2014 11th IEEE International
Conference on Control & Automation (ICCA)
June 18-20, 2014. Taichung, Taiwan

Speed and Point-to-Point Control for Trajectory Tracking of a Skid-Steered Mobile Robot

Marissa G. Campa[1], J.L. Gordillo[1], and Rogelio Soto[1]

Abstract— **This paper presents the development of a motion control scheme of a Skid-Steered Mobile Robot (SSMR). The System Architecture used in this work consists in four systems: Trajectory Tracking, Point-to-Point Control, Speed Control and Pose Estimation. These four systems work together to perform autonomous navigation. The Speed Control System consists in a digitally designed R-S-T control, for both the left and right side of the mobile robot. This configuration allows to obtain similar transient responses, ensuring the achieving of the reference speed. Using wheel encoders and an IMU, the Pose Estimation System obtains the pose of the robot, based on the kinematics. The Point-to-Point and the Trajectory Tracking System provide the robot with the necessary speeds each side must execute, in order to reach the desired coordinated points. Experimental results on a real world robotic platform show the reliability of the method to perform trajectory tracking.**

I. Introduction

Mobile robots (MR) have become a focus of interest because of their variety of applications, such as search-and-rescue, space exploration, industrial warehouse navigation, human-aid service, among others. In autonomous navigation, a MR is required to translate to a specific place, without external assistance from an user.

Four-wheel drive MR, known as Skid-Steered Mobile Robots (SSMR), have robust mechanisms for outdoor usage, because of the power provided by four motors. These robots are been used in many tasks [1] [2]. In SSMR, the left and right side wheels need to skid laterally in order to move in a curved path. These configuration adds difficulty to both the trajectory tracking and the estimation of the robot's pose.

Over the years, research in the area of autonomous navigation has been conducted in order to improve control methods for wheeled MR. An autonomous navigation system for an unmanned MR was presented in [3], which allows the MR to travel through an urban area, using information from its sensors. A motion control scheme for a MR was developed in [4], taking into account the mathematical model of the robot, in order to perform trajectory tracking. The work presented in [5] illustrates robust motion control of a MR, considering speed control and path tracking. In [6], an identification of the dynamic behavior of the robot is performed, in order to obtain a various mathematical models and then be used for autonomous navigation of a MR.

A control system based on a R-S-T type controller is proposed as a solution to the speed control of a SSMR. The R-S-T controller uses the discrete model of the plant. A desired behaviour is specified through a second order model. With these two models, a control law can be obtained. With this scheme, the dynamic behavior of the left and right side of the robot can be specified, reducing the double tuning iterations of a conventional PID. Furthermore, the trajectory tracking problem is approached from the standpoint of control, were the inertial characteristics of the MR are implied in the tuning parameters of a PID control system.

This paper is organized as follows: Section II describes the System Architecture used in this work, as well as the kinematics of a SSMR. The development of the digitally designed R-S-T control system is explained in Section III. Section IV shows experimental results of the Speed Control System and the Trajectory Tracking for autonomous navigation. Finally, conclusions are presented in Section V.

II. Autonomous Mobile Robot Development

A. Kinematics of a Skid-Steered Mobile Robot

A Skid-Steered Mobile Robot (SSMR) is a differential drive type of robot, because the movement of the robot is caused by the difference between left and right sides speeds. To move in a straight line, the left and right speeds need to be equal. When the right side speeds is larger than the left side speed, the resulting movement is turning in a CCW direction, the same for the opposite case, in which turning of the robot in a CW direction is caused by the left side speed been larger than the right side speed

The kinematics of a SSMR is presented in [8]. In this model, the speeds of the robot in a local coordinate frame can be used to estimate its pose in a global coordinate frame. The state vector of generalized velocities is :

$$\dot{\boldsymbol{q}} = [\ \dot{X}\ \ \dot{Y}\ \ \dot{\theta}\ ]^T \in R^3 \tag{1}$$

