

Imitation Learning-based Visual Servoing for Tracking Moving Objects

Rocco Felici¹, Matteo Saveriano², Loris Roveda³, and Antonio Paolillo³

¹ Università della Svizzera italiana (USI), Lugano, Switzerland
`rocco.felici@usi.ch`,

² Department of Industrial Engineering (DII),
University of Trento, Trento, Italy

³ Dalle Molle Institute for Artificial Intelligence (IDSIA),
USI-SUPSI, Lugano, Switzerland

Abstract. In everyday life collaboration tasks between human operators and robots, the former necessitate simple ways for programming new skills, the latter have to show adaptive capabilities to cope with environmental changes. The joint use of visual servoing and imitation learning allows us to pursue the objective of realizing friendly robotic interfaces that (i) are able to adapt to the environment thanks to the use of visual perception and (ii) avoid explicit programming thanks to the emulation of previous demonstrations. This work aims to exploit imitation learning for the visual servoing paradigm to address the specific problem of tracking moving objects. In particular, we show that it is possible to infer from data the compensation term required for realizing the tracking controller, avoiding the explicit implementation of estimators or observers. The effectiveness of the proposed method has been validated through simulations with a robotic manipulator.

Keywords: Visual Servoing, Imitation Learning, Visual Tracking

1 Introduction

Today robots are not merely asked to execute tasks in controlled environments, but they must have friendly interfaces so that everyone can conveniently operate them in everyday life. In fact, given their high level of ubiquity, more and more robots are at the disposal of people with no technical expertise. As a consequence, easy control frameworks that do not require specific engineering or programming skills are urgently needed. Furthermore, modern robots operating “in the wild” need to be highly adaptive, to cope with changes of dynamic environments.

Imitation Learning (IL) [29], also known as programming by demonstrations [4] or learning from demonstrations [3], promises to avoid specific coding duties by imitating the desired behavior as performed by an expert [5]. With respect to classic control paradigms, IL is easier and more convenient for non-expert operators, as they only need to provide demonstrations of the desired robotic tasks. Among the IL approaches, Dynamical System (DS)-based

methods [15,27,28] allow realizing the imitation strategy while ensuring stability properties. Adaptive capabilities, instead, can be realized by including exteroceptive sensing, such as vision, into the IL strategy. In particular, recent work [23,24,30] have explored the possibility to combine Visual Servoing (VS) [7,8] with DS-based IL. We name such integration Imitation Learning for Visual Servoing (ILVS). Such combination brings benefit to both techniques: on the one side, the visual perception adds adaptability to the IL scheme to cope with environmental changes; on the other, the imitation strategy allows the addition of tasks or constraints to the VS law with no specific implementation.

This work aims at resorting to the ILVS paradigm to tackle the specific problem of tracking moving objects. Traditional tracking techniques need to estimate the motion of the target, e.g., specifically implementing a Kalman filter [6] or predictive controllers [13]. Instead, we provide a framework that leverages ILVS and extrapolates from demonstrations of tracking experiments the required information for adding the tracking skill to the basic VS law. In particular, we propose to use the so-called Reshaped Dynamical System (RDS) approach [28] to imitate the tracking behavior into the basic VS control. The resulting learning-aided control system has been validated with robotic simulations.

2 Background

The well-known VS technique [7,8] employs vision to control the motion of a robot. In particular, in image-based VS, considered in this work, the objective is to zero the difference between desired and measured visual features that are directly defined on the camera image. Such visual features represent the feedback of the controller that computes camera velocities to achieve a desired task; they can be detected with standard image processing [16] or more sophisticated methods, e.g., artificial neural network [22]. Assuming an eye-in-hand configuration, a static target, and constant desired features, the basic VS law computes the camera velocity $\mathbf{v} \in \mathbb{R}^6$ with a simple reactive controller. Its objective is to nullify the visual error $\mathbf{e} \in \mathbb{R}^k$ between the detected and desired visual features:

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}^+ \mathbf{e}, \quad (1)$$

where λ is a positive scalar gain and $\widehat{\mathbf{L}}^+ \in \mathbb{R}^{6 \times k}$ an approximation of the Moore-Penrose pseudoinverse of the interaction matrix [7]. Such approximation is normally due to unknown information, such as the depth of the visual features⁴. The simple law (1) can be augmented with other tasks or constraints to enable additional skills, by employing planning techniques [10,17], predictive controllers [2,20,21,26], and other sort of optimization-based frameworks [1,18,19]. However, such approaches require careful design and implementation of the additional modules, which is desirable to avoid for the sake of easiness of use.

⁴ To keep the notation compact, we omit the dependence of the interaction matrix on the visual features and their depth.

