Abstract: This work presents the union of two synchronization approaches, the first one is at Cartesian level and the second one at wheel level, in order to keep a desired formation with a swarm of differential mobile robots. The first part of the synchronization scheme involves the entire group of mobile robots, its aim is maintaining a desired spatial formation along the time. The second part is applied individually to control the orientation angle of each robot by the torque applied on each wheel, thus every robot keeps its desired trajectory. The design of the control scheme on two levels is descentralized and it is based in the coupling parameters between one robot and its nearest neighbors. The experimental platform are four iRobot Create® that are controlled at speed level.

Keywords: synchronization; differential mobile robot; control; swarm formations.

1. INTRODUCTION

The synchronization schemes are important in tasks that involve two or more mobile robots, for example, move or carry heavy objects (Rodriguez-Angeles and Nijmeijer [2004]) or in the case of manufacturing cells where they must perform several tasks in simultaneous manner. According to Blekhman et al [1995] synchronization can be defined as the mutual agreement in time of two or more processes. This phenomenon aroused scientific interest since it was observed in vibro-exciters and unbalanced rotors (Blekhman [1988]).

In the past decade, Cartesian level synchronization works were developed for omnidirectional robots (Sun and Wang [2007]), which proposes a control that ensures that multiple omnidirectional robots maintain a desired formation. Synchronization techniques have also been used in differential mobile robots as in Nijmeijer and Rodriguez-Angeles [2004] where a control at wheel level is proposed in order to follow a desired trajectory. Moreover in Slotine and Chung [2009] the dynamics of various groups of Lagrangian systems are synchronized to track a desired common trajectory. The methodology proposed in this paper is to use the synchronization Cartesian approach to generate a desired trajectory in the plane, which implies the desired spatial formation, the path is mapped at the wheel level with the pseudo-inverse of the Jacobian matrix of each differential mobile robot and finally synchronization at wheel level is applied to follow the desired path (figure 1).

The paper is structured as follows: Section II explains the model of a differential mobile robot. Section III describes the synchronization functions used in Cartesian level which provide a desired formation and generate the desired trajectory. Section IV describes the procedure to map the desired path of the Cartesian space to wheel space. In Section V synchronization at wheel level is used to control the wheels of each mobile robot in order to follow the desired path and thus the formation. Section VI shows the results of simulations and experiments, finally in Section VII conclusions are exposed.

Fig. 1. Proposed methodology.
radius that it is assume to be identical for both wheels.

The kinematic model of a mobile robot in Cartesian coordinates is given by:
\[
\dot{x} = V \cos(\theta), \quad \dot{y} = V \sin(\theta), \quad \dot{\theta} = \omega
\]  
(1)
Where \( V \) is the translational velocity of the mobile robot and \( \omega \) is the angular velocity, furthermore a non-holonomic constraint is imposed \((\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0)\) because of the characteristics of the differential mobile robot.

3. SYNCHRONIZATION AT CARTESIAN LEVEL

At Cartesian level when orientation is neglected, the position of the \( i \)th-mobile robot is given by:
\[
q_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}
\]  
(2)
The Cartesian position error for this robot can be defined as:
\[
\epsilon_i = q_i^d - q_i
\]  
(3)
The formation idea is to control the swarm such that the position errors of each robot at the swarm converges to zero and furthermore, the robots maintain some established formation along the time. This formation control problem can be solved using the synchronization control concept shown in Sun and Wang [2007] where the synchronization function is defined as \( f(q_1 \ldots q_n) \).

When the goal is to keep the \( i \)th-robot in a specific formation, in this case an ellipse, as the one shown in the figure 3, the approach in Sun and Wang [2007] might be considered, where the desired position for the \( i \)th-robot is given by:
\[
q_i^d = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \cos(\phi_i) \\ \sin(\phi_i) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}
\]  
(4)
Where \( a \) and \( b \) are the longest and shortest ellipse’s radices respectively, and:
\[
\phi_i = \arctan \left( \frac{a \sin(\alpha_i)}{b \cos(\alpha_i)} \right)
\]  
(5)
\[
\alpha_i = \frac{y_i}{x_i}
\]  
(6)
The equations (5) and (6) are designed such that \( \alpha \) is always constant, therefore the position of \( i \)th-robot always will have the same orientation angle with respect to the formation’s center. With the equations (5) and (6) it is guaranteed that the robots are in formation along the time.

