Low Level Control in States Space for the Pioneer

I. García Daza, L. M. Bergasa Pascual, M. A. Sotelo Vázquez, E. López Guillén,
R. Barea Navarro, L. Boquete Vázquez.

***Abstract*— In this paper a new model and low level control system based on state variables theory for the commercial robotic platform called Pionner 2 has been developed. We have identified and modelled the robot from the angular speeds of the wheels. We have controlled the translation and rotation speed of the robot, the orientation angle and the position increments. We have used encoders as position sensors. A comparation between the default model and control proporcioned by Aria, commercial software of ActivMedia robotics company, for the Pioneer 2 and our controller, called Charkos, has been carried out using several tests.**

***Index Terms*— Control, mobile robots, states space.**

I. Introduction

NOWADAYS there are several projects and research groups working on the development of assistant robots. In order to contribute to this research field, the Electronics Department of the University of Alcal is working on the SIRAPEM project (Spanish acronym of Robotic System for Elderly Assistance). The goal of this project is the development of a robotic assistant (called SIRA) which allows the user to be completely monitored 24 hours a day and tele-diagnosed from the assistance centers. SIRA is based on a commercial platform called Pioneer 2 of ActivMedia Robotics [1]. This is a very popular robot used as experimental mobile platform by a lot of research groups in the world. The robot has a differential drive mobile base and it is composed of four wheels moved by two motors. Each of them moves the two wheels of its size. SIRA is endowed with one encoder per motor, bumpers, two sonar rings (high and low), a 2D laser and a vision system based on a PTZ (pan-tilt-zoom) color camera connected to a frame grabber. This paper is focused on a new model and low control system design for the Pioneer 2 based on the states space theory and called Charkos. First of all a review of the main works respecting the design of low level controllers for the Pionner 2 are presented. Then, a comparison between our system and the default model and control proportioned by Aria for the Pioneer 2 and based on fuzzy logic is carried out. Finally, some conclusions about the comparison are presented.

II. Previous works

Focusing the study on the Pioneer 2, there are a lot of works about control systems using this robot. **Diolaiti N.** [2] study the robot movements in unknown indoor environments. The system generates a map online that can change time along. The control system is based on encoders and ultrasounds. Due to the low level control is based on odometry the accumulative errors obtained by the system are very high. **Agostino Martinelli** [3] models in robot translation and rotation errors. **Lauro Ojeda** and **Johann Borenstein** [4] describe three methods to reduce encoders lecture errors. Likewise, the correct calibration of encoders is based to known the robot position. ***Stergios I.** et al.* [5] use a Kalman filter that works on encoders data in order to improve position and orientation of the robot. **Zachary Randles** et al. [6] control several robots and the basic behaviour algorithms used are to go to a goal, to avoid obstacles, to keep a certain distance and angle among them and to stop. All these actions mean at last, to control wheels robot angular speed, which is done by a PID. This is the starting point of our work.

III. Identification

Pioneer 2 robot, from ActivMedia Robotics, which wheels angular speeds are controlled by a PID, has been modelled by a system of two equations of order two and it has been identified by Matlab [7]. The equation, at state variables is shown in (1).

$$\vec{W}(k+1) = \mathbf{A}\,\vec{W}(k) + \mathbf{B}\,\vec{W}_{REF}(k) \qquad (1)$$

being **A** and B:

$$\mathbf{A} = \begin{bmatrix} 1.3476 & 0 & -0.4323 & 0 \\ 0 & 0.9767 & 0 & -0.0601 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0.0759 & 0 \\ 0 & 0.0849 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

the state vector:

$$\vec{\mathbf{W}}(k) = \begin{bmatrix} \omega_R(k) \\ \omega_L(k) \\ \omega_R(k-1) \\ \omega_L(k-1) \end{bmatrix}$$

that represents right and left wheels angular speeds at the moment (K) and (K-1) and the inputs:

$$\vec{W}_{REF}(k) = \begin{bmatrix} \omega_{R\,REF}(k) \\ \omega_{L\,REF}(k) \end{bmatrix}$$

that represent rigth and left wheels angular speeds references at moment (K).