To this end, inspired by the DS paradigm, it has been proposed to augment the skills of the basic law with an ILVS strategy [24]. In particular, by using the specific RDS method [28], one could write the augmented VS law as follows:

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}^+ \mathbf{e} + h \boldsymbol{\rho}(\mathbf{e}), \quad (2)$$

where $\boldsymbol{\rho}(\mathbf{e})$ is an error-dependent corrective input used to follow complex trajectories and h is a vanishing term used to suppress $\boldsymbol{\rho}$ after a user-defined time and retrieve stability. Such an approach can be used to generate complex visual trajectories, e.g., to avoid collisions, as done in [24]. In this work, instead, we use this formulation to enable the learned compensation terms needed to achieve the tracking of moving objects, as explained in the next section.

3 Method

3.1 Problem definition

The aim of our work is to enable visual tracking of moving targets avoiding explicit programming of the required additional components of the basic law (1).

Assuming a moving target, the VS law has to account for such motion [8]:

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}^+ \mathbf{e} - \widehat{\mathbf{L}}^+ \frac{\partial \mathbf{e}}{\partial t}, \quad (3)$$

where the second term on the right of the equation actually acts as a feedforward term to compensate for the error's time variation due to the target motion [8]. Ad hoc techniques can be implemented to estimate the term due to the motion of the target so that it can be inserted in (3) and compensated, e.g., with the introduction of integrators [9], feedforward terms [6,12] or filters [13,31].

In this work, instead, our aim is to rely on an imitation strategy to infer the compensation term of the law (3) from previous demonstrations of tracking experiments. In particular, inspired by DS-based approaches as in (2), we treat the reshaping term $\boldsymbol{\rho}$, to be learnt from data, as the compensation term in (3):

$$\boldsymbol{\rho} = -\widehat{\mathbf{L}}^+ \frac{\partial \mathbf{e}}{\partial t}. \quad (4)$$

Therefore, our problem can be formulated as follows: learn from previous demonstrations an estimate of the compensation term $\hat{\boldsymbol{\rho}}$ so that the VS law

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}^+ \mathbf{e} + \hat{\boldsymbol{\rho}}(\mathbf{e}) \quad (5)$$

realizes tracking of moving objects. It is worth mentioning that (5) is formally the same as (2). However, the vanishing term h is not used in (5) since the estimate $\hat{\boldsymbol{\rho}}$ has to be always active to perform the tracking skill.

3.2 Dataset

We assume that an “oracle” is available to provide a few demonstrations of the full desired tracking behavior. A possible oracle could be a human user, who can kinesthetically teach the robot the tracking motion, or an ideal controller in simulated environments, where all the required information is perfectly known.

During the oracle’s executions, data describing how the task is carried out are recorded for each timestamp. In particular, we log the evolution of the visual error, as measured on the camera image, and the corresponding velocities, as shown to the camera in order to achieve the full desired task:

$$\mathcal{D} = \{\mathbf{e}_n^d, \mathbf{v}_n^d\}_{n=1, d=1}^{N, D}, \quad (6)$$

where N is the number of samples and D the number of demonstrations. This dataset serves as the basis for the actual training set \mathcal{T} that is built as follows:

$$\mathcal{T} = \{\boldsymbol{\varepsilon}_n^d, \boldsymbol{\rho}_n^d\}_{n=1, d=1}^{N, D}, \quad (7)$$

considering that $\boldsymbol{\varepsilon}_n^d = \widehat{\mathbf{L}}^+ \mathbf{e}_n^d$ and $\boldsymbol{\rho}_n^d = \mathbf{v}_n^d + \lambda \widehat{\mathbf{L}}^+ \mathbf{e}_n^d$. Note that for all the demonstrations we consider that the value of the control gain λ does not change, as well as the value of the approximated inverse of the interaction matrix $\widehat{\mathbf{L}}^+$ is assumed to be constant and equal to its value at convergence.

3.3 Learning the compensation term

Given the training dataset (7), an estimate of the compensating term can be conveniently retrieved from vision data using any regression function \mathbf{r} . In particular, we train a Gaussian Mixture Model (GMM) on \mathcal{T} to estimate the velocity term needed to compensate for the motion of the target object. Therefore, Gaussian Mixture Regression (GMR) is used to retrieve a smooth estimate of $\boldsymbol{\rho}$, namely $\hat{\boldsymbol{\rho}}$. The GMR takes as input the current value of $\boldsymbol{\varepsilon}$ and provides $\hat{\boldsymbol{\rho}}$ as

$$\hat{\boldsymbol{\rho}} = \mathbf{r}_{\text{GMR}}(\boldsymbol{\varepsilon} \mid \mathcal{T}). \quad (8)$$

Therefore, the compensation term is online estimated using (8) and inserted in the control law (5) to achieve the tracking of moving objects.