The synchronization function associated to the formation can be defined as:
\[
f(q_1 \ldots q_n) = c_1 q_1 + c_2 q_2 = \cdots = c_n q_n = \begin{bmatrix} a \\ b \end{bmatrix}
\]  
(7)
Where \( c_i \) are the coupling matrices non equal to zero of the robot \( i \). If the desired Cartesian coordinates are considered, then from (7) the synchronization function is:
\[
f(q_1^d \ldots q_n^d) = c_1 q_1^d + c_2 q_2^d = \cdots = c_n q_n^d
\]  
(8)
Considering the position error (3) equations (7) and (8) can be combined:
\[
c_1 \epsilon_1 = c_2 \epsilon_2 = \cdots = c_n \epsilon_n
\]  
(9)
The equation (9) involves the formation control goal because the position errors of all robots are included. The goal of the synchronization in equation (8) can be divided in sub-goals if \( c_i e_i = c_{i+1} e_{i+1} \), with the boundary condition: \( i = n, i + 1 = 1 \).

The synchronization error concept can be defined as a subset of all the possible pairs of neighbor robots in the formation, this implies a coupling between a mobile robot and its nearest neighbors, this is:
\[
\epsilon_1 = c_1 e_1 - c_2 e_2 \\
\epsilon_2 = c_2 e_2 - c_3 e_3 \\
\vdots \\
\epsilon_n = c_n e_n - c_1 e_1
\]  
(10)
Where \( \epsilon_i \) is the synchronization error of the robot \( i \) in the formation with respect to next neighbor.

Now the control problem is stated as that the \( n \) robots converge to their desired position such that \( \epsilon_i = 0 \) while the equation (9) is achieved making that the synchronization errors in (10) tend to zero. The coupled position error is defined as:
\[
E_i = c_i \epsilon_i + \beta_c \int_0^t (\epsilon_i(w) - \epsilon_{i-1}(w))dw
\]  
(11)
Where \( \beta_c \) is a positive constant gain, the synchronization error \( \epsilon_i \) is limited to the boundary condition; if \( i = 1, \)
$i - 1 = n$. Deriving the coupled position error the next equation is obtained:

$$
\dot{E}_i = \dot{c}_i c_i + \dot{c}_i c_i + K_D (x_i - x_{i-1})
$$

(12)

To keep the swarm formation a reference acceleration at Cartesian level is defined of the form:

$$
\ddot{q}_i = \tau_{ci}
$$

(13)

Where $\tau_{ci}$ are auxiliary signals to generate the desired Cartesian acceleration, therefore the position on the plane for each mobile robot on the formation. With the equations (11) and (12) is possible to create a tracking control that makes that $E_i \rightarrow 0$ and $\dot{E}_i \rightarrow 0$, to do this an auxiliary reference is considered.

$$
u_i = c_i \dot{\nu}_i + \dot{c}_i c_i + K_D (x_i - x_{i-1}) + \Lambda \dot{E}_i
$$

(14)

Where $\Lambda$ is a diagonal matrix of positive gains. The auxiliary references defined in (14) lead to the next position/velocity vectors:

$$
r_i = u_i - c_i \dot{\nu}_i = c_i \dot{c}_i + \dot{c}_i c_i + K_D (x_i - x_{i-1}) + \Lambda \dot{E}_i
$$

(15)

Its derivative is $\dot{r}_i = \dot{u}_i - c_i \ddot{\nu}_i - c_i \dot{\nu}_i$. Now, it is important to design the control vector $\tau_{ci}$ that restricts $r_i$ at a descending surface such that $E_i$ and $\dot{E}_i$ tend to zero. The controller is decentralized because it only considers the synchronization of a robot with its nearest neighbors, it means that to control the position of a single robot the information of the whole swarm is not needed. The auxiliary control $\tau_{ci}$ is defined as:

$$
\tau_{ci} = A_r (\dot{\nu}_i - c_i \dot{\nu}_i) + K_r A_r \dot{r}_i + c_i^T K_D (x_i - x_{i-1})
$$

(16)

Where $K_r$ and $K_D$ are positive gains, the last term in (16) is used to compensate for the effects of introducing coupling parameters. If the equation (16) is replaced in (13) then a close loop dynamic is gotten:

$$
A_r \dot{r}_i + K_r A_r \dot{r}_i + c_i^T K_D (x_i - x_{i-1}) = 0
$$

(17)

With this controller, the Lyapunov function to prove stability is designed as shown in (Sun and Wang [2007]).

