Single Neuron Proportional-Derivative trained with Regularized Extended Kalman Filter for Leader-Follower Formation Control

Carlos Lopez-Franco[1], Ingrid Palomares[2], Nancy Arana-Daniel[1], Jesus Hernandez-Barragan[2], and Carlos Villaseñor[2]

[1] Departamento de Ciencias Computacionales, Centro Universitario de Ciencias Exactas e Ingenierías, Universidad de Guadalajara, Blvd. Marcelino García Barragán 1421, Guadalajara C.P. 44430, Jalisco, Mexico;
[2] Departamento de Innovación Basada en la Información y el Conocimiento, Centro Universitario de Ciencias Exactas e Ingenierías, Universidad de Guadalajara, Blvd. Marcelino García Barragán 1421, Guadalajara C.P. 44430, Jalisco, Mexico;

Abstract. This work presents a single neuron proportional-derivative (SNPD) control strategy for leader-follower formation control trained with the Extended Kalman Filter (EKF). SNPD trained with unregularized EKF may adapt its weights without limits, which is most significant over time which drives the formation control to instability. To address this issue, we propose to include a regularization strategy in the EKF. Since pose estimation with wheel odometry lacks precision, in this work we propose a pose estimation based on visual sensors and apriltags to improve precision with robustness against external perturbations. The effectiveness and performance of the implementations are demonstrated through real-world experiments using Turtlebot3 Waffle PI robots.

Keywords: Extended Kalman Filter · Leader-Follower Formation Control · Turtlebot3 Waffle PI · Adaptive PD control · Weights regularization.

1 Introduction

Collaborative and cooperative robotic systems is crucial for solving problems related to surveillance, exploration, recognition, environmental monitoring, and cooperative manipulation [1–3]. This methodology is known as networked robot systems. The technique that ensures cooperation between robots is called cooperative control. Formation control for multiple agents is considered a special case of cooperative control [2]. The objective of formation control is to maintain a certain pattern or shape during the execution of the task. This pattern defines a formation, which is essentially the relative distances between robots [4]. There is a wide variety of strategies for formation control, in the Leader-Follower scheme, one of the robots is assigned as the leader, while the other robots are followers [3]. The control law is applied to each follower independently of the

https://doi.org/10.61728/AE20255091

leader. The purpose is for the follower robot to maintain a formation position relative to the leader.

Currently, the presence and importance of artificial intelligence cannot be ignored. It has enabled rapid progress in society and continues to advance in key areas like industry, education, science, and entertainment [5]. Neural PID controllers propose methodologies to improve the performance of classic PID controllers. Adaptive PID techniques trained with gradients have been introduced [6–8]. Although it has been shown that PD and PID controllers trained with the Extended Kalman Filter show superiority in learning rates and faster convergence times [9, 10]. However, the weight adaptations may grow without limits, in some cases, significant weight values might drive the systems to instability.

To improve the SNDP, we proposed to include a regularization strategy in the EKF to prevent weights grow without limits. Moreover, a pose estimation based on visual sensors and apriltags is proposed to avoid the use of wheel odometry. Since wheel odometry lacks precision, visual sensors and apriltags will provide better precision and robust pose estimation.

The paper is organized as follows: Section 2 presents the Leader-Follower kinematics and control. Section 3 describes the design of the SNPD controller trained with the regularized EKF. Section 4 introduces the description of the proposed approach, which is the SNPD control for Leader-Follower formation and the pose estimation based on visual sensors and apriltags. Section 5 presents the real-world experimental results. Finally, the conclusions are presented in Section 6.

2 Formation Kinematics

The leader-follower system considered in this work is presented in Fig. 1, which is composed of a leader i and a follower j. The linear (v_i, v_j) and angular (ω_i, ω_j) velocities are expressed respect to the local frames $\{i, j\}$. The position and orientation of the leader are (x_i, y_i) and θ_i, in the world frame respectively. Similarly, the position and orientation of the follower are (x_j, y_j) and θ_j. Finally, the position (x_{ij}, y_{ij}) and orientation $\theta_{ij} = \theta_j - \theta_i$ are related to the leader frame $\{i\}$. The position (x_{ij}, y_{ij}) can be computed as

$$\begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \left(\begin{bmatrix} x_j \\ y_j \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \end{bmatrix} \right) \tag{1}$$