Department of Electronics, University Alcalá, Alcalá de Henares, 28805 Madrid, Spain.
email: igdaza1977@avired.com,
{bergasa,sotelo,elena,barea,boquete}@depeca.uah.es

IV. Mobile kinematic

A. Robot translation and rotation speeds

If we divide the robot movement in the plane, one into a translation component and another one into a rotation component, we will obtain equations (2) and (3), which represents robot translation and rotation speeds.

$$\overrightarrow{V} = R\frac{\omega_R + \omega_L}{2}\,\hat{\imath} \tag{2}$$

$$\overrightarrow{\Omega} = \frac{180\,R}{\pi\,D}\,(\omega_R - \omega_L)\,\hat{k} \tag{3}$$

R is the wheel radius and D the distance between wheels.

Over the previous equations a state variable change is done by the transformation matrix **MDV**:

$$\mathbf{MDV} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{180}{\pi\,D} & -\frac{180}{\pi\,D} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{180}{\pi\,D} & -\frac{180}{\pi\,D} \end{bmatrix}$$

and also an inverse transformation **MIV** in the input reference signals:

$$\mathbf{MIV} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{180}{\pi\,D} & -\frac{180}{\pi\,D} \end{bmatrix}^{-1}$$

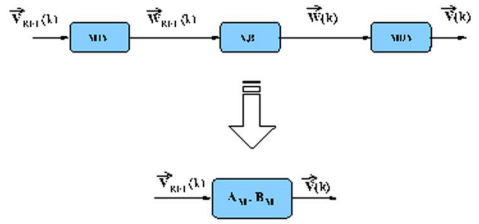

Fig. 1. Model of the mobile translation and rotation speeds

The general equation is given by (4).

$$\overrightarrow{\mathbf{V}}(k+1) = \mathbf{A}_M\,\overrightarrow{V}(k) + \mathrm{B}_M\,\overrightarrow{V}_{\mathbf{REF}}(k) \tag{4}$$

where

$$\mathbf{A}_M = \mathbf{MDV}\cdot\mathbf{A}\cdot\mathbf{MDV}^{-1}$$

$$\mathrm{B}_M = \mathbf{MDV}\cdot\mathrm{B}\cdot\mathbf{MIV}$$

B. Orientation angle $\theta(k)$

Mobile orientation changes during a sampling period in $\Omega(k)\,T$ degrees, so the equation that controls the orientation is:

$$\theta(k+1) = \theta(k) + T\,\Omega(k) \tag{5}$$

This equation represents an unstable dinamic system because its eigenvalue is into the unit circle. The complete system is depicted in figure (2).

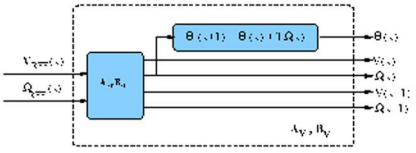

Fig. 2. Orientation angle diagram.

which equation will be:

$$\begin{bmatrix} \theta(k+1) \\ V(k+1) \\ \Omega(k+1) \\ V(k) \\ \Omega(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 & 0 \\ 0 & & & & \\ 0 & & [\mathbf{AM}] & & \\ 0 & & & & \\ 0 & & & & \end{bmatrix} \begin{bmatrix} \theta(k) \\ V(k) \\ \Omega(k) \\ V(k-1) \\ \Omega(k-1) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ [\mathbf{BM}] & \end{bmatrix} \begin{bmatrix} V_{REF}(k) \\ \Omega_{REF}(k) \end{bmatrix} \tag{6}$$

or in a matrix way:

$$\overrightarrow{\mathbf{\Theta}}(k+1) = \mathbf{A}_V\,\overrightarrow{\Theta}(k) + \mathrm{B}_V\,\overrightarrow{V}_{\mathbf{REF}}(k) \tag{7}$$

To control the state vector $\overrightarrow{\Theta}(k) = \begin{bmatrix} \theta(k) \\ V(k) \\ \Omega(k) \\ V(k-1) \\ \Omega(k-1) \end{bmatrix}$ it's been used [8] the blocks diagram shown at figure (3). The

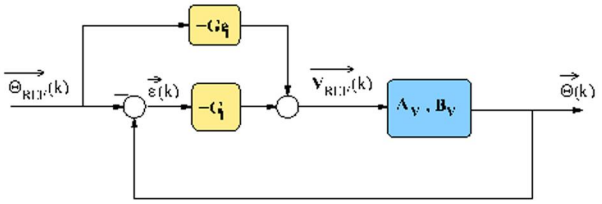

Fig. 3. Control diagram of orientation angle.