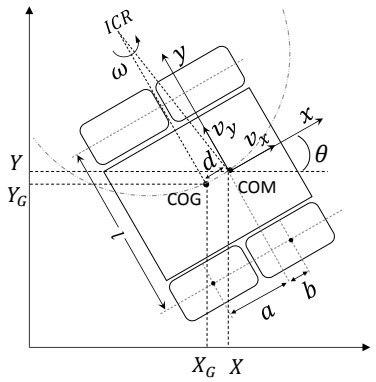

Fig. 1: Kinematics of a SSMR

[1]Center for Robotics and Intelligent Systems (CRIS), Tecnológico de Monterrey, Monterrey 64849, Mexico;
Corresponding author marissag.campa@gmail.com

The transformation between the local velocities defined in the local coordinate framed attached on the vehicle's center of mass (COM) and the global frame velocities is [9] [8](see Fig. 1):

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{2}$$

A kinematic constraint introduced by [9] is that when the SSMR is moving, there is no lateral skid. This constraint can be expressed as follows: $v_y - d\omega = 0$; $v_y = d\omega$, where d is the distance between the center of mass (COM) and the center of geometry (COG). The last equation is non-integrable, for it describes a non-holonomic constraint, which can be rewritten as:

$$\begin{bmatrix} -\sin\theta & \cos\theta & -d \end{bmatrix} \begin{bmatrix} \dot{X} & \dot{Y} & \dot{\theta} \end{bmatrix}^T = \boldsymbol{A}(\boldsymbol{q})\,\dot{\boldsymbol{q}} = 0. \tag{3}$$

We can rewrite (2) in the form below

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix}, \tag{4}$$

that can be also expressed as: $\dot{\boldsymbol{q}} = \boldsymbol{S}(\boldsymbol{q})\,\eta$, where

$$\boldsymbol{S}(\boldsymbol{q}) = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix}, \qquad \eta = \begin{bmatrix} v_x \\ \omega \end{bmatrix}$$

Since the columns of $\boldsymbol{S}(\boldsymbol{q})$ are always in the null space of $\boldsymbol{A}$ the following expression is satisfied.

$$\boldsymbol{S}(\boldsymbol{q})^T \boldsymbol{A}(\boldsymbol{q})^T = 0 \tag{5}$$

As we can see in (4), $\dot{\boldsymbol{q}}$ can be obtained in the global reference frame. However, these components are with respect to the center of mass of the robot. The transformation between the inertial frame and the geometrical frame is shown in (6):

$$\begin{bmatrix} X_G \\ Y_G \end{bmatrix} = \begin{bmatrix} X - d\cos\theta \\ Y - d\sin\theta \end{bmatrix} \tag{6}$$

B. System Architecture

A general diagram of the System Architecture used for autonomous navigation is depicted in Fig. 2. The blocks with gray shade show the systems that are implemented in this paper. The System Architecture is composed of four systems: Trajectory Tracking, Point-to-Point Control, Speed Control and Pose Estimation.

C. Trajectory Tracking System

The Trajectory Tracking System (TTS) supervises the arrival of the robot to each point in the discrete trajectory. It receives a 2D-array containing the set of points in the $X-Y$ plane that constitute the discrete trajectory for the robot to travel, along with a *margin error*. And the system returns one by one the reference point (X_r, Y_r) to be assigned to the Point-to-Point Control System.

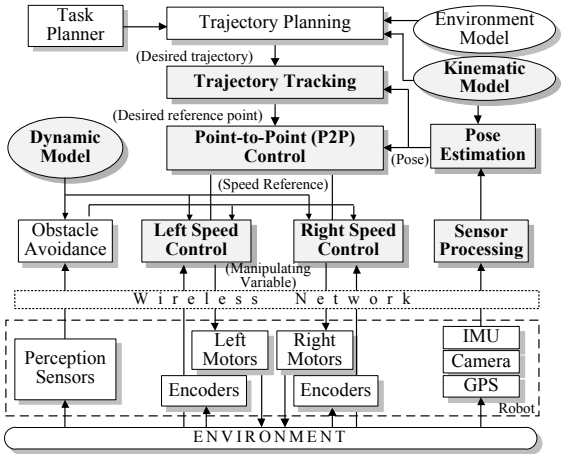

Fig. 2: System Architecture Diagram

A discrete trajectory 2D-array defined by $\mathbf{Q}$ $(n\,\mathrm{x}\,2)$, is shown in (7)

$$\mathbf{Q}_i = [\, X_i, Y_i \,;\, X_{i+1}, Y_{i+1} \,;\, \ldots \,;\, X_{n-1}, Y_{n-1} \,;\, X_n, Y_n \,] \tag{7}$$

where n is the total number of reference points in the $X-Y$ plane, that the robot must reach.

