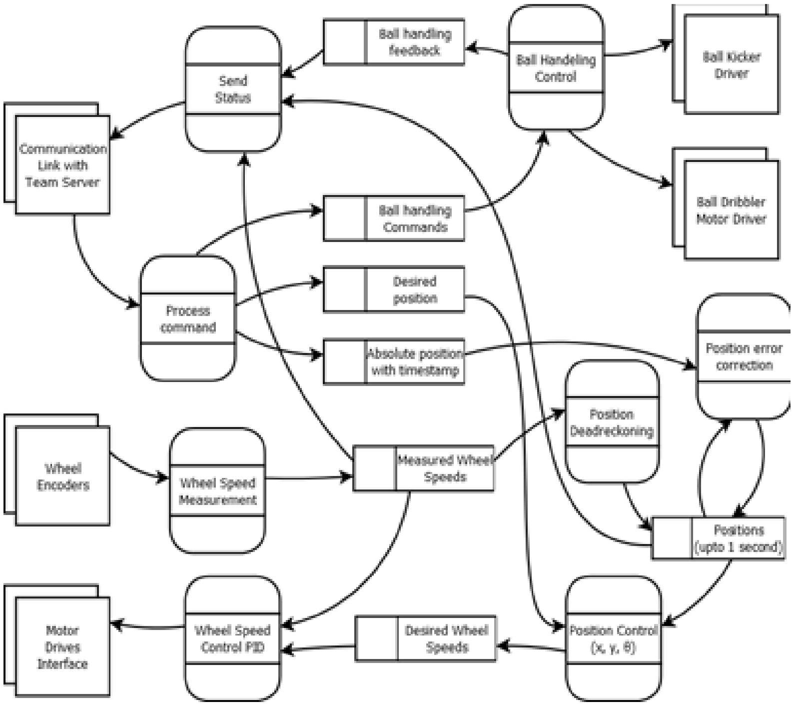

FIG. 7. DATA FLOW DIAGRAM FOR THE ROBOT

Wheel Speed Control PID task to control the power to the wheel motors.

The Process Command task gets data received from the team server and stores it into variables used by the ball handling control task, position error correction task and the position control task. The Send Status task sends the feedback from the ball handling control, as well as the internally measured wheel speeds and calculated position of the robot back to the team server.

The Position Error Correction task uses implements the position fusion algorithm described in the Position Determination subsection. The fused position information is used by the Position Control task to figure out the desired wheel speeds to bring the robot into the desired position and orientation received from the team server.

The Ball Handling Control task is responsible for executing the relevant commands received from the team server. This could be to capture the ball by inducing a backspin using the dribbler. The amount of backspin is controlled via commands. Similarly, the force with which to kick the ball as well as the moment to kick is also controlled via commands.

The main control loop repeats every 20 milliseconds. This time period was selected based on the responsiveness of the open loop system. The total CPU utilization when running at 72MHz is under 20%. The tasks are staggered so as not to overload any time tick of the cooperative scheduler. An added benefit of the staggering is that the task launch jitter in all the tasks used for sampling data and for updating the outputs is minimized. The sequence of tasks in the control loop is as follows:

(1) Wheel Speed Measurement

(2) Position Deadreckoning

(3) Process Command

(4) Position Error Correction

(5) Position Control

(6) Wheel Speed Control

(7) Ball Handling Control

(8) Send Status

Robot Hardware Cost Comparison: One of the stated goals of the project was to develop the robots at a significantly lower cost than those used by the major teams in RoboCup. The design utilized by a lot of the teams (e.g. SKUBA, CMDragons, etc.)used either four 30 Watt Maxon flat brushless DC motors to drive the four wheels through custom built gear boxes or larger 50 Watt motors in a direct drive configuration [16-19]. These translate to around US$150for the motor, gear box and encoder per wheel, thus, a total of US$600 for only drive mechanism of a single robot. By contrast, our design uses a total of three geared permanent magnet brushed DC motors with built in encoders costing US$40 or US$120 in total for one robot. Similarly, the elimination of chip kicker from the usual designs also helped to reduce the cost of the ball handling mechanism. The downside is that our design is too big to enter RoboCup SSL matches but is sufficient for the requirements for the RoboSprint competition. The other tradeoff is the performance, where our robot has about two thirds of the top speed and half of the acceleration in some of the top teams of the time [16-18].The developed robots can be seen in action during a tournament match in Fig. 8.

4. PERFORMANCE

The dynamic performance of the robots on the field was measured by using the SSL vision system. The lag of the vision system was then compensated for during the processing of test data. To access the performance of the developed robots, short and long displacement move commands were given to the robot. For each type of command, the robot was put in at starting point and was ordered to move a predetermined distance at orientation angles of 0, 45, 90 and 180° for the robot with regards to the direction of commanded motion. The time taken by the robot to move to withing 50mm of the destination was measured. The maximum time taken to reach the destination over twenty tries was noted.

For multiple small move commands of 0.5m displacement in any direction while keeping the orientation (i.e. θ_r) constant, the robot reached 50mm of the destination within 1.7s, with error in the angle under ±3°.

For multiple larger move commands specifying a destination with a displacement of 4m from the starting point in any direction while keeping the orientation constant, the robot reached 50mm of the destination within 4.2s, with error in the angle under ±5°.

It should be noted that a wait of an additional 1s allowed the robot to reorient and reposition more correctly by utilizing the accurate position information from the SSL vision system. This reorientation was not dependent on any additional commands from the team server but relied on continuous position updates from the vision system.

The enhanced electronic drive for flat kicker was able to kick the ball with a speed of 2.8m/s. This was a significant improvement over the initial system that was employed in RoboSprint 2015 as it was only able to manage 1.4 m/s.