control goal has consisted on minimizing the scalar const function shown in (8).

$$J = \frac{1}{2}\sum\left[\overrightarrow{\epsilon}^T(k)\,\mathbf{Q}\,\overrightarrow{\epsilon}(k) + \overrightarrow{V_{REF}}^T(k)\,\mathrm{R}\,\overrightarrow{V_{REF}}(k)\right] \tag{8}$$

with $\overrightarrow{\epsilon}(k) = \overrightarrow{\Theta}(k) - \overrightarrow{\Theta}_{REF}(k)$ and the matrixes **Q** y **R** which value is:

$$\mathbf{Q} = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$\mathbf{G}_1$ value is calculated by Matlab **dlqr** function, obtaining:

$$\mathbf{G}_1 = \begin{bmatrix} -0.1587 & 0.8870 & 0.5379 & -0.2468 & -0.9817 \\ 2.0129 & 0.2195 & 2.4687 & -0.1455 & -0.8023 \end{bmatrix}$$

$\mathbf{Ge}_1$ value is calculated to eliminate the error in steady state [8], resulting (9).

$$\mathbf{Ge}_1 = \mathbf{M}_1\mathrm{B}_v \setminus \mathbf{M}_1(\mathbf{A}_v - \mathbf{I}) \tag{9}$$

with $\mathbf{M_1} = (\mathbf{I} - \mathbf{A}_v + \mathbf{B}_v\mathbf{G}_1)^{-1}$, so we obtain:

$$\mathbf{G_{e1}} = \begin{bmatrix} 0 & 0.0422 & 0.5379 & -1.0916 & -0.9817 \\ 0 & 0.1759 & 2.4687 & -0.1892 & -0.8023 \end{bmatrix}$$

The equation corresponding to the closed loop system depicted in the figure (3) is:

$$\vec{\Theta}(k+1) = (\mathbf{A_v} - \mathbf{B_v}\mathbf{G_1})\vec{\Theta}(k) + \mathbf{B_v}(\mathbf{G_1} - \mathbf{Ge_1})\,\vec{\Theta}_{REF}(k) \tag{10}$$

or doing some transformations:

$$\vec{\Theta}(k+1) = \mathbf{A_{Th}}\,\vec{\Theta}(k) + \mathbf{B_{Th}}\,\vec{\Theta}_{REF}(k) \tag{11}$$

with:

$$\mathbf{A_{Th}} = \mathbf{A_v} - \mathbf{B_v}\mathbf{G_1}$$
$$\mathbf{B_{Th}} = \mathbf{B_v}(\mathbf{G_1} - \mathbf{Ge_1})$$

We have compared the results obtained from Aria (using the commands *setVel()* and *setHeading()*) and our system Charkos for some references as 0 mm/s of translation speed and orientations of 90°, 180° and 360°. This comparation can be seen in figure (4). Results obtained from Charkos are more precise than the obtained from Aria.

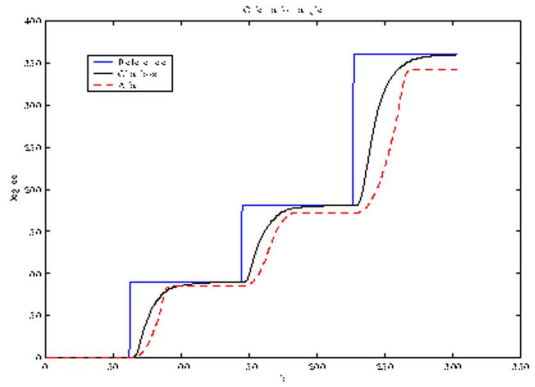

Fig. 4. Orientation angle.

C. Position $\overrightarrow{\Delta x}$ increment

At time k, the mobile is in the position defined by the coordinates [x(k), y(k)] and it is orientated according to $\theta(k)$ angle.

During the sampling period existing from k to k+1, the vehicle will have covered a space, due to the translation, with value $V(k)\,T$ and at the same time, it will have turn an angle of $\Omega(k)\,T$ degrees.