D. Point-to-Point Control System

The Point-to-Point Control System (P2P) is responsible for allowing the robot to reach a specific coordinate point on the $X-Y$ plane. The P2P receives a reference point (X_r, Y_r) and a desired velocity v_r, and it returns left and right speed commands that will be sent as reference to the Speed Control System.

The P2P can be invoked as a function:

$$ref\_speed = \mathrm{P2P\_control}\,(ref\_point)$$

where *ref_point* is an array containing the reference point (X_r, Y_r) and *ref_speed* is an array containing the left and right reference speeds $[v_L, v_R]$ that the Speed Control System will receive.

When the *P2P_control()* function is called, it consults the current pose $[X, Y, \theta]$, and it compares it with the reference point $[X_r, Y_r]$, as can be seen in the Fig. 3. The P2P obtains

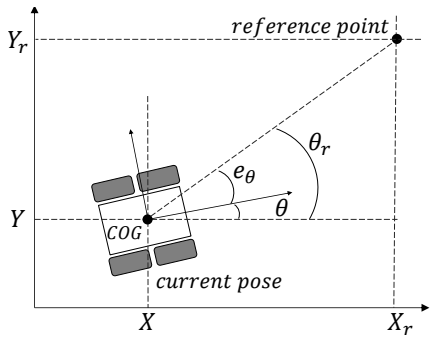

Fig. 3: Angle error in the global coordinate

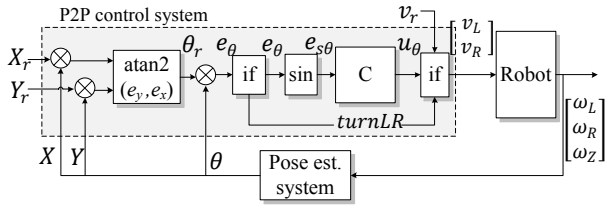

Fig. 4: Detailed diagram of the P2P Control System

the distance errors as follow:

$$e_X = X_r - X,\ e_Y = Y_r - Y,\ \theta_r = \text{atan2}\left(\frac{e_Y}{e_X}\right),\ e_\theta = \theta_r - \theta\ .$$

where e_X and e_Y are the distance errors in the X and Y axis respectively, θ_r is the auxiliary angle and e_θ is the angle error between the reference point and the current pose of the robot.

Once the angle error (e_θ) is calculated, the sense of rotation is calculated as in [10], defining if the robot will turn left or counterclockwise (CCW) or if it will turn right or clockwise (CW).

A PID controller is implemented for the P2P, symbolized as the control block "C" in Fig. 4. The sine function $e_{s\theta} = \sin(e_\theta)$ is used as in [4], and it is used as input to the P2P.

The output signal u_θ is scaled from the interval $[0\ \ 100]$ to the interval $[0\ \ 1]$ so as to be consistent with the input units of the Speed Control System, i.e. *meters per second* (m/s).

III. Motion Control Method Development

An identification of each side of the robot was performed with a RBS (Random Binary Sequence) test, and the adjusting of an ARX model through minimum squares [11]. The dynamic models of the left and right side of the SSMR are the following:

$$\text{left side} \quad \frac{y_L(q^{-1})}{u(q^{-1})} = \frac{0.256972\,q^{-2} + 0.239032\,q^{-3}}{1 - 0.721039\,q^{-1}}$$

$$\text{right side} \quad \frac{y_R(q^{-1})}{u(q^{-1})} = \frac{0.302718\,q^{-2} + 0.222288\,q^{-3}}{1 - 0.722490\,q^{-1}}$$

A. The R-S-T Controller

When a particular behavior of the plant is required and its dynamic model is known, it is possible to design a digital control system through a pole placement strategy, by using the R-S-T controller [11]. Each side of the robot can be represented as a control plant. The canonical structure of the three branched R-S-T structure is shown in Fig. 5.

