Control and Self-Localization of an Omni-Directional Mobile Robot

S. ZIAEI-RAD[1], F. JANABI-SHARIFI[2], M. DANESHPANAH[1], A. ABDOLLAHI[1], H. OSTADI[1], and H. SAMANI[3]

(1) Mechanical & Electrical Engineering Departments
Isfahan University of Technology, Isfahan,
IRAN

(2) Mechanical and Industrial Engineering
Ryerson University, Toronto
CANADA

(3) Computer Science Department
Oxford University, Oxford
UNITED KINGDOM

Abstract: - Omni-directional mobile robots are becoming popular for many applications especially in soccer playing robots. Effective control and self-localization of omni-directional mobile robots constitute important and challenging issues. In this work, a simplified model of the system has been derived for fast tuning of the control system parameters. In particular, strategies for fast tuning of PID/PD coefficients for position and orientation control are devised. A vision-based self-localization and the conventional odometry systems have been fused for robust self-localization. The methods have been tested in the RoboCup competition field using three Persia middle size omni-directional robots.

Key-Words: - Omni-directional, Mobile robot, Control, Self-localization.

1 Introduction

Among many suggested motion mechanisms such as universal wheel, ball wheel, crawler and offset steered wheel, and omni-directional wheel [1-4], omni-directional wheels can provide high mobility with no motion restriction. In practice, providing high speed with an acceptable error is very important factor for success in a competitive and dynamic environment such as RoboCup competitions (Fig. 1). An omni-directional robot can reach to any position with no rotation through a straight line. For this purpose, *fast* yet *robust* and reliable self-localization and control approaches must be adopted. Additionally, in the context of novice operation (such as in the student's competition contest), or time-pressured situations, the system must be *simple* to develop and tune.

Despite many works related to self-localization of robots [5-12], the problem is still open. Common methods of dead-reckoning [5] are prone to errors that are accumulated over time. Therefore, it is necessary to combine other methods such as triangulation landmarks or map matching, in order to probabilistically update robot localization. The problem is usually formulated with a likelihood function over all possible positions of the robot and a measure is used to find a probabilistic match between local and global maps [10-12]. However, these approaches are usually complicated and time-consuming. Reliability and robustness of many of these approaches are also questionable for robotic

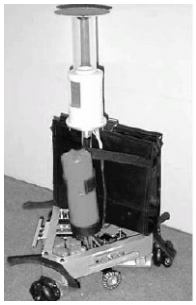


Fig. 1. Persia omni-directional soccer player robot.

soccer competitions [7, 13]. This paper contributes by proposing a simple, efficient, and reliable hybrid self-localization method using a fused system of odometry and vision feedbacks. Each of the feedbacks used have their own advantages and limitations. Odometry provides ease and low cost of implementation and computation, but is limited by the slippage effect and accumulation of odometry errors. Vision-based self-localization ensures flow of rich information unaffected by the slippage effect, yet limited by the camera occlusion and camera calibration errors (of extrinsic and intrinsic parameters). Also, image processing techniques might be time-consuming. The hybrid odometry system is proposed to compensate for disadvantages of both methods. In particular, localization errors, e.g., the slippage effects of driving wheels, will not dominate the self-localization results. Additional contribution of this work includes the sensitivity analysis of the performance of a vision self-localization and feedback system. The objective was to obtain sensitivity of the localization method to visual noise. The results showed that using one method for all points in the field was not perfect. Hence utilizing other landmarks in the field was proposed.

From control perspective, advanced control techniques have been proposed for omni-directional robots, with many being computationally inefficient, or impractical, or difficult to tune, and/or implement [1, 14-16]. Among many control techniques, Proportional-Integral-Derivative (PID) control remains outstanding due to its simplicity, robustness, effectiveness, a wide range of applicability, and near-optimal performance [17]. Therefore, PID strategy was adopted for the position control in this work. This paper also contributes by proposing a simple strategy for fast yet effective tuning of a PID control. The orientation control was achieved using PD control law. It is a time consuming process to set the PID controllers coefficients manually with no prior estimation and based on just trials and errors. On the other hand, solving the set of coupled differential equations is very complicated and may not be practical for a real time control [14]. Some teams decoupled the mathematical model of the system while the others used fault tolerant control strategy for their systems [15]. Real-time path generation based on the polynomial spline-interpolation with prediction of velocities of spline functions was also proposed and used [16]. A fuzzy model of the omni-directional robot control was studied analytically in [1]. However, these approaches had problems such as lengthy effort for control tuning, complicated mathematical models for a real-time trajectory generation, and/or use of a single feedback system for control structure. Also, some of these models offered only theoretical but impractical solutions. This paper also contributes by outlining practical considerations for implementing and realizing a pose control through integrating PID and PD control laws for position and orientation control, respectively.

By combining the proposed strategies and utilizing the comprehensive omni-directional robot [18], Persia Middle Size team won the 1st place in World RoboCup Technical Challenge Competitions in Portugal 2004 and the 3rd place in Italy 2003.

This paper is organized as follow. Section 2 describes the robot kinematics. The control strategy and the feedback generation for position control are represented in sections 3 and 4, respectively. The experimental results are explained in section 5. Finally, section 6 concludes the paper.