4 Results

4.1 Validation setup

To validate our framework we consider a robotic experiment with the robot manipulator Franka Emika [14], which has 7 joints and an Intel RealSense D435i sensor (used as a monocular camera) mounted on the end-effector. The sensor has a field of view of $69^\circ \times 42^\circ$ and a frame resolution of 1920×1080 pixel. The robot and the environment for the experiments are simulated in CoppeliaSim [11], as

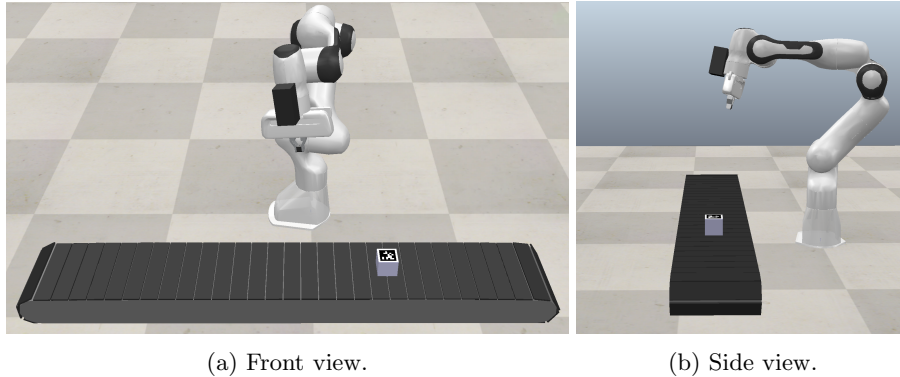


Fig. 1: Validation setup: the Franka Emika robot manipulator in the CoppeliaSim environment has to reach a box moving on a conveyor belt.

shown in Fig. 1. The goal of the experiment is to allow the robot to reach a box that moves at a constant velocity on a conveyor belt. In other terms, we set the desired features so that at convergence the robot centers the box on the image plane. The box is marked with an AprilTag marker, whose corners provide the visual features for the VS law. In particular, we use the 4 corner points of the marker as visual features (i.e., $k = 8$). As classically done in VS, 4 points are enough to ensure robust visual feedback. At the start of the experiments, the conveyor belt accelerates from zero to 0.1 m/s and keeps the velocity constant for the rest of the experiment. The implementation of the framework has been done in Python 2.7 language within the ROS [25] infrastructure.

The oracle used to collect the demonstrations consists of an ideal VS controller provided with complete knowledge of the dynamics of the target, available in the simulated environment. In practice, we use the law (3) with $\lambda = 2$, and the compensation term is built from the perfect knowledge of the box velocity. The interaction matrix has been approximated by using the value of the visual features depth at the target, which is 0.09116 m. In total, we have collected three demonstrations of the task. If not otherwise mentioned, the same value of the gain and the same approximation of the interaction matrix are kept for the online experiments. It is worth mentioning that other teaching methodologies could be used, such as kinesthetic teaching or teleoperation. Our choice was dictated by the need for high precision in tracking the object: a tracking controller with complete knowledge, as available in simulation, provides way better performances for precise movements than human demonstration. Furthermore, human demonstrations usually require preprocessing of the trajectories to grant exact convergence to the target in the feature space. The regression is carried out using GMM with 11 components. The number of components has been set performing a grid search. At each iteration of the controller, the framework detects new visual features and computes the new value of ε , which is used by the GMR to compute an estimate $\hat{\rho}$ of the compensation term that is finally inserted

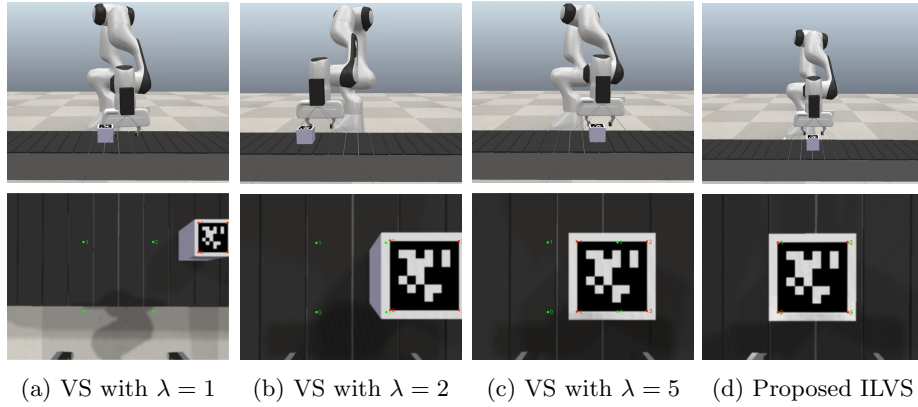


Fig. 2: Comparison between three versions of the standard VS controller and the proposed ILVS strategy.

in the control law as in (5). The camera velocity thus computed is sent to the kinematic control of the manipulator that transforms it into joint velocities to move the robot towards the desired tracking behavior. With this setup, multiple tests are carried out to evaluate firstly the learning and replication capabilities of the demonstrated target tracking tasks, and secondly, the system's ability to adapt to new scenarios and sudden changes in the environment.