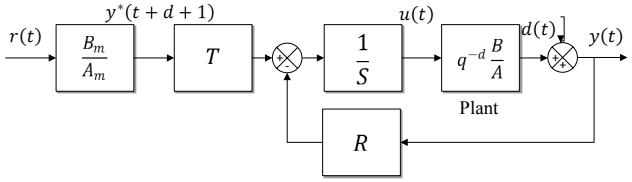

Fig. 5: The R-S-T canonical structure of a digital controller.

The discrete filters used in this controller are the polynomials $R(q^{-1})$, $S(q^{-1})$ and $T(q^{-1})$, which have the form:

$$R(q^{-1}) = r_0 + r_1 q^{-1} + \cdots + r_{nr} q^{-nr}$$
$$S(q^{-1}) = 1 + s_1 q^{-1} + \cdots + s_{ns} q^{-ns}$$
$$T(q^{-1}) = t_0 + t_1 q^{-1} + \cdots + t_{nt} q^{-nt}$$

The closed-loop control transfer function (between $r(t)$ and $y(t)$) is given by:

$$H_{CL}(q^{-1}) = \frac{q^{-d} T(q^{-1}) B(q^{-1})}{P(q^{-1})},$$

where $P(q^{-1})$ determines the desired poles of the closed-loop behavior, and can be expressed in the form.

$$P(q^{-1}) = A(q^{-1})S(q^{-1}) + q^{-d}B(q^{-1})R(q^{-1})$$
$$= 1 + p_1 q^{-1} + p_2 q^{-2} + \cdots + p_{np} q^{np}$$

1) Specification of the Dynamic Behavior: The polynomial $P(q^{-1})$ can be specified with a desired behavior. One can choose a second order model, whose discrete time model is:

$$G(q^{-1}) = \frac{b_1 q^{-1} + b_2 q^{-2} + b_3 q^{-3}}{1 + a_1 q^{-1} + a_2 q^{-2}}$$

where,

$$p_1 = a_1 = 2e^{\zeta \omega_n T} \cos\left(\omega_n T \sqrt[2]{1-\zeta^2}\right);\ p_2 = a_2 = e^{-\zeta \omega_n T}$$

The second order discrete-time model denominator can then become the characteristic polynomial of the entire closed control loop as $P(q^{-1}) = 1 + p_1 q^{-1} + p_2 q^{-2}$.

2) Pole Placement: Once the closed-loop poles have been defined with $P(q^{-1})$, the polynomials $R(q^{-1})$ and $S(q^{-1})$ can be computed by solving the "Benzout identity" (equation) (see [12]), defined as:

$$A(q^{-1})S(q^{-1}) + q^{-d}B(q^{-1})R(q^{-1}) = P(q^{-1}). \tag{8}$$

The polynomial $S(q^{-1})$ is defined to have zero gain to disturbances by adding an integrator:

$$S(q^{-1}) = S'(q^{-1})\left(1 - q^{-1}\right) \tag{9}$$

3) Computation of $T(q^{-1})$: The polynomial T can be obtained as follows, $T(q^{-1}) = K_b P(q^{-1})$, where $P(q^{-1})$ defines the desired closed-loop poles, and

$$K_b = \begin{cases} \frac{1}{B(1)} & \text{if } B(1) \neq 0 \\ 1 & \text{if } B(1) = 0 \end{cases}.$$

4) Closed Loop System Robustness: The transfer function between the disturbance $d(k)$ and the output $y(k)$ is called *output sensitivity function* [11], and it is given by:

$$S_{yd}(q^{-1}) = \frac{A(q^{-1})S(q^{-1})}{P(q^{-1})}$$

A Bode analysis can be applied to $S_{yd}(j\omega)$ in order to obtain its frequency response. In order to ensure system robustness to disturbances, the maximum response magnitude should be less than 6 dB (modulus margin greater than 0.5), as $|S_{yd}(j\omega)|_{max} \leq 6$ dB.

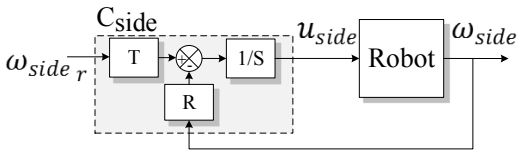

Fig. 6: R-S-T control diagram for one side of the robot.