2 Robot Kinematics

Omni-directional robots usually use omni-directional poly-roller wheels. The most common wheel consists of six spindles like rollers that can freely rotate about their rotation axes [1, 19]. Therefore a robot with three omni-directional wheels can follow any planar trajectory. Three active omni-directional wheels (for motion system) and three small passive wheels with shaft encoders (as a feedback mechanism) were used in the experimental robot (Fig. 2). The schematic view of robot kinematics with omni-directional wheels is shown in Fig. 3. From the kinematics model of the robot [14], one can derive the vector of the coordinates of the wheels centers with respect to a local coordinate frame ($\boldsymbol{P}_w$) and drive directions as:

$$\boldsymbol{P}_w = \begin{bmatrix} P_{w1}^T \\ P_{w2}^T \\ P_{w3}^T \end{bmatrix} = L \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}, \tag{1}$$

$$\begin{bmatrix} \boldsymbol{D}_{w1}^T \\ \boldsymbol{D}_{w2}^T \\ \boldsymbol{D}_{w3}^T \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix}. \tag{2}$$

where L is the distance of wheels center from the

robot center of gravity (O), and vector $\boldsymbol{D}_{wi}$ is the drive direction of the i-th motor. The vector of linear velocities of the wheels ($V_i(t)$, $i=1,2,3$) can be written as:

$$\boldsymbol{V} = \dot{\boldsymbol{P}}_w + \dot{\boldsymbol{R}}(\theta)\boldsymbol{P}_o, \quad (3)$$

where $\boldsymbol{R}(\theta)$ is the rotation matrix. Then it can be readily shown that the wheels angular velocity vector, $[\dot{\varphi}_1, \dot{\varphi}_2, \dot{\varphi}_3]^T$, can be written as a function of linear and angular velocities of the robot (i.e., $[\dot{x}, \dot{y}, \dot{\theta}]^T$):

$$\begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \end{bmatrix} = \frac{1}{r}\begin{bmatrix} -\sin\theta & \cos\theta & L \\ -\sin(\frac{\pi}{3}-\theta) & -\cos(\frac{\pi}{3}-\theta) & L \\ \sin(\frac{\pi}{3}+\theta) & -\cos(\frac{\pi}{3}+\theta) & L \end{bmatrix}\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (4)$$

where r is the major radius of wheels. Linear and angular momentum equations for the robot can be formulated as:

$$\sum_{i=1}^{3} F_i \boldsymbol{R}(\theta)\boldsymbol{D}_{wi} = m\ddot{\boldsymbol{p}}_o, \; L\sum_{i=1}^{3} F_i = J\ddot{\theta}, \quad (5)$$

where $\ddot{\boldsymbol{p}}_o = [\ddot{x}, \ddot{y}]^T$ is the linear acceleration vector of the center of mass with respect to Cartesian coordinate frame, F_i is the magnitude of the force produced by the i-th motor, m is the mass of the robot, and J is its moment of inertia about its center of gravity. Assuming no-slip condition, the force generated by a DC motor can be written as:

$$\boldsymbol{F} = \alpha\boldsymbol{U} - \beta\boldsymbol{V}, \quad (6)$$

where $\boldsymbol{U} = \{U_i(t), i=1,2,3\}$ is the voltage applied by a supplier to the DC motors The constants α and β are motor characteristic coefficients and can be determined either from experiments or from the motors catalogue. Substituting (6) into (5) yields:

$$\sum_{i=1}^{3}(\alpha U_i - \beta V_i)\boldsymbol{R}(\theta)\boldsymbol{D}_{wi} = m\ddot{\boldsymbol{p}}_o, \quad (7a)$$

$$L\sum_{i=1}^{3}(\alpha U_i - \beta V_i) = J\ddot{\theta}, \quad (7b)$$

Fig. 2. Omni-directional chassis.

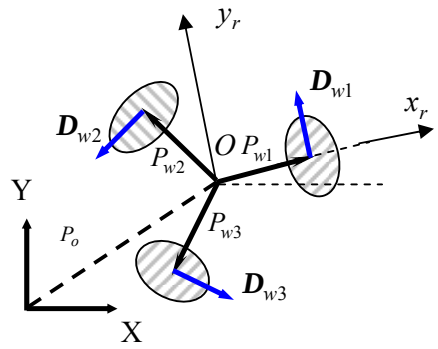


Fig. 3. Robot kinematic diagram with local and global coordinate frames.

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ J\ddot{\theta} \end{bmatrix} = \alpha\boldsymbol{P}(\theta)\boldsymbol{U} - \frac{3\beta}{2}\begin{bmatrix} \dot{x} \\ \dot{y} \\ 2L^2\dot{\theta} \end{bmatrix}, \quad (8)$$

$$\boldsymbol{P}(\theta) = \begin{pmatrix} -\sin\theta & -\sin(\frac{\pi}{3}-\theta) & \sin(\frac{\pi}{3}+\theta) \\ \cos\theta & -\cos(\frac{\pi}{3}-\theta) & -\cos(\frac{\pi}{3}+\theta) \\ L & L & L \end{pmatrix}. \quad (9)$$