In the presented plots of the experiments, the trajectories saved in the demonstrations are shown with black dotted lines, whereas the execution of our ILVS framework is in blue; red dots represent the starts of the demonstrated trajectories, while the red crosses are their ends.

The experiments are shown in the video accompanying the paper, available at the following link: <https://youtu.be/ORdAZDmCQsA>.

4.2 Comparison with the standard VS controller

The first set of experiments aims at comparing the behavior of standard VS without compensation term, as in (1), with different values of the gain λ , against our proposed ILVS strategy. The results of this comparison are shown in Fig. 2. As expected, even if the standard VS law manages to approach the box, due to its motion, it never manages to center it on the image plane. Indeed, a constant error between the current state of the features (denoted in red and numbered from zero to three in Fig. 2) and their desired position (in green) is kept at a steady state. Such error is lower by increasing the value of λ from 1 to 5, but cannot be nullified. It is indeed noteworthy that extremely high gain values cannot provide a reasonable solution to the tracking problem, since it would introduce instability in the control system [7,8]. Unlike the standard controllers, our ILVS manages to infer from data the required information to compensate for the box motion. As shown in Fig. 2d, ILVS provides the robot with the capability

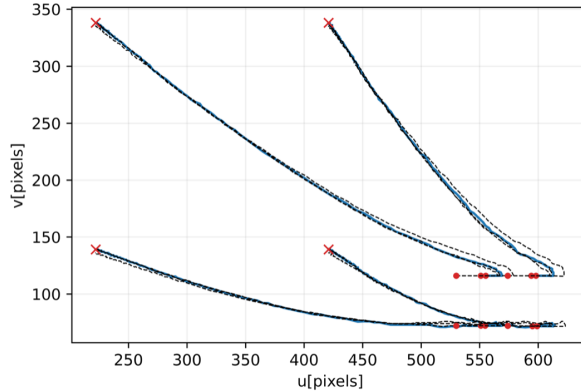


Fig. 3: ILVS experiment with the same initial condition as in the demonstration: visual features trajectories as in the demonstrations and executed by our method.

to approach the target, reach convergence, and keep the camera above the box at the desired pose for the duration of the experiment. Indeed, in this case, the measured visual features match their desired counterpart at steady-state.

Fig. 3 shows, for the same experiment, a qualitative evaluation of the trajectories of the visual features from the demonstrations (black dotted lines), and the trajectories executed by the ILVS strategy (in blue). One can observe the ability of the system to accurately replicate the demonstrated trajectories when starting from a known location (the same as the demonstrated ones).

The correspondent quantitative results of this experiment are presented in terms of average Root-Mean-Square Error (RMSE)⁵ and its standard deviation measuring the accuracy of the predicted camera position and velocity, and the predicted feature position w.r.t the corresponding quantities contained in the demonstrations. In particular, the average RMSE regarding the predicted visual features position is 22 ± 11 pixel. For the camera positions and the linear camera velocities, the obtained results are 33 ± 24 mm and 69 ± 71 mm/s, respectively.

4.3 Target tracking experiments with unseen initial conditions

The second set of experiments is carried out to test the adaptability of the system w.r.t. unseen initial conditions, i.e., when the starting orientation or the position of the camera is different from those demonstrated in the training dataset.

We tested the framework with incremental levels of difficulty. In the first experiment of this set, the initial conditions are analogous (but not identical as in the experiment shown in Fig. 2d and Fig. 3) to the ones in the training dataset. As illustrated in Fig. 4 (left), the starting point of the experiment in the image plane are in the nearby of the starting points (red dots) of the demonstrations

⁵ RMSE values rounded up to the nearest whole number.