	Left Side of Robot	Right Side of Robot
$B(q^{-1})$	$0.256972\,q^{-2}+0.239032\,q^{-3}$	$0.302718\,q^{-2}+0.222288\,q^{-3}$
$A(q^{-1})$	$1-0.721039\,q^{-1}$	$1-0.722490\,q^{-1}$
$A'(q^{-1})$	$1-1.721039\,q^{-1}+1.721039\,q^{-2}$	$1-1.72249\,q^{-1}+0.72249\,q^{-2}$
$B_m(q^{-1})$	$0.0188233+0.0164731\,q^{-1}$	
$A_m(q^{-1})$	$1-1.63502\,q^{-1}+0.67032\,q^{-2}$	
	$T=0.1\,s$, $t_r=2\,s$, $\%Mp=0.01\%$	
$P(q^{-1})$	$1-1.63502\,q^{-1}+0.67032\,q^{-2}$	
	$T=0.1\,s$, $t_r=2\,s$, $\%Mp=0.01\%$	
$R(q^{-1})$	$0.186252-0.115091\,q^{-1}$	$0.177185-0.109954\,q^{-1}$
$S(q^{-1})$	$1-0.961846\,q^{-1}-0.038153\,q^{-2}$	$1-0.966171\,q^{-1}-0.033829\,q^{-2}$
$T(q^{-1})$	$2.01611-3.29639\,q^{-1}+1.35144\,q^{-2}$	$1.90474-3.1143\,q^{-1}+1.27679\,q^{-2}$
	Modulus margin: $1.121\,(0.993\,dB)$	Modulus margin: $1.169\,(1.358\,dB)$

TABLE I: Parameters of the R-S-T controller for the left and right side of the Speed Control System

B. Speed Control System Implementation

The Speed Control System (SCS) receives from the P2P the reference speeds in $[v_L, v_R]$ in (m/s), and returns the manipulating signal $[u_L, u_R]$ in $(0\,to\,100)\%$ that will be sent to the actuators on the robot.

The Fig. 6 shows a representation of the R-S-T control system implemented on both the left and right sides of the robot, represented inside the dotted rectangle "C_{side}".

The Table I shows the obtained polynomials of the R-S-T controller. It includes the dynamic model of the left and right side of the robot, as well as the discrete models of the tracking and regulation dynamics, which determine the closed-loop behavior.

The polynomials $B(q^{-1})$ and $A(q^{-1})$ are the numerator and denominator of each side's dynamic model, previously obtained by system identification techniques, respectively. $B_m(q^{-1})$ and $A_m(q^{-1})$ are the tracking dynamics. The regulation dynamics is dictated by the polynomial $P(q^{-1})$, and in this case, the tracking and regulation dynamics are selected to behave like a second order model with time rise $t_r = 2$ and overshoot percentage $\%Mp = 0.01\%$.

The modulus margin of Table I validates that both the left and the right side controllers of the Speed Control System are robust to disturbances, because they are both under the $6\,dB$ limit for the output sensitivity function.

C. Pose Estimation System

For the Pose Estimation System, a dead reckoning scheme is used, in which the pose of the robot is estimated through on-board sensors, using the kinematic model of a SSMR.

The linear velocity v_x and the angular velocity ω of the robot are needed. From [8], the equation (10) can be used to estimate $\eta = [v_x,\ \omega]$ with the angular speed of each side of the robot,

Fig. 7: Jaguar-4x4-Wheel Robotic Platform, Dr. Robot Inc.

$$v_x = r\left(\frac{\omega_L+\omega_R}{2}\right),\ \omega_{enc} = r\left(\frac{-\omega_L+\omega_R}{l}\right),\ \omega = \frac{\omega_{enc}+\omega_Z}{2} \tag{10}$$

where ω_L and ω_R are the left and right side angular speed respectively, provided by wheel encoders, l is the width of the robot, and r is the effective radius of the wheels. The angular velocity of the robot ω is obtained from the average filter of ω_{enc} (from wheel encoders) and ω_Z from IMU Z-axis gyroscope).

Once the estimation of $[v_x, \omega]$ is obtained, the kinematic model can be used to determine the state vector $q = [X\ \ Y\ \ \theta]$.