3 Robot Controller

In this work, PID and PD controllers were used for controlling the robot pose (position and orientation). The experiments showed that such system was robust enough for controlling a soccer player robot [15]. For obtaining the PID controller gains, one needs to obtain first the whole transfer functions of the system and then use it for initial tuning. Determining overall equations governing the system behavior is not straightforward. Since the equations are a set of coupled nonlinear differential equations, it is very difficult to solve them in a time-efficient fashion. Even if one manages to solve the equations, the resultant PID gains will not be reliable because they will depend on many other parameters such as ground surface friction factor, characteristics of batteries and so on. For many robotic competitions, an efficient and fast tuning method is desired. Therefore, the equations need to be decoupled with the use of the following assumptions:

(1) Omni-directional mechanism is a mechanism which can reach to any position with no rotation (i.e., without loss of generality, one can assume $\theta = 0$) through a straight line. This prescription would help the robot to reach the desired position in the shorter time than that with a 2-wheel

mechanism. It can be also assumed that any curve could be approximated by dividing it into straight line segments and at the end of each segment, the robot would not need to rotate to follow the next segment.

(2) Whenever it is necessary to rotate (e.g., when the kicker robot needs to be in a particular position), the robot rotates while it is moving in a straight line to reach the target position. This can be regarded as a pure rotation in addition to the first assumption. The pure rotation in our robot is obtained by applying equal voltages to each motor.

(3) In order to find PID coefficients for the robot position controller, moving through a straight line is very similar to moving through an axis like X-axis (i.e., y = 0 in (8)). The voltage obtained from position controller is then added to the voltage found by orientation controller.

Based on the above assumptions, the robot position does not depend on θ. Therefore, for position control, one would assume that $\theta = 0$. In the cases where rotation is required, the voltage obtained from orientation control for each motor is equally added to the position controller output. For PID tuning in position controller, a simple movement was considered, i.e., θ = 0, y = 0 (or a constant value) in (8). Similarly, for orientation control, a pure rotation is considered, i.e., $x = 0$ (or constant), and $y = 0$ (or constant).

3.1 Position Control Structure

Fig. 4 shows the overall block diagram of the system. As it is shown in Fig. 4, the omni-directional robot control loop contains a PID controller (with the transfer function H_{PID}) and a PD control law, a plant transfer function (H_{P} which is obtained from the system dynamics), and a self-localization transfer function (as a feedback function that only senses the robot's position). A noise node, N, is also included that has an additive effect on the system position input. The input of the system is considered to be a step function and the output is the robot position and orientation. Experiments showed that this type of controller is robust enough for controlling a soccer player robot [14].

Two simple motions were considered and solved, namely straight-line motion of the robot, e.g., along X direction and pure rotation about the Z-axis. The former means that one motor is turned off and the other two are turned on with the same but opposite angular velocity while the latter means

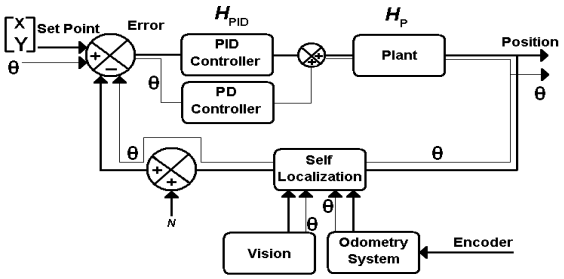


Fig. 4. Control diagram of the omni-directional robot.

that all three motors are turning with the same angular velocities.

The orientation will be studied separately in section 3.2. The output voltage from the orientation controller (w) is then added to the voltage obtained from the position controller output (v_i). The assumption of summing up these voltages is valid while motors are operating in their linear regions. In order to apply the straight line motion, one can consider (8) with:

$$\theta = 0,\ \dot{\varphi}_1 = \dot{y} = \dot{\theta} = \ddot{\theta} = 0,\ \ \dot{\varphi}_2 = -\dot{\varphi}_3 .$$

Equation (8) then reduces to:

$$m\ddot{x} + 3\frac{\beta \dot{x}}{2} = \sqrt{3}\alpha U_2 . \qquad (10)$$

Applying Laplace transform to (10) with the initial conditions: $X(0) = 0, \dot{X}(0) = 0$, one obtains:

$$H_P(s) = \frac{X(s)}{U_2(s)} = \frac{\sqrt{3}\alpha}{ms^2 + \frac{3\beta s}{2}} . \qquad (11)$$

It should be noted that for ideal case (in the absence of noise), the complete transfer function for position control would be obtained as follows (assuming $H_{Self\ Localization} = 1$):

$$H_{Total}(s) = \frac{H_{\mathrm{PID}} H_{\mathrm{P}}}{1 + H_{\mathrm{PID}} H_{\mathrm{P}}} = \frac{\sqrt{3}\alpha (K_D s^2 + K_P s + K_I)}{m s^3 + (\frac{3\beta}{2} + \sqrt{3}\alpha K_D) s^2 + \sqrt{3}\alpha K_p s + \sqrt{3}\alpha K_I} \qquad (12)$$

Here K_P, K_I, and K_D are proportional, integral and derivate gains, respectively. Fig. 5 shows the step and noise response curves with various K_P, K_I, and K_D values. The following observations can be deducted. The dotted line in Fig. 5 shows a step function with an additive white (zero-mean) Gaussian noise (AWGN). In this curve, the noise was applied to the system every 40 microseconds due to the robot processing time. As observed from Fig. 5, by increasing K_P and K_I (dash-dotted line

and solid line), the system settling time would increase. Also, there are some overshoots in these curves. However, by increasing the K_D value, this effect reduces drastically. In order to find optimum values for the PID gains, different combinations of the parameters were selected and examined. Eventually, the proper PID gains were obtained for the proposed system as $K_P = 1,\ K_I = 1,$ and $K_D = 10$. The response of the system for these values is depicted by thick solid line in Fig. 5.