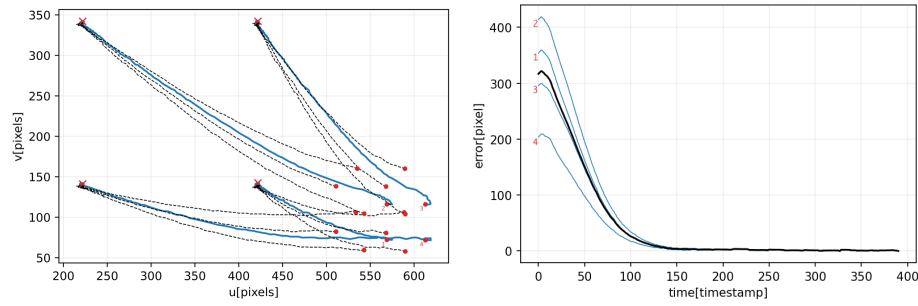


Fig. 4: ILVS experiment with similar initial conditions of the demonstrated ones: visual features trajectories (left) and visual error (right).

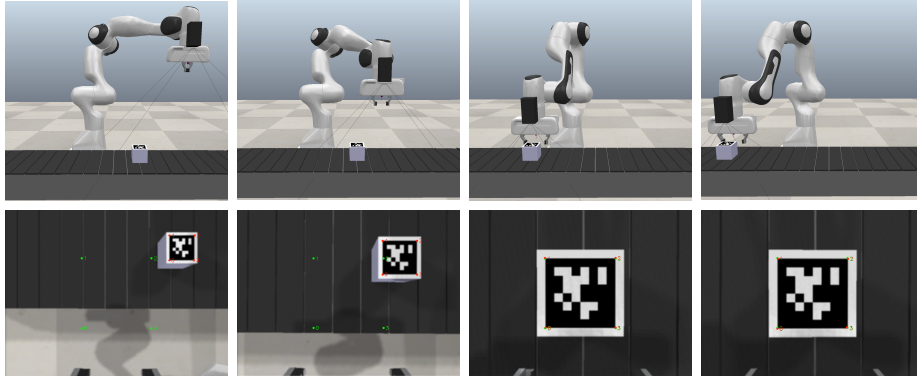


Fig. 5: Snapshots of the ILVS experiment with similar initial conditions of the demonstrations: robot's external views (top) and camera images (bottom).

(black dotted lines), since the initial position of the camera has been slightly moved away from the one in the demonstrations. The starting orientation of the camera is, instead, the same as the demonstrations. Given similar initial conditions, as expected, the system executes the task (blue lines in the plot) without any particular difficulties. Fig. 4 (right) shows the time evolution of the visual error for each of the four features (blue lines), which is kept to zero after a transient time for the duration of the experiment; it is also depicted the average visual error among all features (black line). Four snapshots of this experiment are presented in Fig. 5 showing the manipulator approaching the object and tracking the target moving on the conveyor belt during all its motion.

The second experiment of this set aims to evaluate the effectiveness of the approach in handling unseen conditions. In particular, at the beginning of the experiment, the camera is oriented as in the demonstrations but has a substantial difference in position. The large initial positional offset is well visible in the plot of Fig. 6, where the initial value of the visual features is far off from the

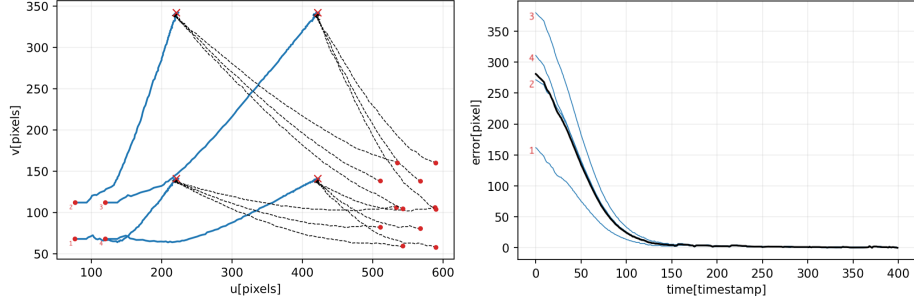


Fig. 6: ILVS experiment with unseen initial position and initial orientation as in the demonstrations: visual features trajectories (left) and visual error (right).