IV. Experiments and Results

The experimental robotic platform used for this research work is the Jaguar-4x4-Wheel, from Dr. Robot Inc (Fig. 7). The System Architecture scheme was implemented in ROS (Robot Operating System), with C++. An User Interface was developed in order to interact with the robot.

A. Straight Line Experiments

To test the reliability of the Speed Control System, a straight line experiment was conducted. The R-S-T scheme is compared with a PID control scheme. The designed experiment consisted in applying equal speed references to the left and right side of the robot, maintaining these references until the robot reaches ten meters, then command the robot to stop. The final position in the Y axis was measured. A percentage of lateral deviation is calculated after conducting the experiment ten times. These results are condensed in the Table II.

The first column of Table II shows the deviation of the open loop tests, the second column shows the lateral deviation with the PID control scheme, column three shows the same PID control after several tuning, column four shows the deviation with the R-S-T control scheme with a desired overshoot of $\%Mp = 0.1\%$ and a time rise of $t_r = 2\,sec$, columns 5 shows an improved R-S-T control design with $\%Mp = 0.01\%$ and $t_r = 2\,sec$.

As can be seen, the results for the PID and the R-S-T control scheme are very similar, the difference lies in the fact that the PID control scheme required a heuristically tuning session from the user, for both the left and the right side. This control scheme demands more time and effort. On the other hand, the R-S-T control scheme did not require as much time of implementation as the required by the PID control scheme.

	Speed Control Scheme				
	Open loop	PID	R-S-T ($\%Mp = 0.1\%$)	PID (after tuning)	R-S-T ($\%Mp = 0.01\%$)
Percentage Deviation (%)	6.02	5.79	2.35	1.57	1.56

TABLE II: Comparison of the straight line experiments.

	Time rise (sec)	
	Left side	Right side
PID	1.06	0.98
R-S-T	1.75	1.66

TABLE III: Comparison of the speed control step responses.

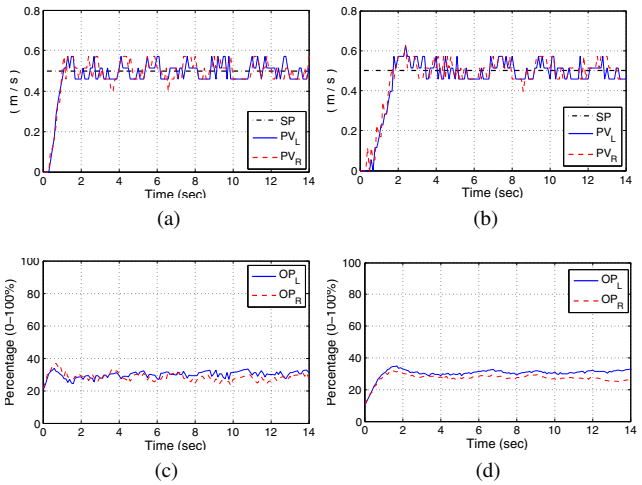

Fig. 8: Speed Control System experimental results. In (a) and (b) is shown the reference and actual speed for the PID and R-S-T type of control respectively, (c) and (d) show the respectively manipulating signal of each control type.

B. Step Response

A step response to a reference of $0.5\,m/s$ of both the R-S-T and the PID control scheme is shown in Fig. 8. The Fig. 8a shows the speed response of the left and right side of the robot in m/s for the PID control system and 8c shows its corresponding manipulating signal. As can be seen the response of the PID control system has a time rise of $t_r = 1.06$ sec on the left side and $t_r = 0.98$ sec on the right side (using the 2% criteria). The response of the PID control system was accomplished after several heuristically tuned iterations.

The R-S-T was digitally designed to behave with the characteristics of a second order model, with a time rise $t_r = 2$ sec and an overshoot of $\%Mp = 0.01\ \%$. The Fig. 8b shows the speed response for the R-S-T control system, Fig. 8d shows its corresponding manipulating signal. The response of the R-S-T control system has $t_r = 1.75$ sec on the left side and $t_r = 1.66$ sec on the right side.

Both control schemes have a difference of 0.08 sec of time rise between the left and right side and both have similar performance. The difference lies in the fact that the PID control of the left and right sides had to be tuned several times, unlike the R-S-T control system, whose desired performance was selected, and no further tuning was necessary. The Table III shows a compilation of the step response results mentioned above.