$$
V_e = \sum_{i=1}^{n} \left[ \frac{1}{2} (e_i - 1)^T (e_i - 1) r_i + \frac{1}{2} e_i^T K_d e_i \right]
$$

$$
+ \frac{1}{2} \left[ \int_{0}^{t} \sum_{i=1}^{n} (e_i (w) - e_{i-1} (w)) d w \right] \Lambda \dot{K}_e \ldots
$$

$$
\ldots \left[ \int_{0}^{t} \sum_{i=1}^{n} (e_i (w) - e_{i-1} (w)) d w \right]
$$

$$
\dot{V}_e = \sum_{i=1}^{n} \left[ c_i^{-1} r_i^T (-K_r + c_i^{-1} c_i) c_i^{-1} r_i \right] - \sum_{i=1}^{n} e_i^T \Lambda \dot{K}_e e_i
$$

$$
- \sum_{i=1}^{n} (e_i - e_{i+1}) \beta \dot{K}_e e_i (x_i - x_{i+1})
$$

$$
\therefore \lambda_{\min} (K_r) \geq \lambda_{\max} (c_i^{-1} c_i)
$$

The minimum eigenvalue of the matrix $K_r$ must be greater or equal than the maximum eigenvalue of the matrix $c_i^{-1} c_i$.

4. MAPPING FROM CARTESIAN COORDINATES TO WHEEL LEVEL

The Cartesian synchronization generates a desired path on the plane for each mobile robot keeping a formation, but it is wanted to obtain the desired trajectory at wheel level, then the mapping between the Cartesian position and the angular position of each wheel is needed. Therefore the first step is considering equation (1) and the figure 2, where:

$$
V = (r_w \dot{\psi}_1 + r_w \dot{\psi}_2)/2, \quad \omega = (r_w \dot{\psi}_2 - r_w \dot{\psi}_1)/(2R)
$$

(18)

Where $\dot{\phi}_1$ and $\dot{\phi}_2$ are the angular velocities for the right wheel and left wheel respectively. The Jacobian matrix of the mobile robot ($J$) that maps the wheel velocities to Cartesian space is given by:

$$
\begin{pmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{pmatrix} =
\begin{pmatrix}
r_w \cos(\theta) & r_w \cos(\theta) & \frac{r_w}{2R} \\
r_w \sin(\theta) & r_w \sin(\theta) & \frac{-r_w}{2R} \\
\frac{\dot{\psi}_2}{\dot{\psi}_1}
\end{pmatrix}
\begin{pmatrix}
\dot{\psi}_1 \\
\dot{\psi}_2
\end{pmatrix} = J
\begin{pmatrix}
\dot{\psi}_d \\
\dot{\psi}_d
\end{pmatrix}
$$

(19)

5. SYNCHRONIZATION AT WHEEL LEVEL

Continuing with the ideas in Nijmeijer and Rodriguez-Asenjo [2004], $\psi_1$ and $\psi_2$ are de angular displacements of the left wheel and right wheel respectively, $v_1$ and $v_2$ are the lineal velocities, then mobile robot’s orientation angular velocity is given by:

$$
\dot{\theta} = \frac{v_2 - v_1}{2R} = \frac{r_w}{2R} (\dot{\psi}_2 - \dot{\psi}_1)
$$

(20)

Integrating the equation (20) the robot’s orientation angle can be obtained:

$$
\theta = \int_{0}^{t} \dot{\theta}(\xi) d \xi = \theta_c + \frac{r_w}{2R} (\psi_2 - \psi_1)
$$

(21)

$\theta_c$ is an integration constant that depends of the initial conditions of the mobile robot.

The goal of the synchronization is to regulate the angular position of the wheels $\psi_1$, $\psi_2$ to the desired values $\psi^d_1$, $\psi^d_2$ (figure 4). Then it is possible to define an angular displacement error:

$$
e_{\text{ang}} = \psi_1 - \psi^d_1
$$

(22)

This error tends to zero regulating the orientation angle $\theta$ in (21), this means to follow a desired trajectory controlling only the mobile robot’s wheels. Based on the figure 4 some considerations should be made for the desired trajectory.

- The trajectory must be soft and continuous.
- The difference of the curvature radius $R_{ca}$ y $R_{cd}$ is small enough to be neglected; $R_{ca} \approx R_{cd} = R_c$.
- The curvature radius $R_c$ must be $R_c > R$. 