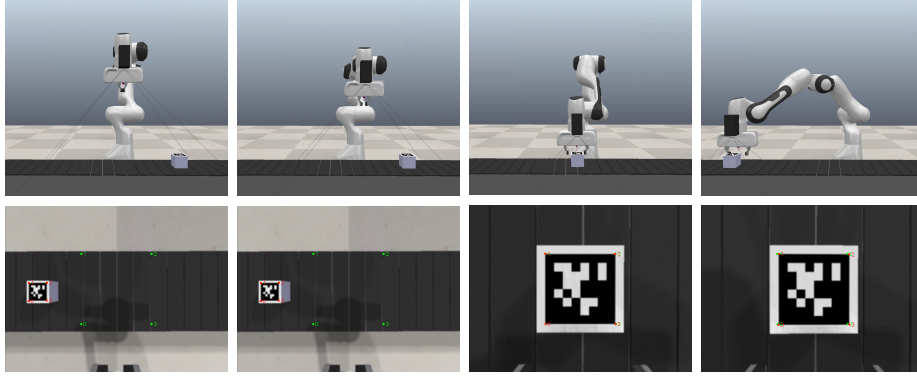


Fig. 7: Snapshots of the ILVS experiment with unseen initial position and demonstrated orientation: robot's external views (top) and camera images (bottom).

demonstration. Nevertheless, the visual features trajectories shown in Fig. 6 (left) demonstrate that the robot manage to successfully achieve the VS task, as the current value of the feature converges to their desired one, as also demonstrated in the dataset. Similarly, target tracking performance can be evaluated also from the time evolution of the visual error presented in Fig. 6 (right). From this plot, one can evaluate that the visual error is kept to zero after a transient time, even while the box continues moving on the conveyor belt. Four snapshots of this ILVS experiment can be evaluated in Fig. 7: the manipulator can reach the box and keep it tracking for all the experiments. The last two snapshots show how the robot manages to keep the box at the center of the image for the experiment, accommodating the motion induced by the conveyor belt.

The third experiment is meant to test at its greatest degree the handling of unseen initial conditions. As can be seen in Fig. 8 (left) from the position of the features in the image plane, the end-effector of the manipulator at the beginning of the experiment has a pose that is not present in the training data. Neverthe-

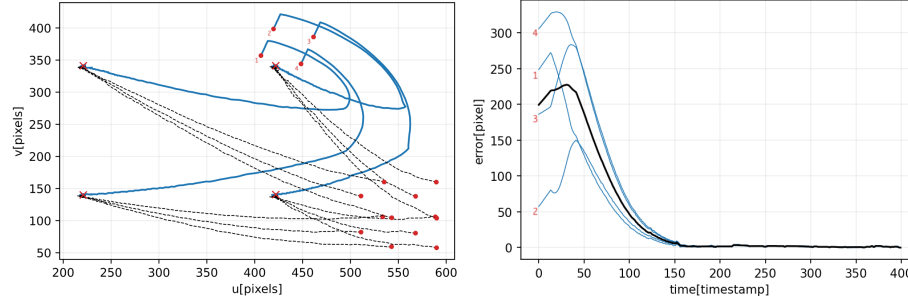


Fig. 8: ILVS experiment with unseen initial position and orientation: visual features trajectories (left) and visual error (right).

less, the robot still manages to adjust its movement to successfully approach the moving target ensuring convergence, and once reached, it is able to track the target along its motion (see also the snapshots of Fig. 9). For this experiment, we also show in Fig. 10 the plots of the camera velocities, as demonstrated (grey lines in the plots) and as executed by our method (in blue).

For these three experiments, we provide a quantitative evaluation of the tracking performances. In particular, we considered the phase of the experiments that starts when the visual error is lower than 5 pixels (cfr. Fig. 4 (right), Fig. 6 (right), and Fig. 8 (right)). For this portion of the experiments, the visual error is on average 1.795 ± 0.984 pixels, corresponding to 0.475 ± 0.257 mm of error in the camera position.

Finally, we perform one last test in which we suddenly move the target object during the execution of the experiment. We observed the system’s ability to adjust to such sudden and unexpected movements of the target object (tests were pursued with both low gain $\lambda = 2$ and high gain $\lambda = 10$ yielding satisfactory results in both cases). The results of this experiment can be evaluated from the accompanying video.

5 Discussion and conclusion

In this work, we have addressed some of the needs that arise from the introduction of friendly robots in domestic and industrial contexts where users are not necessarily experts. In these situations, adaptability and easiness of use are must-haves for robots. Therefore, we have proposed an imitation learning-based visual servoing framework for target tracking operations that avoids explicit programming, leveraging previous demonstrations of the desired behavior. Our approach relies on the VS paradigm and the DS-based IL rationale. In particular, we take advantage of the imitation strategy to learn the compensation term required to achieve the visual tracking experiment. Our approach permits us to realize the tracking without the specific implementation of an estimator or observer of the