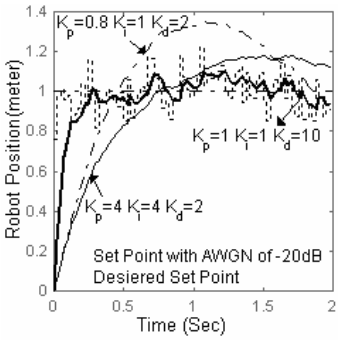


Fig. 5. System step response of position control with different values of PID gains.

3.2 Orientation Control

Suppose that the robot only rotates about its vertical axis, i.e., Z-axis. Thus: $\dot{\varphi}_1 = \dot{\varphi}_2 = \dot{\varphi}_3,\ U_1 = U_2 = U_3$. Substituting these values into the third equation in (8) leads to:

$$J\ddot{\theta} + 3\alpha\beta L^2\dot{\theta} = 3LU_3 . \qquad (13)$$

Applying Laplace transform to the above equation yields

$$\theta(s)/U_3(s) = 3L/(Js^2 + 3\alpha\beta L^2 s), \qquad (14)$$

and considering a PD controller for this case, the total transfer function for orientation control is given as:

$$H_{Total}(s) = \frac{3L(K_P + K_D s)}{Js^2 + (3\alpha\beta L^2 + 3LK_D)s + 3LK_P} . \qquad (15)$$

Fig. 6 shows the step response of the control system. Experiments showed that the level of noise (measured by noise/signal ratio) in orientation controller was considerably less than that in the position controller (almost 3 times). Therefore, the noise was ignored in tuning PD control gains (Fig. 6). Since the experience showed that residual error for orientation control is not of great importance in the given scenario (i.e., robotic soccer competitions), a PD controller will result in desired system response. Therefore, there was no need to apply PID controller for the orientation control. The optimum parameters for PD gains were obtained as $K_P = 100,$ and $K_D = 10$. The step response for these parameters values is shown by a solid line in Fig. 6. The slight overshoot is desirable since the effect of friction (damping the response in our model) was ignored.

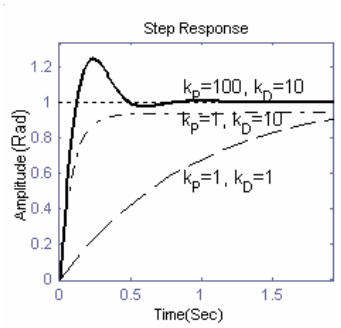


Fig. 6. Step response of orientation control for different values of PD gains.

3.3 Overall Robot Controller

In order to implement the position controller, the position error vector is determined as follows:

$$\boldsymbol{e} = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x' \\ y' \end{bmatrix}, \qquad (16)$$

while the vectors $[x\ y]^T$ and $[x'\ y']^T$ are the desired and the actual position of robot in the field, respectively. Thus, the position control output can be written as:

$$\boldsymbol{V}_m = K_P\boldsymbol{e} + K_I\int \boldsymbol{e}dt + K_D\frac{d\boldsymbol{e}}{dt}, \qquad (17)$$

where $\boldsymbol{V}_m$ expresses the output vector of the position controller for the driving units whose components on each driving wheel (V_{mi}) are extracted from:

$$V_{mi} = \boldsymbol{V}_m{}^T \cdot \boldsymbol{D}_{wi} . \qquad (18)$$

For orientation control (using PD law), the orientation error can be calculated using the desired and current head angles of the robot, namely Δ and δ, respectively, as follows:

$$e_\Delta = \Delta - \delta . \qquad (19)$$

The orientation controller output will be then:

$$w = K_P e_\Delta + K_D\frac{de_\Delta}{dt} . \qquad (20)$$

The voltage from the orientation controller output will then be added to the voltage obtained from the position control output. Next, the final applicable voltages will be computed as:

$$U_i = v_i + w . \qquad (21)$$

This voltage is applied to each motor to reach the desired point. Since the system sensitive parts such as electronic board, computer, batteries, etc., may be

damaged by rapid rotation of the robot, one needs to apply upper and lower cut-off thresholds for the orientation controller output. Practically, the threshold was set to be ± 10 v. The PID and PD gains were obtained from the two previous cases, and used as first estimation, leaving only fine-tuning to the scene. This was due to the robot working conditions such as friction, and gear boxes clearances and tolerances that were not available in advance and thus not considered in initial modeling. The proper coefficients were then fine-tuned experimentally during each competition. The results showed that for real cases, the maximum changes in the calculated values were bound to ±10% of the original gains values. Therefore, such simplification proved to provide good initial approximation, considerably simplifying final gains tuning.

4 Position Feedback

The position control method, described in the former sections, calls for some form of position feedback. The performance of this feedback depends on its reliability, accuracy and real-time computability. There have been plenty of algorithms and methods proposed by different researchers in the literature [5-12]. Among them self-localization by visual information and odometry approach are dominant due to their special characteristics which will be discussed in the following paragraphs.