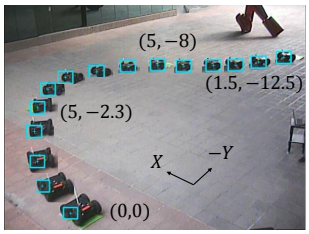

Fig. 9: Trajectory Tracking experiment. Composed image.

C. Trajectory Tracking

The autonomous navigation method was verified with a discrete trajectory $\mathbf{Q}$ that contains three pair of coordinate points (X_r, Y_r), represented as: $\mathbf{Q} = [5, -2.3;\ 5, -8;\ 1.5, -12.5\,]$.

As can be seen in in Fig. 9, the robot was successful to accomplish each of the points in the trajectory $\mathbf{Q}$.

The trajectory tracking experiment was conducted eleven times, in order to ensure repeatability. The trajectory tracking parameters are a desired speed $v_r = 0.3\ m/s$ and an $error\,margin = 0.5\ m$. A 2-D vision system provides information regarding the position of the robot. This information is used only for validation of the autonomous navigation performance.

The Fig. 10 depicts the position of the robot when each reference point was considered reached. The *error margin* is shown in the black circle around each of the reference points. As can be seen, all points from the estimated position of the robot are within the $0.5\ m$ *error margin*. The green $\Delta's$, corresponding to the vision-based position of the robot, are not necessarily within the *error margin* because of the dead-reckoning draft.

The Fig. 11 shows a composed image of the eleven trajectory tracking experiments. An interpolated trajectory contains the reference coordinate points. The black line trajectories are from the Pose Estimation System. The blue dotted line represents the actual position provided by the 2D-vision system.

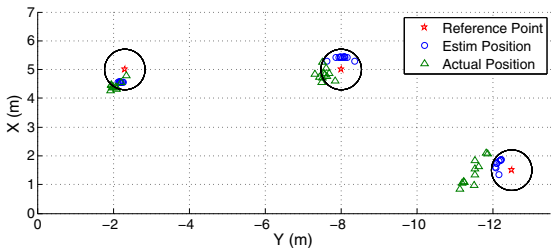

Fig. 10: Trajectory Tracking experiments. Estimated position and actual position at each of the reference points

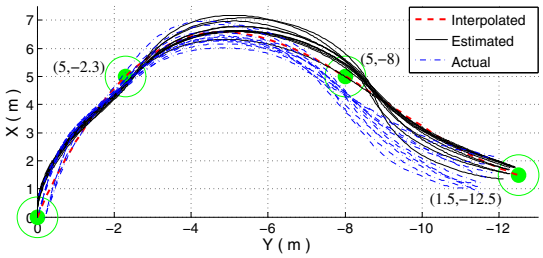

Fig. 11: Compilation of the eleven trajectory tracking experiments. The interpolated trajectory is shown, along with the estimated and the actual position of the robot.

1) Speed Control System Data of the Trajectory Tracking Experiment: After the P2P outputs left and right reference speeds (m/s), the Speed Control System receives these speeds and provides a manipulating signal with range $[0\ \ 100]\,\%$ on the left and right sides that will be sent through the wireless network to the Jaguar-4x4-Wheel robot.

The Fig. 12 shows the Speed Control System input and output data. In Fig. 12a and Fig. 12b the reference speed (SP) of the left and right side, as well as the actual speed (PV) in m/s is shown. To be able to turn, the P2P gives opposite symmetric reference speed to the left and right sides. To reach the points $(5,-2.3)$ and $(5,-8)$, the robot turns right (CW), so the left is greater than the right reference speed. To reach $(1.5,-12.5)$, the robot turns left (CCW).

As can be seen in Fig. 12a and Fig. 12b, when turning, even when the one of the speed references is zero, the actual speed of that side is non-zero, because the left and right side of the robot are not independent, and a dragging phenomenon occurs. It should be noted that the highest reference speed on either side is followed consistently.