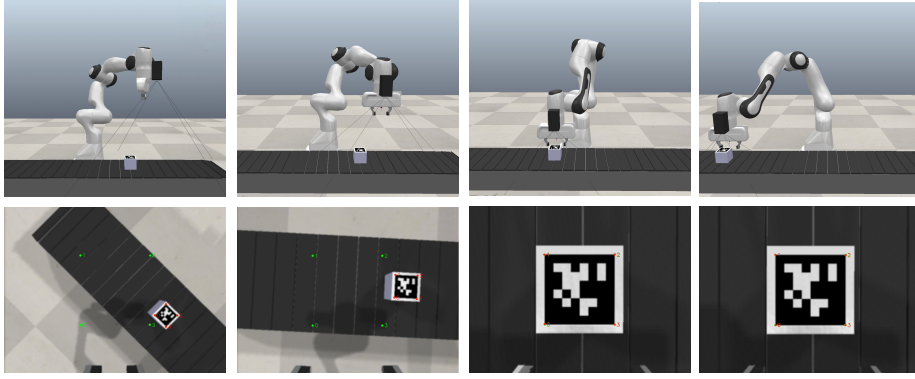


Fig. 9: Snapshots of the ILVS experiment with unseen initial position and orientation: robot’s external views (top) and camera images (bottom).

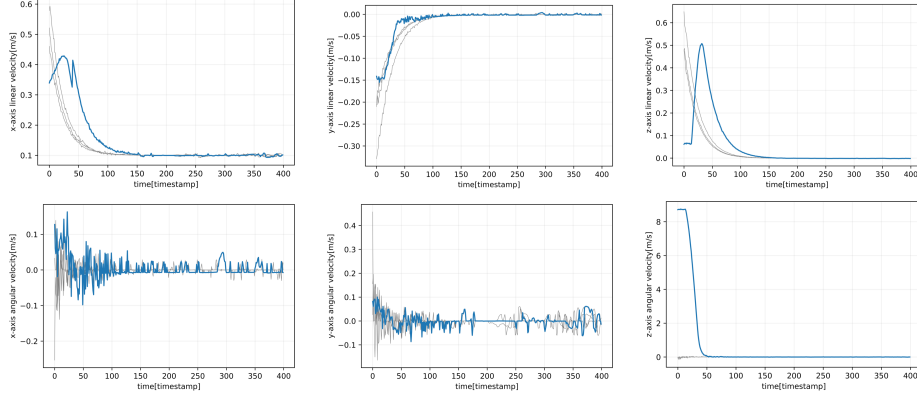


Fig. 10: Camera velocity during the ILVS experiment with unseen initial position and orientation: linear (top) and angular components (down).

compensation term. The framework has been evaluated with several simulations, which show the ability to handle unseen initial conditions.

As shown by the experiment in Fig. 6 and Fig. 7, the robot can converge to the visual target even starting relatively far from the initial value of the demonstrations. This out-of-domain generalization capability is a structural property of our approach that effectively combines a stable component (from standard VS) and a learned one in the closed-loop control law (5). Indeed, the standard VS component always drives the robot close to the target, i.e., in the training data domain, where the learning of the compensation term is put in an ideal condition to work. Stronger generalization capabilities (e.g., to handle the doubled velocity of the conveyor belt seen during the demonstrations) would require re-training our compensation term. The stability of the proposed controller has not

been formally investigated (for instance, using tools from the Lyapunov theory). However, in the conducted experiments, the robot was always able to reach the target with sub-millimeter precision. Moreover, we also tested the robustness to disturbances like changes in the object position on the conveyor belt. The fact that the controller behaved as expected in several practical cases suggests that it should have some (local) stability property. However, a formal stability proof is left as future work. Another interesting line for future development is the test of our framework with velocities of the object that are different from the one seen during the demonstrations. Indeed in our current study, the velocity of the object during the validation experiment is the same as the one used during the collection of the demonstrations. Finally, we plan to test our approach with real experiments; to this end, further development will be required to handle the noise in the input data (typical of real-life applications).

References

1. D. J. Agravante, G. Claudio, F. Spindler, and F. Chaumette. Visual servoing in an optimization framework for the whole-body control of humanoid robots. *IEEE Robotics and Automation Letters*, 2(2):608–615, 2017.
2. Guillaume Allibert, Estelle Courtial, and François Chaumette. Predictive control for constrained image-based visual servoing. *IEEE Trans. Robot.*, 26(5):933–939, 2010.
3. Brenna Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.
4. Aude G Billard, Sylvain Calinon, and Rüdiger Dillmann. Learning from humans. *Springer handbook of robotics*, pages 1995–2014, 2016.
5. R. Caccavale, M. Saveriano, G. A. Fontanelli, F. Ficuciello, D. Lee, and A. Finzi. Imitation learning and attentional supervision of dual-arm structured tasks. In *IEEE International Conference on Development and Learning and Epigenetic Robotics*, pages 66–71, 2017.
6. François Chaumette and A Santos. Tracking a moving object by visual servoing. *IFAC Proceedings Volumes*, 26(2):643–648, 1993.
7. François Chaumette and Seth Hutchinson. Visual servo control. I. Basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, 2006.
8. François Chaumette and Seth Hutchinson. Visual servo control. II. Advanced approaches. *IEEE Robotics & Automation Magazine*, 14(1):109–118, 2007.
9. Francois Chaumette, Patrick Rives, and Bernard Espiau. Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In *IEEE International Conference on Robotics and Automation*, pages 2248–2253, 1991.
10. G. Chesi and Y. S. Hung. Global path-planning for constrained and optimal visual servoing. *IEEE Trans. Robot.*, 23(5):1050–1060, 2007.
11. CoppeliaSim. Coppeliasim, coppelia robotics, 2013.
12. Peter I Corke and Malcolm C Good. Dynamic effects in visual closed-loop systems. *IEEE Transactions on Robotics and Automation*, 12(5):671–683, 1996.