In this work, a compound novel method was developed and optimized for RoboCup Middle Size League in which both visual and odometry information were used to ameliorate a real time, accurate and reliable method. Although optimized for soccer player robots, the self-localization method proposed here has enough modularity and flexibility to be applicable in many robotic applications involving self-localization.

Each of these complementary methods (vision/odometry self-localization) operates autonomously and has its own advantages and drawbacks in providing position feedback for robot control. For example, odometry method is known to have memory-based operation, accumulative error, low jitter, simplicity of implementation, cheap hardware, etc. On the other hand, vision-based self-localization algorithms often provide memory-less implementations (despite memory-based ones), no error accumulation, high jitter, relatively high computational complexity, and expensive hardware. Amalgamating these methods can present good performance in vast and diverse conditions. Each of these methods and their fusion are explained in the coming subsections.

4.1. Vision-Based Self-Localization

Vision module was designed with several goals in mind, including obtaining spatial information of ball, opponents, and teammates. Robot platforms were equipped with omni-directional cameras [6], with which the projection of the whole field area was available to the camera with a hyper-parabolic mirror (See [18] for more details and Fig. 1) with the following parabolic profile:

$$y^2/1135.7 - x^2/233.3 = 1, \quad (22)$$

where x and y are given in mm. Since the omni-directional mirror introduces a map with very high non-linearity between pixel separation in the scene and the real physical distance (of such pixels) in the field itself, it is not reliable enough to develop algorithms that use distances as their input data. Instead angles are preserved completely in a linear manner if the center of mirror and camera are aligned perfectly. Therefore, the algorithms with angles as their input data are more reliable and can perform more efficiently. The proposed approach in vision-based self-localization is based on arcs. In basic geometry, there is a fact that having an angle of observation ω to a fixed and spatially known object in a 2D plane, can provide one with possible loci of the observation points. Actually, the points are located on the circumference of two circles (C_1, C_2). This simple idea is illustrated graphically in Fig. 7.

The proposed algorithm here employs three different observation angles to constrain the unique position of observer (robot) in the field (assuming the ideal case with no visual noise). A good set of observation angles should have the following properties: (i) availability from different locations in the field; (ii) extractability from visual data with low computational effort; (iii) independency of the arcs resulting from these angles which means that the resulting arcs should leave no location ambiguity at any point in the filed; and (iv) lower sensitivity to visual noise with the increase of the angles magnitude.

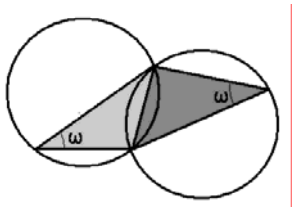


Fig. 7. Angle of observation ω and the two related arcs.

Since goals are fixed landmarks and at least one of them has reasonable observation angle within the whole field, their use for self-localization is popular in RoboCup Middle Size League [8]. An insightful examination through different combinations of possible observation angles for this purpose revealed that the following three angles are suitable regarding the above characteristics:
(i) The observation angle from the robot itself to the nearest goal (α_{Goal}). (ii) Angle between the center of the farthest goal and left side of the nearest one (β_{Goal}). (iii) Angle between the center of the farthest goal and right side of the nearest one (γ_{Goal}). These angles are depicted for an arbitrary location of a robot in Fig. 8. Assume that the intersection points between Arc(j), and Arc(k) to be defined as:

$$P_i^{j,k} \quad \begin{matrix} j,k \in \{1,1',2,2',3,3'\}; j \neq k \\ i \in \{1,2\} \end{matrix} \tag{23}$$

where the superscripts denote intersecting arcs and a subscript denotes the index of intersection. Note that the robot position is always at a point located on Arc(1).

First, a list of intersection points pairs are prepared using (23). In order to find the exact location of the robot, the Euclidian distances of different pairs of intersections are computed and the one that has zero norm is selected as the answer. In other words, there is only one point that is located on the intersection of three arcs and this point is the real position of the robot in ideal case (i.e., with no noise).

$$\min_{i,j,s,t} \left\| P_s^{1,i} - P_t^{1,j} \right\|_2 \quad \begin{matrix} i,j = 2,2',3,3' \\ s,t = 1,2,3,... \end{matrix} \qquad i \neq j \tag{24}$$

Considering imperfections in visual information extraction, the intersection of Arc(1) with other two arcs may not coincide. In such a case, the set that yields the minimum Euclidean distance introduces the possible position of the robot. The final position is simply computed by averaging over the neighboring intersection points that satisfy the above criterion (Fig. 9).

4.2. Sensitivity Analysis

The performance of vision-based self-localization method, developed in this work, relies on accurate visual information obtained from the vision module by means of image processing techniques. Since goals are of two distinct colors in the play field (Yellow and Blue), the pixels representing them are distinguished by their position in RGB color space. Thus, the position and angle of observation are extracted with special region growing algorithms. As mentioned before, although the angles are preserved linearly in the omni-directional filed-of-view projected by the hyperbolic mirror, there is always the possibility that some error would exist in the detection procedure. The sensitivity analysis of vision-based self-localization method reveals the regions in which the method is most sensitive to visual noise. The sensitivity of some performance characteristic y regarding parameter x_i, is defined as the measure of its change Δy, resulting from a change Δx_i in the parameter x_i. Suppose:

$$y = y(x_1, x_2, ..., x_n) \tag{25}$$

The variation of y is defined as:

$$dy = y \sum_{i=1}^{n} \left[\frac{x_i}{y} \frac{\partial y}{\partial x_i} \right] \frac{dx_i}{x_i} = y \sum_{i=1}^{n} S_{x_i}^{y} \frac{dx_i}{x_i}, \tag{26}$$

where $S_{x_i}^{y}$ denotes the sensitivity of y with respect to parameter x_i, and is computed as:

$$S_{x_i}^{y} = \frac{x_i}{y} \frac{\partial y}{\partial x_i}. \tag{27}$$

Applying the above analysis on the proposed self-localization method showed that in certain areas near the corner posts, the accuracy and reliability of the method degraded drastically (Fig. 10). Therefore, the proposed algorithm may be prone to severe errors in those regions. Since there are flags

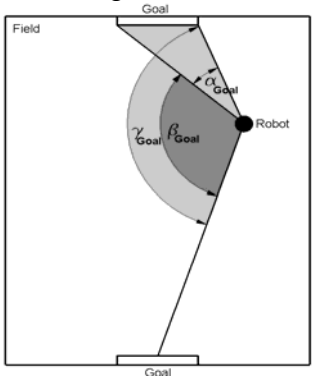


Fig. 8. Angles observed by the robot.

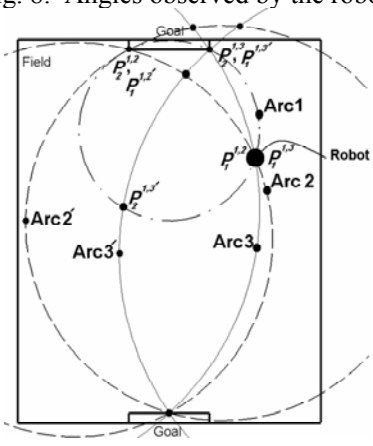


Fig. 9. The arcs and possible intersections.

in the corner posts (providing good visibility and detectibly in that region), these landmarks are proper candidates for self-localization in those regions.

4.3. Localization Using Flags

For achieving better performance in the regions in which the sensitivity of the vision-based self-localization method is high, flags are used instead of goals to determine the position of robot. The procedure can be summarized as follows.

- By using visual data of goals and previous location of robot from its memory, the location of robot is roughly determined as *Front-Left, Front-Right, Back-Left, Back-Right*, where *Front* and *Back* show opponent and own side fields respectively.
- The nearest flag is then detected and the distance of robot to the flag base is approximated by a non-linear map constructed experimentally.
- Since the exact position of flag ($[X_{FLAG}, Y_{FLAG}]^T$) is known and the relative position of robot with respect to the flag is also available (R), then calculating the final robot position (Fig. 11) is a trivial task, i.e.:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X_{FLAG} \\ Y_{FLAG} \end{bmatrix} + \begin{bmatrix} R\cos\varphi \\ R\sin\varphi \end{bmatrix}. \quad (28)$$

Since the method of localization changes in those regions, and in order to avoid potential hystersis and confusion between the two presented methods, a hystersis strip (the grey area between two arcs near the flag in Fig. 11) is defined. Therefore, once a robot crosses the inner ring, the method is switched to use flags, until the robot moves out of the outer ring in the hystersis strip.

4.4. Self-Localization Using Odometry

As it can be seen in Fig. 3, three free rotating omni-directional wheels are placed 60 degrees apart from the main driving wheels. These wheels are only passive, attached to three independent shaft encoders, and have the role of odometry wheels. The shaft encoders data could be used to extract

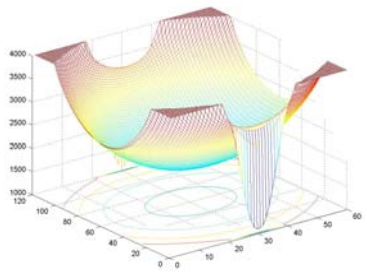


Fig. 10. Sensitivity of vision-based self-localization method at different points.

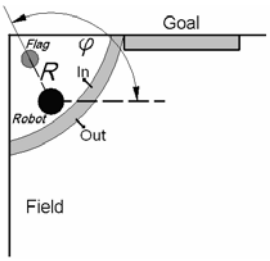


Fig. 11. The schematic view of the robot and flag near the corners (the grey strip is where the hystersis occurs).

pose of the robot [18] as follows

$$x = r\int \frac{1}{3\sin(\frac{\pi}{3})}\Big[\ (\cos(\frac{\pi}{3}+\theta)-\cos(\frac{\pi}{3}-\theta))\dot{\varphi}_1 + (-\cos(\theta)-\cos(\frac{\pi}{3}+\theta))\dot{\varphi}_2 + (\cos(\theta)+\cos(\frac{\pi}{3}-\theta))\dot{\varphi}_3\ \Big]dt,$$

$$y = r\int \frac{1}{3\sin(\frac{\pi}{3})}\Big[(\sin(\frac{\pi}{3}-\theta)-\sin(\frac{\pi}{3}+\theta))\dot{\varphi}_1 + (\sin(\theta)-\sin(\frac{\pi}{3}-\theta))\dot{\varphi}_3\ \Big]dt,$$

$$\theta = r\int \frac{\dot{\varphi}_1+\dot{\varphi}_2+\dot{\varphi}_3}{3L}dt\ , \quad (29)$$

where $[x \quad y \quad \theta]^T$ is a vector containing the position and orientation of the robot. Further simplification of the third equation in (29) results in:

$$\theta = \frac{r}{3L}\Big[\int\dot{\varphi}_1\,dt + \int\dot{\varphi}_2\,dt + \int\dot{\varphi}_3\,dt\Big] = \frac{r}{3L}(\varphi_1+\varphi_2+\varphi_3). \quad (30)$$