To deal with non-holonomic constraints, position (x_{ij}^h, y_{ij}^h) at distance d_j is defined for control purposes. It can be calculated as

$$\begin{bmatrix} x_{ij}^h \\ y_{ij}^h \end{bmatrix} = \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} + d_j \begin{bmatrix} \cos(\theta_{ij}) \\ \sin(\theta_{ij}) \end{bmatrix} \tag{2}$$

Differentiating (2) respect to time, the following formation kinematics can be obtained [11]

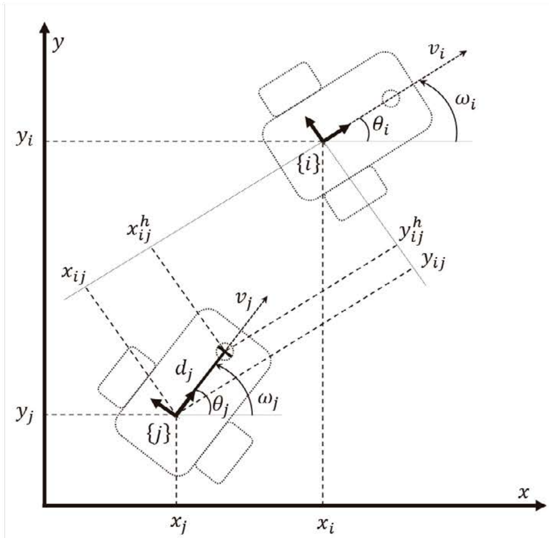

Fig. 1: The leader-follower setup based on relative Cartesian coordinates.

$$\begin{bmatrix} \dot{x}^h_{ij} \\ \dot{y}^h_{ij} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{ij}) & -d_j \sin(\theta_{ij}) \\ \sin(\theta_{ij}) & d_j \cos(\theta_{ij}) \end{bmatrix} \begin{bmatrix} v_j \\ w_j \end{bmatrix} + \begin{bmatrix} -v_i \\ 0 \end{bmatrix} + \omega_i \begin{bmatrix} y^h_{ij} \\ -x^h_{ij} \end{bmatrix} \tag{3}$$

The leader-follower control scheme is designed with a control input (u_x, u_y) that drives the follower to a related position with respect to the leader. Considering $\dot{x}^h_{ij} = u_x$ and $\dot{y}^h_{ij} = u_y$ into (3) yields to

$$\begin{bmatrix} v_j \\ \omega_j \end{bmatrix} = \begin{bmatrix} \cos(\theta_{ij}) & \sin(\theta_{ij}) \\ -\dfrac{1}{d_j} \sin(\theta_{ij}) & \dfrac{1}{d_j} \cos(\theta_{ij}) \end{bmatrix} \left(\begin{bmatrix} u_x \\ u_y \end{bmatrix} + \begin{bmatrix} v_i \\ 0 \end{bmatrix} - \omega_i \begin{bmatrix} y^h_{ij} \\ -x^h_{ij} \end{bmatrix} \right) \tag{4}$$

In this work, (u_x, u_y) will be computed with adaptive neuron PD controllers trained with EKF. Detailed descriptions of leader-follower kinematics can be found in [11].

3 Adaptive single neuron PD controller

The adaptive single neuron PD controller (SNPD) is illustrated in Fig. 2. The error e defines the difference between the reference y_r and the system output y in step time k, see (5). The inputs x_1 and x_2 represents the proportional (6) and the derivative (7) errors [12]. The weights w_1 and w_2 represent the proportional gain, and the derivative gain, respectively. These gains will be adapted online based on the EKF algorithm [10]. The output u is computed as the weighted sum of the inputs of the neuron (8).

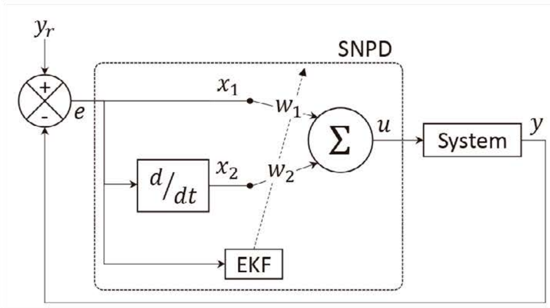

Fig. 2: Adaptive single neuron PD controller (SNPD).

$$e(k) = y_r(k) - y(k) \tag{5}$$
$$x_1(k) = e(k) \tag{6}$$
$$x_2(k) = e(k) - e(k-1) \tag{7}$$
$$u(k) = w_1(k)\, x_1(k) + w_2(k)\, x_2(k) \tag{8}$$