Having in mind that the sampling period is short (T=100 ms), orientation angle in a sampling period will have slightly changed, so we could consider, taking into account the figure (5), that:

$$\Delta x(k) = T\,V(k)\,\cos\theta(k) \tag{12}$$

$$\Delta y(k) = T\,V(k)\,\sin\theta(k) \tag{13}$$

These equations constitute a variable change, **MDX**, from $\begin{bmatrix} \theta(k) \\ V(k) \end{bmatrix}$ to $\begin{bmatrix} \Delta x(k) \\ \Delta y(k) \end{bmatrix}$ which is no lineal and variable with time.

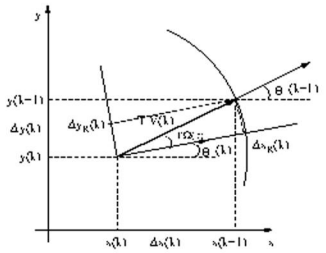

Fig. 5. Position increments.

$$\overrightarrow{\mathbf{\Delta x}}(k) = \mathbf{MDX}\,\vec{\mathbf{\Theta}}(k) \tag{14}$$

In order to apply position increments references, in the input directly, it is necessary to introduce a variable change, inverse to the previous one in the reference signals. That is to say we have to find the equaations that go from $\overrightarrow{\mathbf{\Delta x}}_{REF}(k)$ to $\vec{\mathbf{\Theta}}_{REF}(k)$.

The inverse transformation **MIX** is:

$$\theta_{REF}(k) = \arctan\frac{\Delta y_{REF}(k)}{\Delta x_{REF}(k)} \tag{15}$$

$$V_{REF}(k) = \frac{\sqrt{\Delta x^2_{REF}(k) + \Delta y^2_{REF}(k)}}{T} \tag{16}$$

1) Position increments limits: As

$$\Delta x(k) = T\,V(k)\,\cos\theta(k)$$
$$\Delta y(k) = T\,V(k)\,\sin\theta(k)$$

and taking into account that the highest translation speed is 800 mm/s, the result is:

$$\Delta x^2(k) + \Delta y^2(k) = [TV(k)]^2 \leq (0{,}1 \cdot 800)^2 = 80^2 \tag{17}$$

This inequation determines a 80mm radius circle.

D. Control position

As it is shown in figure (5) the mobile coordinates change depending on the equation:

$$\begin{bmatrix} x(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} \Delta x(k) \\ \Delta y(k) \end{bmatrix} \tag{18}$$

The system that is wanted to be controller is shown in figure (6)

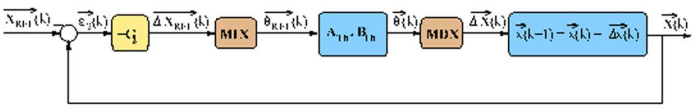

Fig. 6. Control position.

where the subsystem **MIX**, $\mathbf{A}_{TH}$, $\mathbf{B}_{TH}$ y **MDX** is already controlled and their equations are not linear.

If it is consider that $\overrightarrow{\varepsilon}_2(k)$ error is, precisely, a position increment, if it is transformed by a G_2 attenuator in order to be into the limits, which have been established in the previous paragraph, the system would be controlled.

If $\overrightarrow{\Delta X_{REF}}(k) = -G_2\,\overrightarrow{\varepsilon_2}(k)$ is out of $80mm$ radius circle the reference could not be tracked. In order to avoid it, the gain variable with the time $G_2(k)$ is used. It will have to carry out the inequation (17), which is the same as:

$$g_{11}^2\left[x(k) - x_{REF}(k)\right]^2 + g_{22}^2\left[y(k) - y_{REF}(k)\right]^2 \leq 80^2 \quad (19)$$

Supposing that $g_{11} = g_{22} = g$ the result is:

$$g(k) \leq \frac{80}{\sqrt{\left[x(k) - x_{REF}(k)\right]^2 + \left[y(k) - y_{REF}(k)\right]^2}} \quad (20)$$

So we obtain:

$$\mathbf{G}_2(k) = \begin{bmatrix} g(k) & 0 \\ 0 & g(k) \end{bmatrix} \quad (21)$$

When the position is so far away from the reference, the gain will be small and the increment always be into the 80 mm circle. When it is approching:

$$\mathbf{x}(k) - \mathbf{x}_{REF}(k) \to 0$$

the gain would tend to infinity. To avoid it, when the obtained position is into a 5 cm radius circle and the centre is in the reference position, a stop order is sent.