To keep the robot in the desired path, the wheel synchronization error is defined:

$$\epsilon = D_1 e_{w1} + D_2 e_{w2}$$

Where $D_1$ are the cross-coupling parameter and are defined as:

$$D_1 = \frac{R_e}{R_e \pm R}$$

In the specific case of a straight line $R_e = \infty$, then $D_1 = 1$ and $D_2 = -1$. If $\epsilon \to 0$ is equivalent to maintain the kinematic relationship in (23) and therefore it ensures that the mobile robot follows the desired trajectory.

The angular displacement’s dynamic of each wheel is given by:

$$H_i \ddot{\psi}_i + C_i \dot{\psi}_i + F_i = \tau_{ri}$$

Where $H_i$ is the inertia matrix, $C_i$ is the Coriolis matrix, $F_i$ represents friction forces and $\tau_{ri}$ is the torque for each wheel. The control can be defined as follows where the friction effects and the Coriolis matrix are not considered because it is assume a mass center at the middle of the wheel axis and there is no slipping between the wheels and the floor.

$$\tau_{ri} = H_i (\dot{\psi}_i - K_{di} \dot{s}_i - K_{pi} s_i)$$

Where $K_{pi}$, $K_{di}$ are positive gains, $s_i$ and $\dot{s}_i$ are the coupled synchronization errors and are defined as:

$$s_i = \psi_i - \psi_{ri}$$

$$\dot{s}_i = \dot{\psi}_i - \dot{\psi}_{ri}$$

The terms $\psi_{ri}$ and $\dot{\psi}_{ri}$ that appear in (28) beside with $\dot{\psi}_{ri}$ in the eq. (27) are the nominal references in which the desired path is based, Slotine and Li [1987], and they are defined as follows:

$$\psi_{ri} = \psi_i^d + \beta_a \int_0^t \epsilon (\varpi) \, d\varpi; \quad \dot{\psi}_{ri} = \dot{\psi}_i^d + \beta_a \epsilon$$

$$\ddot{\psi}_{ri} = \ddot{\psi}_i^d + \beta_a \dot{\epsilon}$$

Where $\beta_a$ is a positive gain and $\epsilon$ is the wheel synchronization error defined in (24). With the controller shown in 27, the Lyapunov function to prove stability is designed as:

$$V_r = \frac{1}{2} \dot{s}_i^T p_1 \dot{s}_i + \frac{1}{2} \dot{s}_j^T p_3 \dot{s}_j$$

$$p_1 = p_2 K_p = 1$$

$$p_3 = K_d$$

$$\dot{V}_r = - (\dot{\epsilon}_j + \beta_a \epsilon)^T K_d (\dot{\epsilon}_j + \beta_a \epsilon)$$

$$- (\dot{\epsilon}_j + \beta_a \epsilon) (K_p - K_d)$$

$$\therefore K_p > K_d, \beta_a > (|\dot{\epsilon}_j|, |\dot{\epsilon}_j|)$$

The gain $K_p$ must be greater than the gain $K_d$ and $\beta_a$ must be greater than the absolute values of the position error and its derivative.

6. SIMULATIONS AND EXPERIMENTS

The robots used in the experimentation are four iRobot Create, the distance between its wheels are 0.26 [m] and the wheel’s radius is 0.035 [m]. This robots can be programed in C language.

At Cartesian level four robots are considered on the ellipse’s boundary, their initial coordinates are (2.64 [m], 0.70 [m]), (-0.39 [m], 1.48 [m]), (-2.64 [m], -0.70 [m]) and (0.39 [m], -1.48 [m]) respectively. The formation changes with respect to time due to the next equations:

$$a = a_0 + (a_f - a_0) \left( \frac{t}{t + e^{1-t}} \right)$$

$$b = b_0 + (b_f - b_0) \left( \frac{t}{t + e^{1-t}} \right)$$

Where $a_0, b_0$ are the initial values of the ellipse’s radices (3 [m] and 1.5 [m] respectively) and $a_f, b_f$ are the final values (1.5 [m] and 3 [m] respectively). The values used for the control gains are those reported in Sun and Wang [2007] and Nijmeijer and Rodriguez-Angeles [2004] $\beta_c = 0.5$, $K_{ri} = 100$, $K_i = 1$, $\Lambda = 20$, $\beta_a = 500$, $K_{pi} = 1000$ and $K_{di} = 100$, the simulation time is 20 [sec] with a sample period of 0.005 [sec] and the mobile robot’s parameters are: $r_w = 0.035$ [m], $R = 0.13$ [m] and $H = diag[0.50, 0.52][kg \cdot m^2]$. In the figure 5 can be appreciate the path of each robot to reach its desired position on the final ellipse.