4.5. Fused Position Estimator

In order to obtain the final position estimation for the robot, both visual and odometry outputs must be fused in an appropriate fashion that would take advantage of each method to make flaws from the other one ineffective. For example, due to the inherent nature of vision-based self-localization, there is undesired jitter at its output, but, in return, odometry self-localization has smooth changes that can be used as a low-pass filter for vision-based self-localization results. Having this in mind, the following procedure is proposed for estimating the final position:

Step 1: Vision-based self-localization is used to estimate the current position of the robot based on visual information from the current frame.

Step 2: The last computed position is utilized by odometry and the new position is determined through (28).

Step 3: The position of robot is then computed as a weighted average of odometry and vision-based self-localization as:

$$\overline{P} = \eta P_{Odometry} + (1-\eta)P_{Vision}\ , \quad (31)$$

where η represents a fusion parameter that was determined experimentally to be 0.9 for this application. Use of η coefficient resulted in smoothing the variation (due to jitter in vision-based self-localization) of the final position estimation. The coefficient in (31) was obtained by conducting experiments on different robot positions.

Step 4: The initial position for odometry in step 2 is then set to the computed robot position in step 3 and the calculation continues for the next frame.

Since the outputs of both odometry and vision-based self-localization are prone to errors, and due to inherent random nature of these errors, a 2D AWGN is added to the output of a perfect self-localization block in the feedback path, as shown in Fig. 4. The noise can be formulated as:

$$n_g(x,y)=\frac{1}{2\pi\sigma_x\sigma_y}\exp(-(\frac{x^2}{2\sigma_x^2}+\frac{y^2}{2\sigma_y^2})), \qquad (32)$$

where σ_x and σ_y are noise standard deviations in X and Y directions, respectively. These values are then added to the position obtained from the self-localization module, (x_0, y_0), to obtain the probabilistic location of the robot, i.e., (x, y) as:

$$P(x,y)=\frac{1}{2\pi\sigma_x\sigma_y}\exp(-(\frac{(x-x_0)^2}{2\sigma_x^2}+\frac{(y-y_0)^2}{2\sigma_y^2})) \quad . (33)$$

5 Experiments

In order to evaluate the performance of the proposed position controller and self-localization error, four experiments were designed. First, a PID position control was applied. The robot was tracked on a straight line of 1 m length near the center of the field with no rotation. Second, the PD orientation control was employed with just rotation about the *Z*-axis of the robot. Third, the robot was programmed to follow a sinusoidal curve ("A" in Fig. 12.a) with the wave-length of 5 m and amplitude of 3.5 m near the center of the field. Finally, the robot pursued two sinusoidal curves similar to curve A, but far from the center of the field ("B" and "C" in Fig. 12.a).

In the first experiment, the PID constants were set as those calculated in section 3.1. The maximum deviation from the straight-line tracking and the final position error were measured to be 8 cm and 4 cm, respectively (Fig. 12.b). In the second experiment, again the PD controller parameters were set to the calculated values for orientation control (in section 3.2). The maximum error from the set point angle was 0.03π radians. These two experiments showed that the final error for both tracking and pure rotation were in an acceptable range and the PID/PD controller parameters were selected properly.

In the third experiment, the robot had to track the sinusoidal curve ("A" in Fig. 12) while rotating about its Z-axis. The measured errors were between 10 cm and 12 cm, and occurred at points 4, 10, 13, and 17 in curve "A" (Fig. 13). The maximum deviation was measured to be around 12 cm that occurred in point 4. In the last experiment, the curves were located near the edges of the field ("B", "C" in Fig. 12). The maximum deviation between the real and desirable path was measured to be around 23 cm that is less than 7% for this case study.

6 Conclusion

In soccer playing robotic competitions, control and self-localization of robots need to be simple, time-efficient, and reliable. Obtaining a transfer function for an omni-directional system is very complicated and also requires tuning by a trail-and-error procedure that may take long time in practice. In this study, PID and PD controllers were used for position and orientation controls, respectively. Therefore a simplified model of an omni-directional robotic system was developed for tuning the PID and PD coefficients of the robots' position and orientation control, respectively. Then, the controller parameters were set using a simplified model by taking into account the effect of noise. The adopted strategy proved its effectiveness in robotic competitions.

In order to reduce positioning error, a hybrid self-localization method with fused odometry and vision-based localization was proposed. Using the geometrical properties of circles, the exact position of the robot in the field was determined. Next, a sensitivity analysis was conducted to determine the inaccurate points in the field. For those points, the flags were used as landmarks in the corners to overcome such difficulty. The resultant techniques were developed and tested on the real field. The test results showed that typical asymmetric errors for omni-directional mobile robots were reduced drastically on those areas. The improvement of performance was more than 80% in position and orientation in comparison with the case of using only the purely odometric localization.