The EKF algorithm adjusts the weights $\mathbf{w}(k) = \left[w_1(k)\ w_2(k)\right]^T$ for the single neuron using online training. The EKF settings are: $\mathbf{K} \in \mathbb{R}^{2\times1}$ is the Kalman gain vector, $\mathbf{P} \in \mathbb{R}^{2\times2}$, $\mathbf{Q} \in \mathbb{R}^{2\times2}$, and $\mathbf{R} \in \mathbb{R}^{1\times1}$ are covariance matrices of weight estimation error, estimation noise, and error noise, respectively. Moreover, $\eta \in \mathbb{R}$ is the Kalman filter learning rate, and $\mathbf{H} \in \mathbb{R}^{2\times1}$ is a matrix whose entries are derivative of the neuron output with respect to each weight. Finally, the error $e \in \mathbb{R}$ is given by (5), and λ is a regularization gain.

$$\mathbf{K}(k) = \mathbf{P}(k)\,\mathbf{H}(k)\left[\mathbf{R}(k) + \mathbf{H}^\top(k)\,\mathbf{P}(k)\,\mathbf{H}(k)\right]^{-1} \tag{9}$$
$$\mathbf{w}(k+1) = (1-\lambda)\,\mathbf{w}(k) + \eta\mathbf{K}(k)\,e(k) \tag{10}$$
$$\mathbf{P}(k+1) = \mathbf{P}(k) - \mathbf{K}(k)\,\mathbf{H}^\top(k)\,\mathbf{P}(k) + \mathbf{Q}(k) \tag{11}$$

The matrix $\mathbf{H}$ is defined as

$$\mathbf{H}(k) = \begin{bmatrix} \dfrac{\partial u(k)}{\partial w_1(k)} & \dfrac{\partial u(k)}{\partial w_2(k)} \end{bmatrix}^T = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \tag{12}$$

During neuron training, the weights $w_1(k)$ and $w_2(k)$ could be adjusted with negative values; otherwise, they may grow without bound. It is important to consider $w_1(k) := \max(0, w_1(k))$ and $w_2(k) := \max(0, w_2(k))$. Moreover, by tuning λ, it helps prevent that weights grow without limit.

More detailed information about adaptive single neuron PD and PID controllers trained with the EKF are given in [9, 10].

4 Description of the proposed approach

In order to perform the leader-follower control scheme (4), we propose to design u_x and u_y, based on two SNPD schemes. Let us define the formation errors $e^x_{ij} = x^*_{ij} - x^h_{ij}$ and $e^y_{ij} = y^*_{ij} - y^h_{ij}$, where (x^*_{ij}, y^*_{ij}) is the reference of the formation scheme. In other words, $u_x := \mathrm{SNPID}(e^x_{ij})$ and $u_y := \mathrm{SNPID}(e^y_{ij})$.

In this work, we estimate the pose of each robot based on visual markers called AprilTags, see Fig 3. Leader and follower are associated with a marker that provides a relative pose estimation and an ID to difference one of each other. The matrices ${}^c\mathbf{T}_i$ and ${}^c\mathbf{T}_j$ are the pose estimation by AprilTags of the leader and the follower, respectively. Both matrices are expressed with homogeneous transformations measured with respect to the camera frame $\{c\}$. To obtain the pose estimation with respect to the world frame $\{w\}$, we compute

$$ {}^w\mathbf{T}_i = {}^w\mathbf{T}_c {}^c\mathbf{T}_i \tag{13} $$
$$ {}^w\mathbf{T}_j = {}^w\mathbf{T}_c {}^c\mathbf{T}_j \tag{14} $$

where ${}^w\mathbf{T}_i$ and ${}^w\mathbf{T}_j$ are the homogeneous matrices of the leader and follower, respectively, in the world frame. Moreover, ${}^w\mathbf{T}_c$ is a matrix transformation that describes the position and orientation of the camera with respect to the world frame.

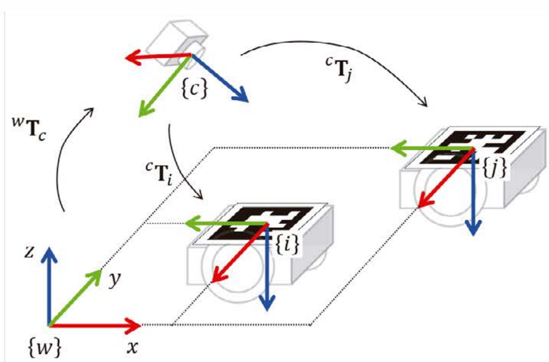

Fig. 3: Mobile robot pose estimation based on AprilTags.