In the graphics 7 and 9 the formation synchronization error and the wheel synchronization error converge to zero, on the other hand the Cartesian position error and the angular position error (figures 6 and 8) converge at some
values close to zero.

Fig. 6. Position error at Cartesian level ($e_i$).

Fig. 7. Synchronization error at Cartesian level ($\varepsilon_i$).

Fig. 8. Position error at wheel level ($e_{wi}$).

The experimental results are shown in the figure 10, where $R_1$, $R_2$, $R_3$ and $R_4$ are the initial positions for each robot. The table 1 shows the measured distance for each robot.

<table>
<thead>
<tr>
<th>Robot</th>
<th>Simulation distance [m]</th>
<th>experiment distance [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1.1947</td>
<td>1.21</td>
</tr>
<tr>
<td>II</td>
<td>1.1947</td>
<td>1.205</td>
</tr>
<tr>
<td>III</td>
<td>1.1947</td>
<td>1.215</td>
</tr>
<tr>
<td>IV</td>
<td>1.1947</td>
<td>1.205</td>
</tr>
</tbody>
</table>

Table 1. Robots at 15, 105, 195 and 285 degrees.

The experimental results are shown in the figure 10, where $R_1$, $R_2$, $R_3$ and $R_4$ are the initial positions for each robot. The table 1 shows the measured distance for each robot.

The experimental results are shown in the figure 10, where $R_1$, $R_2$, $R_3$ and $R_4$ are the initial positions for each robot. The table 1 shows the measured distance for each robot.

The table 1 shows the measured distance for each robot.

Fig. 9. Synchronization error at wheel level ($\epsilon$).

Fig. 10. Experimentation of a changing shape.

The figure 5 shows the trajectories followed by the robots to maintain on the ellipse’s boundary while it changes its shape but always centered at the origin, now it is proposed a method to move the formation outside the origin. The center of the formation will change respect to time while the ellipse keeps its shape. The equations that allows the formation’s change are:

$$O_x = (O_{xf})\left(\frac{t}{t + e(1-t)}\right)$$

$$O_y = (O_{yf})\left(\frac{t}{t + e(1-t)}\right)$$

It is assumed that the formation will start to move from the origin, then in similar manner to equation (29) $O_{xf}, O_{yf}$ are the final coordinates formation’s center. In order to maintain the formation along the time the equation (6) is modified:

$$\alpha_i = \arctan\left(\frac{y_i - O_{yi}}{x_i - O_{xi}}\right)$$

This change in $\alpha_i$ has effects in the desired Cartesian coordinates that now should consider the change of formation’s center, then it can be calculated by:

$$q_i^d = A_i\left[\begin{array}{c}a \\ b\end{array}\right] + \begin{bmatrix}O_x \\ O_y\end{bmatrix}$$

For a final point at $(1\ [m], 0\ [m])$ and the initial conditions of the robots 1-4 on the ellipse are: $(1.34\ [m], 1.34\ [m]), (-1.34\ [m], 1.34\ [m]), (-1.34\ [m], -1.34\ [m]), (1.34\ [m], -1.34\ [m])$, the simulation results are presented in the figures.
11-15 where the synchronization error converge to zero, the Cartesian position error and the wheel position error converge close to zero.

Fig. 11. Trajectories for a ellipse with change of center.

Fig. 12. Position error at Cartesian level ($e_i$).

Fig. 13. Synchronization error at Cartesian level ($e_i$).

In the physical system the formation is shown in the figure 16, where $R_1$, $R_2$, $R_3$ and $R_4$ are the initial positions for each robot. The table 2 shows the measured displacements for each robot.

7. CONCLUSIONS

In this article it is shown how to join a Cartesian synchronization scheme level and other at wheel level by pseudo-inverse of the Jacobian matrix of the mobile robot, resulting in a method that allows formation changing on the plane that generates a desired path for each robot in the swarm which is tracked by synchronizing each mobile wheels. The synchronization scheme maintains Cartesian robots formation and the control synchronization at wheel.

Table 2. Robots at 45, 135, 225 and 315 degrees.
level modifies the orientation angle and translational velocity of each mobile robot to force tracking of the desired path through synchronization or coupling of the wheels. The simulation results and the experiments show that it is feasible to use the union of the synchronization schemes to maintain formations with differential mobile robots. The experiments were done in speed mode with four iRobot® with a velocity profile pre-loaded. As future work this scheme will be implemented in real time with three differential mobile robots in torque mode.

REFERENCES