FIG. 8. YELLOW TEAM COMPOSED OF TWO U-BOTS (LEFT HALF OF THE PICTURE) DURING A MATCH IN ROBOSPRINT 2015

5. CONCLUSION

This paper presents a detailed design of a small size robot used in a RoboCup style competition in Pakistan called RoboSprint. The design philosophy as well as the “Design and Development of an Omni Wheel Soccer Robot”robots developed due to it are presented along the real-time measurements in the playing field. These measurements show the successful working of the developed design. The total cost for each of the developed robot is less than the quarter of the cost of drive mechanism in some of the leading team robots in the RoboCup SSL.

6. FUTURE WORK

For future work, the ball handling mechanism will be enhanced by developing a chip kicker in addition to the existing flat kicker. Moreover, an onboard computer vision system will be added to enhance and extend the systems capability in close quarters situations during a match. Finally, torque control instead of speed control for the wheel motors should be implemented.

ACKNOWLEDGEMENT

Authors would like to thank the Faculty of Engineering, and Faculty of Information Technology, University of Central Punjab, Lahore, Pakistan, for their support, feedback and suggestions.

REFERENCES

[1] Mitchell, O., "What Has Twenty Years Of RoboCup Taught Us?," Robot Rabbi, 21 May 2017. [Online]. Available: https://robotrabbi.com/2017/05/21/robocup/. [Accessed 23 October 2018].

[2] RoboCup.org, "RoboCupSoccer - Humanoid," [Online]. Available: https://www.robocup.org/leagues/3. [Accessed 25November 2019].

[3] RoboCup.org, "RoboCupSoccer - Standard Platform," [Online]. Available: https://www.robocup.org/leagues/5. [Accessed 25 November 2019].

[4] RoboCup.org, "RoboCupSoccer - Small Size," [Online]. Available: https://www.robocup.org/leagues/7. [Accessed 25 November 2019].

[5] RoboCup.org, "RoboCupSoccer - Middle Size," [Online]. Available: https://www.robocup.org/leagues/6. [Accessed 25 November 2019].

[6] Aftab, H., "CASE Robosprint: Fostering Love for Robotics and Technology," Pro Pakistani, 26 March 2015. [Online]. Available: https://propakistani.pk/2015/03/26/case-robosprint-fostering-love-for-robotics-and-technology/. [Accessed 23 October 2018].

[7] RoboCup.org, "Objective - RoboCup.org," [Online]. Available: https://www.robocup.org/objective. [Accessed 23 October 2018].

[8] Veloso, M., and Stone, P., "Video: RoboCup Robot Soccer History 1997 – 2011",IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 2012.

[9] Weitzenfeld, A., Biswas, J.,Akar, M., and Sukvichai, K., "RoboCup Small-Size League: Past, Present and Future",RoboCup 2014: Robot World Cup XVIII, Springer, 2015.

[10] Ruiz, M.S., and Weitzenfeld, A., "Soccer Dribbler Design for the Eagle Knights RoboCup Small Size Robot",IEEE 3rd Latin American Robotics Symposium, Santiago, Chile, 2006.

[11] Witte, J.D., "Mechatronic Design of a Soccer Robot for the Small-Size League of RoboCup", Vrije Universiteit Brussel, Brussel, 2010.

[12] Smit,A., "Development of a Robot for RoboCup Small Size League, Utilizing a Distributed Control Architecture for a Multi-Robot System Development Platform", Stellenbosch University, Stellenbosch, South Africa, 2011.

[13] Bruce, J., and Veloso, M., "Fast and Accurate Vision-Based Pattern Detection and Identification", IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003.

[14] Samani, H.A., Abdollahi, A., Ostadi, H., and Rad, S.Z., "Design and Development of a Comprehensive Omni directional Soccer Player Robot",International Journal of Advanced Robotic Systems, Volume 1, No. 3, pp. 191-200, 2004.

[15] CASE and PAF-KIET, "Robo Sprint 2015 Rulebook," CASE and PAF-KIET, 2015. [Online]. Available: https://www.scribd.com/document/346184177/Robo-Sprint-2015-Rulebook. [Accessed 23 October 2018].

[16] Panyapiang, T., Chaiso, K., Sukvichai, K., and Lertariyasakchai, P., “Skuba 2013 Team Description”, RoboCup SSL Team Description 2013, Available: https://ssl.robocup.org/wp-content/uploads/2019/01 /2013_TDP_Skuba.pdf [Accessed 25 November 2019]

[17] Biswas, J., Mendoza, J.P., Zhu, D., Etling, P.A., Klee, S., Choi, B., Licitra, M., and Veloso, M., "CMDragons 2013 Team Description", RoboCup SSL Team Description 2013, Available: https://ssl.robocup.org/wp-content/uploads/2019/01/2013_TDP_CMDragons.pdf [Accessed 25 November 2019]

[18] Almagro, J., Avidano, C., Lindbeck, C., Neiger, J., Olkin, Z., Peterson, E., Stachowicz, K., Stuckey, W., White, M., Woodward, M., and Burdel, G.P., “RoboJackets 2019 Team Description Paper”, RoboCup SSL Team Description, Available: https://ssl.robocup.org/ wp-content/uploads/2019/03/2019_ TDP_RoboJackets.pdf [Accessed 5 February 2020]

[19] Ito, M., Suzuki, R., Isokawa, S., Du, J., Suzuki, R., Nakayama, M., Ando, Y., Umeda, Y., Ono, Y., Kashiwamori, F., Kishi, F., Ban, K., Yamada, T., Adachi, Y., and Naruse. T., "RoboDragons 2019 Extended Team Description", RoboCup SSL Team Description”, 2019, Available: https://ssl.robocup.org/wp-content/uploads/ 2019/03/2019_ETDP_RoboDragons.pdf [Accessed 5 February 2020]