Intrinsic camera calibration is required previously to the mobile robot pose estimation. Moreover, the matrix ${}^w\mathbf{T}_c$ should also be previously known. The positions (x_i, y_i) and (x_j, y_j) are available in the position vector of ${}^{w}\mathbf{T}_i$ and ${}^{w}\mathbf{T}_j$, respectively. However, orientations θ_i and θ_j should be computed based on the representation of Euler angles of the rotation matrices in ${}^{w}\mathbf{T}_i$ and ${}^{w}\mathbf{T}_j$.

5 Experimental Results

Fig. 4: Real-world setup for experimentation.

The applicability of the proposed approach is demonstrated through real-world experiments. All parameter settings for experimentation are given ahead. The reference for the formation control are: $x^*_{ij} = -0.6$ $y^*_{ij} = 0.0$. The Kalman settings were set to: $\mathbf{R} = 0.001$, $\eta = 0.1$, $\lambda = 0.001$, and

$$\mathbf{P} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}.$$

Moreover, the intrinsic parameters of the camera were calibrated as $f = 623.71$, $C_u = 317.97864$, $C_v = 257.93234$. Finally

$${}^{w}\mathbf{T}_c = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

For the experimentation, two Turtlebot3® Waffle PI robots were used. The Turtlebot3® is a ROS-based differential-drive robot for education and research. The ROS packages provide access to send velocity control inputs (v, w) and read its odometry. However, we used the proposed camera pose estimation rather than odometry measures. The real-world setup for experimentation is presented in

Fig. 4. The leader-follower control scheme is centralized in an external computer running in MATLAB® with the ROS toolbox.

It is important to mention that the linear and angular velocities are limited before being applied to the Turtlebot3® to prevent damage or wear to the actuators. The limits for linear velocity are $-0.2 \leq v \leq 0.2$ and for the angular velocity $-0.9 \leq w \leq 0.9$.

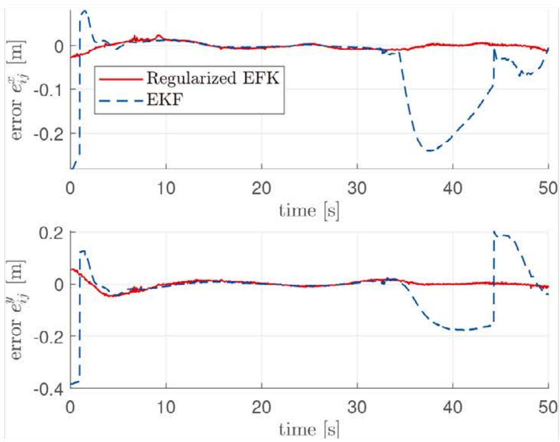

Fig. 5: Formation errors (e_{ij}^x, e_{ij}^y) over time.

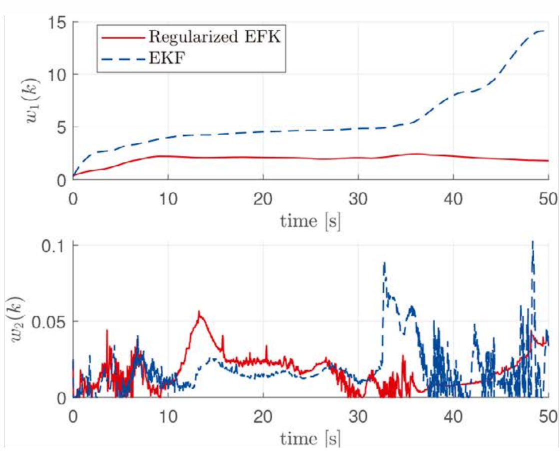

Fig. 6: SNPD weights adaptation over time.

Fig. 5 shows the formation errors (e_{ij}^x, e_{ij}^y) over 50 seconds. We notice that around 35 seconds, the leader-follower formation task fails in the classical SNPD trained with unregularized EKF since errors become significant. In contrast, using the proposed SNPD with regularized EKF, errors were kept near zero keeping the desired formation. The SNPD wights adaptation is shown in Fig. 6, where w_1 corresponds to the proportional gain and w_2 to the derivative gain.

The gain w_1 in unregularized EKF grows without limits over time, which provokes the formation task to fail. In contrast, the proposed regularized EKF prevents that gain w_1 grows more than necessary. Moreover, both unregularized and regularized EFK adapts the derivative gain w_2 similarly.