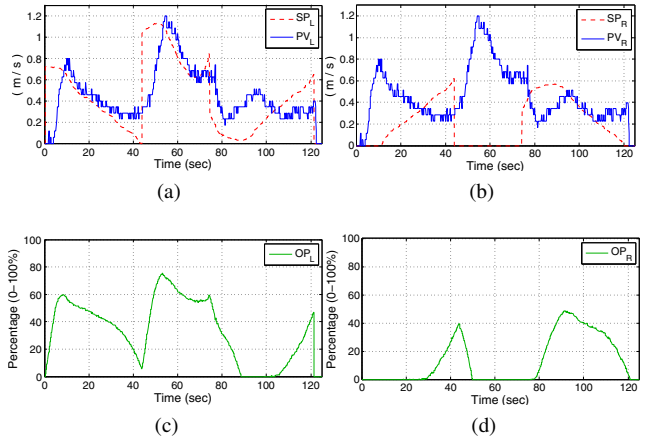

Fig. 12: Speed Control System results while the Trajectory Tracking experiment was conducted where (a) and (b) show the reference speed and the actual speed of the left and right side respectively, (c) and (d) show the manipulating signal from the left and right side respectively.

V. Conclusions

In this paper, a motion control scheme was developed in order to perform autonomous navigation of a skid-steered mobile robot. A Speed and Point-to-Point Control System was designed and implemented in a real world robotic platform. The advantage of a digitally designed controller is that the desired dynamic behavior of the plant can be selected. This configuration has proven to be useful in a Skid-Steer Mobile Robot, because each side of the robot must have very similar transient responses, or the robot would fail to achieve the intended movement for trajectory tracking. Satisfactory results were obtained in the context of trajectory tracking, were the robot successfully arrived to every reference point within a margin.

Acknowledgment

This research was conducted at the Laboratorio Nacional del Noreste y Centro de México supported by CONACyT and Tecnológico de Monterrey.

References

[1] V. Nazari and M. Naraghi, "A vision-based intelligent path following control of a four-wheel differentially driven skid steer mobile robot," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, 2008, pp. 378–383.

[2] J. Yi, D. Song, J. Zhang, and Z. Goodwin, "Adaptive trajectory tracking control of skid-steered mobile robots," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 2605–2610.

[3] J. Zhang, "Autonomous navigation for an unmanned mobile robot in urban areas," in *Mechatronics and Automation (ICMA), 2011 International Conference on*, aug. 2011, pp. 2243 –2248.

[4] C. Tu, G. Qi, B. van Wyk, and S. Du, "Motion control and stabilization of a skid-steering mobile robot," in *Adaptive Science Technology, 2009. ICAST 2009. 2nd International Conference on*, 2009, pp. 325–330.

[5] S. Arslan and H. Temeltas, "Robust motion control of a four wheel drive skid-steered mobile robot," in *Electrical and Electronics Engineering (ELECO), 2011 7th International Conference on*, 2011, pp. II–415–II–419.

[6] L. Pacheco and N. Luo, "Experiences with online local model predictive control for wmr navigation," in *Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on*, 2010, pp. 370–377.

[7] Y. Morales, E. Takeuchi, and T. Tsubouchi, "Vehicle localization in outdoor woodland environments with sensor fault detection," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 449–454.

[8] K. Kozlowski and D. Pazderki, "Modeling and control of a 4-wheel skid-steering mobile robot," in *Applied Mathematics and Computer Science, 2004. AMCS 2004., International Journal in*, vol. 14, 2004, pp. 477–496.

[9] L. Caracciolo, A. De Luca, and S. Iannitti, "Trajectory tracking control of a four-wheel differentially driven mobile robot," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 4, 1999, pp. 2632–2638 vol.4.

[10] M. G. Campa, "Autonomous navigation method for trajectory tracking based on speed and point-to-point control using pose estimation," Master's thesis, Tecnológico de Monterrey, 2013.

[11] I. Landau and G. Zito, *Digital Control Systems: Design, Identification and Implementation*, ser. Communications and Control Engineering Series. Springer-Verlag, 2006. [Online]. Available: http://books.google.com.mx/books?id=s-LJdcGpTc8C

[12] G. Goodwin and K. Sin, *Adaptive Filtering Prediction and Control*, ser. Dover Books on Electrical Engineering Series. Dover Publications, Incorporated, 2009. [Online]. Available: http://books.google.com.mx/books?id=tDt9mAEACAAJ