13. Romuald Ginhoux, Jacques Gangloff, Michel de Mathelin, Luc Soler, Maria Mara Arenas Sanchez, and Jacques Marescaux. Active filtering of physiological motion in robotized surgery using predictive control. *IEEE Transactions on Robotics*, 21(1):67–79, 2005.
14. Sami Haddadin, Sven Parusel, Lars Johannsmeier, Saskia Golz, Simon Gabl, Florian Walch, Mohamadreza Sabaghian, Christoph Jaehne, Lukas Hausperger, and Simon Haddadin. The Franka Emika robot: A reference platform for robotics research and education. *IEEE Robotics & Automation Magazine*, 2022.
15. S Mohammad Khansari-Zadeh and Aude Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.
16. Éric Marchand and François Chaumette. Feature tracking for visual servoing purposes. *Robot. Auton. Syst.*, 52(1):53–70, 2005.
17. Y. Mezouar and F. Chaumette. Path planning for robust image-based control. *IEEE Trans. Robot.*, 18(4):534–549, 2002.
18. Enrico Mingo Hoffman and Antonio Paolillo. Exploiting visual servoing and centroidal momentum for whole-body motion control of humanoid robots in absence of contacts and gravity. In *IEEE International Conference on Robotics and Automation*, pages 2979–2985, 2021.
19. A. Paolillo, K. Chappellet, A. Bolotnikova, and A. Kheddar. Interlinked visual tracking and robotic manipulation of articulated objects. *IEEE Robotics and Automation Letters*, 3(4):2746–2753, 2018.
20. A. Paolillo, T. S. Lembono, and S. Calinon. A memory of motion for visual predictive control tasks. In *IEEE International Conference on Robotics and Automation*, pages 9014–9020, 2020.
21. Antonio Paolillo, Marco Forgione, Dario Piga, and Enrico Mingo Hoffman. Fast predictive visual servoing: A reference governor-based approach. *Control Engineering Practice*, 136:105521, 2023.
22. Antonio Paolillo, Mirko Nava, Dario Piga, and Alessandro Giusti. Visual servoing with geometrically interpretable neural perception. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5300–5306, 2022.
23. Antonio Paolillo, Paolo Robuffo Giordano, and Matteo Saveriano. Dynamical system-based imitation learning for visual servoing using the large projection formulation. In *IEEE International Conference on Robotics and Automation*, pages 755–761, 2023.
24. Antonio Paolillo and Matteo Saveriano. Learning stable dynamical systems for visual servoing. In *IEEE International Conference on Robotics and Automation*, pages 8636–8642, 2022.
25. ROS. Ros, robotic operating system, 2007.
26. M. Sauvee, P. Poignet, E. Dombre, and E. Courtial. Image based visual servoing through nonlinear model predictive control. In *IEEE Conference on Decision and Control*, pages 1776–1781, 2006.
27. Matteo Saveriano. An energy-based approach to ensure the stability of learned dynamical systems. In *IEEE International Conference on Robotics and Automation*, pages 4407–4413. IEEE, 2020.
28. Matteo Saveriano and Dongheui Lee. Incremental skill learning of stable dynamical systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6574–6581, 2018.
29. Stefan Schaal. Learning from demonstration. *Advances in Neural Information Processing Systems*, 9, 1996.

30. Eugene Valassakis, Georgios Papagiannis, Norman Di Palo, and Edward Johns. Demonstrate once, imitate immediately (DOME): Learning visual servoing for one-shot imitation learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 8614–8621, 2022.
31. Shelten G Yuen, Samuel B Kesner, Nikolay V Vasilyev, Pedro J Del Nido, and Robert D Howe. 3D ultrasound-guided motion compensation system for beating heart mitral valve repair. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 711–719. Springer, 2008.