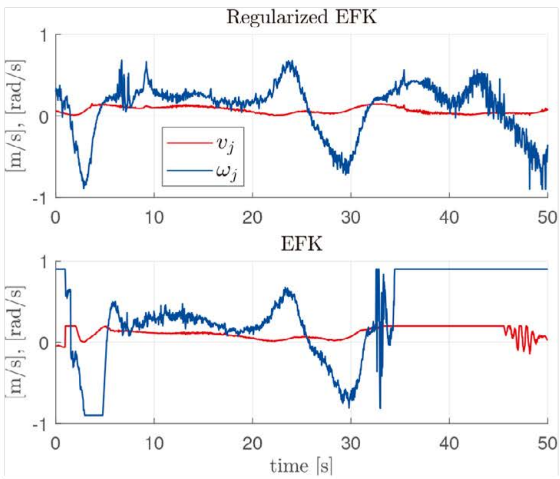

Fig. 7: Follower control input signals (v_j, ω_j) over time.

Finally, follower control signals (v_j, ω_j) during the formation task are given in Fig. 7. As expected, control signals become saturated around 35 seconds, when the proportional gain w_1 becomes significantly big.

6 Conclusions

This work presented a single neuron PD controller for leader-follower formation control trained with a regularized EKF algorithm. Experimental results demonstrate that PD weight grows without limit over time if the EKF weight adaptation is not regularized, which could cause issues in the calculation of the control actions u_x and u_y, leading to decreased performance in the formation task. The weight regularization strategy provides better performance, the weights adapt only the necessary to keep the error near zero. Moreover, this paper also considers the use of visual sensors to estimate leader and follower poses, rather than wheel odometry. Mobile robots were equipped with AprilTags onboard to measure the position and orientation with the advantage of obtaining a pose estimation in the face of external disturbances, such as the robot's tires getting stuck. We let as future work to use the proposed SNPD for consensus-based formation control.

References

1. A. Sanfeliu, N. Hagita, and A. Saffiotti, "Robotics and autonomous systems," *Robotics and Autonomous Systems*, vol. 56, no. 10, pp. 793–797, 2008.
2. M. A. Kamel, X. Yu, and Y. Zhang, "Formation control and coordination of multiple unmanned ground vehicles in normal and faulty situations: A review," *Annual Reviews in Control*, vol. 49, pp. 128–144, 2020.
3. H. Wang, D. Guo, X. Liang, W. Chen, G. Hu, and K. K. Leang, "Adaptive vision-based leader-follower formation control of mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, 2016.
4. K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.
5. J. J. Bryson, "The past decade and future of ai's impact on society," BBVA, Tech. Rep., 2019, obtenido de Open Mind BBVA.
6. R. Hernandez-Alvarado, L. G. Garcia-Valdovinos, T. Salgado-Jimenez, A. Gómez-Espinosa, and F. Fonseca-Navarro, "Neural network-based self-tuning pid control for underwater vehicles," *Sensors*, 2016.
7. W. Tang, L. Wang, J. Gu, and Y. Gu, "Single neural adaptive pid control for small uav micro-turbojet engine," *Sensors*, 2020.
8. J. Zhong, Y. Zhu, C. Zhao, Z. Han, and X. Zhang, "Position tracking of a pneumatic-muscle-driven rehabilitation robot by a single neuron tuned pid controller," *Complexity*, 2020.
9. J. Hernandez-Barragan, J. D. Rios, A. Y. Alanis, C. Lopez-Franco, J. Gomez-Avila, and N. Arana-Daniel, "Adaptive single neuron anti-windup pid controller based on the extended kalman filter algorithm," *MDPI Electronics*, p. 636, 2020.
10. J. Hernandez-Barragan, J. D. Rios, J. Gomez-Avila, N. Arana-Daniel, C. Lopez-Franco, and A. Y. Alanis, "Adaptive neural pd controllers for mobile manipulator trajectory tracking," *PeerJ Computer Science*, vol. 7, p. e393, 2021.
11. X. Liang, H. Wang, Y.-H. Liu, Z. Liu, and W. Chen, "Leader-following formation control of nonholonomic mobile robots with velocity observers," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 4, pp. 1747–1755, 2020.
12. M. Moradi, M. Katebi, and M. Johnson, "Predictive pid control: a new algorithm," in *IECON'01. 27th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.37243)*, vol. 1, 2001, pp. 764–769 vol.1.