References:

[1] K. Watanabe, Control of an omni directional mobile robot, *Proc. Second Int. Conf. on Knowledge-Base Intelligent Electronic*

Systems, L.C. Jain and R.K. Jain, Eds., Adelaide, Australia, April 1998, pp. 51–60.

[2] H. Kitano, J. Siekmann, and J. G. Carbonell, *RoboCup 97: Robot Soccer World Cup I, Lecture Notes in Artificial Intelligence* (Nagoya, Japan, Springer-Verlag, 1998).

[3] M. West, and H. Asada, Design a holonomic omni-directional vehicle, *Proc. of 1992 IEEE Int. Conf. on Robotics and Automation,* Nice, France, May 1992, pp. 97–103.

[4] E. Nakano, and N. Koyachi, An advanced mechanism of the omni-directional vehicle (ODV) and its application to the working wheel chair for the disabled, *Proc. of 93 Int. Conf. Advanced Robotics*, 1993, Tokyo, Japan, pp. 277–284.

[5] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, Mobile robot positioning: Sensors and techniques, *J. Robot. Syst.*, Vol. 14, No. 4, 1997, pp. 231–249.

[6] R. Talluri and J. K. Aggarwal, Position estimation techniques for an autonomous mobile robot—A review, in C. H. Chen, L. F. Pau, and P. S. P. Wang (Eds.), *Handbook of Pattern Recognition and Computer Vision*, (Singapore: World Scientific, 1993), pp. 769–801.

[7] C. F. Olson, Probabilistic self-localization for mobile robots, *IEEE Trans. on Robotics and Automation,* Vol. 16, No. 1, 2000, pp. 55–66.

[8] A. Stroupe, K. Sikorski, and T. Balch, Constraint-based landmark localization, *Proc. RoboCup 2002: Robot Soccer World Cup IV,* Japan, 2002, pp. 239–245.

[9] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: Sensors and techniques," *J. Robot. Syst*., Vol. 14, Aug. 1997, pp. 231–249.

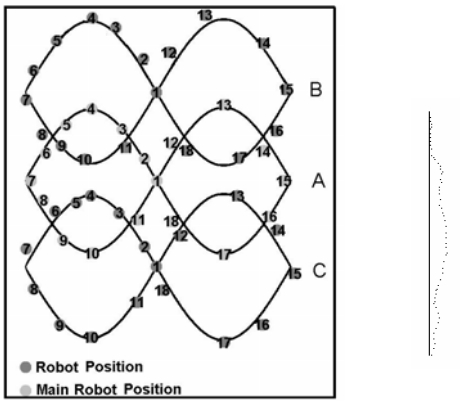
[10] R. Simmons and S. Koenig, Probabilistic navigation in partially observable environments, in *Proc. Int. Joint Conf. Artificial Intell*., Vol. 2, 1995, pp. 1660–1667.

[11] D. Fox, W. Burgard, and S. Thrun, Active Markov localization for mobile robots, *Robot. Auton. Syst*., Vol. 25, No. 3-4, Nov. 1998, pp. 195–207.

[12] S. Thrun,W. Burgard, and D. Fox, A probabilistic approach to concurrent mapping and localization for mobile robots, *Mach. Learning*, Vol. 31, No. 1–3, 1998, pp. 29–53.

[13] A. Martinelli, V. Nguyen, N. Tomatis, and R. Siegwart, A relative map approach to SLAM based on shift and rotation invariants, *Robotics and Autonomous Systems*, Vol. 55, No. 1, Jan. 2007, pp. 50–61.

[14] T. Kalmar-Nagy, P. Ganguly, and R. D'Andea,



(a) (b)

Fig. 12. (a) A, B and C depict the robot trajectories. The numbers show each robot position on each curve. (b) The straight line followed by the robot.

Fig. 13. Notations "A", "B", "C" are the robots that followed the corresponding "A", "B", and "C" curves in Fig. 12.

Real-time trajectory generation for omni – directional vehicles, *Proc. of the American Control Conf.*, Anchorage, AK, USA, May 2002, pp. 285–291.

[15] M. Jung, and J.H. Kim, Fault tolerant control strategy for OmniKity-III, *Proc. 2001 IEEE Int. Conf. on Robotics Automation,* Seoul, Korea, May 2001, pp. 3370–3375.

[16] I. Paromatchik, and U. Rembold, A practical Approach to motion generation and control omni-directional mobile robot, *Proc. IEEE Int. Conf. on Robotic and Automation*, San Diego, CA, 1994, pp. 2790–2795.

[17] P. Cominos and N. Munro, PID controllers: recent tuning methods and design to specification, *IEE Proc. on Control Theory and Applications*, Vol. 149, No. 1, Jan. 2002, pp. 46–53.

[18] H. Samani, A. Abdollahi, H. Ostadi, and S. Ziaie Rad, Design and development of a comprehensive omni directional soccer player robot, *Int. J. Advanced Robotic Systems*, Vol. 1, No. 3, 2004, pp. 191–200.

[19] H. Asama, M. Sato, L. Bogoni, H. Kaetsu, A. Matsumoto, and I. Endo, Development of an omni directional mobile robot with 3 DOF decoupling drive mechanism, *Proc. IEEE Int. Conf. on Robotics and Automation,* Nagoya, Aichi, Japan, 1995, pp. 1925–1930.