The results obtained from Aria and Charkos for the X and Y coordenates in order to follow the trajectory defined by $x_{REF}(k) = 0mm$ e $y_{REF}(k) = 0mm$, ($x_{REF}(k) = 1000mm$, $y_{REF}(k) = 1000mm$), ($x_{REF}(k) = 1000mm$, $y_{REF}(k) = 0mm$) and ($x_{REF}(k) = 0mm$ e $y_{REF}(k) = 0mm$) is shown in figure (7).

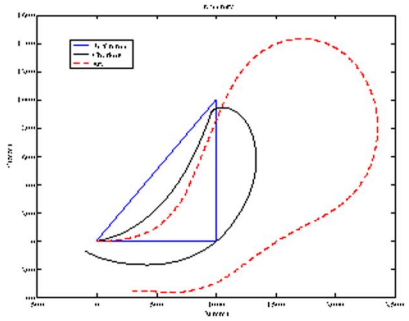

Fig. 7. Trajectory.

The general blocks diagram that gather all the control operations made, is indicated in the figure (8).

V. Conclusions

With Charkos algorithm the orientation obtained with the Pionner 2 is quite more precise in comparation with Aria. The error in robot turning 360° is less than 1° for the Charkos system, while Aria's error is 18° and is accumulative. Settling time in both cases is equivalent. In

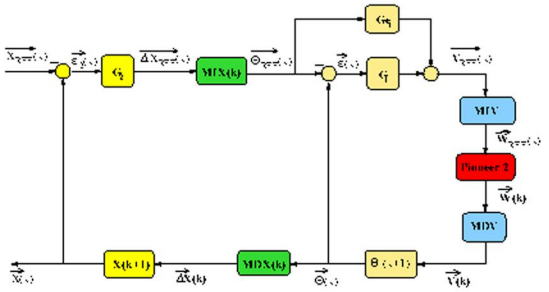

Fig. 8. General block diagram.

the case of the position, Charkos is also more precise in the trajectory tracking than Aria. The highest deviation error for Charkos in X coordinate is 30 cm, while for Aria is 140 cm. Respecting the goal point, Charkos' deviation, in X coordenate, is 9 cm, while Aria's is 30 cm. In the case of Y coordenate, Charkos' highest deviation is 15 cm, and Aria's is 40 cm. Respecting the goal point, Charkos' deviation is 6 cm and Aria's is 35 cm. The transient response is in both cases similat respecting to the settling time, but the overshoots are rather so much bigger in Area than in Charkos. To sum up, Charkos' algorithms based on classic control in state variables, give more accuracy than Aria's commercial algorithms, specially respecting the error in the orientation angle, due to the fact it's being accumulative as function of time. Then, the position errors can be very significant.

Acknowledgement

This work has been supported by grants DPI2002-02193 (SIRAPEM Project) from the Spanish Ministry of Science and Technology (MCyT) and GR/SAL/0860/2004 (TEAPEM Project) from the Education Department of the Madrid Community.

References

[1] A. Robotics, *Pioneer 2-H8S Operations Manual*, August 2002, vol. version 1.0.

[2] D. N., *Sistema di navigazione per robot mobili in ambienti sconosciuti*. Universita degli Studi di Bologna - Facolta di Ingegneria - DEIS.

[3] A. Martinelli, *The Odometry Error of a Mobile Robot With a Synchronous Drive System*. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, JUNE 2002, vOL. 18, NO. 3.

[4] L. Ojeda and J. Borenstein, *Reduction of Odometry Errors in Over-constrained Mobile Robots1*. University of Michigan: Proceedings of the UGV Technology Conference at the 2003 SPIE AeroSense Symposium, April 21-25, 2003.

[5] S. I. R. Puneet Goel and G. S. Sukhatme, *Robot Localization Using Relative and Absolute Position Estimates*. University of Southern California. Los Angeles: Institute for Robotics and Intelligent Systems.

[6] R. Dougherty, V. Ochoa, Z. Randles, and C. Kitts, *A Behavioral Control Approach to Formation-Keeping Through an Obstacle Field*, ser. Version 5. IEEEAC paper 1357, December 12, 2003.

[7] *System Identification Toolbox*. The Math Works, Julio 1991.

[8] B. Friedland, *Control System Design*, 2nd ed., ser. Electrical Engineering. Singapore: McGraw-Hill, 1987, iSBN 0-07-100420-3.