Neural-Network for Position Estimation of a Cable-Suspended Payload Using Inertial Quadrotor Sensing

Julien Mellet[a], Jonathan Cacace[b], Fabio Ruggiero[c] and Vincenzo Lippiello[d]
PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, Via Claudio 21, 80125, Naples, Italy

Keywords: Neural Network, State Estimation, Quadrotor, Cable Suspended Payload.

Abstract: This paper considers a standard quadrotor drone with a cable-suspended payload and minimal sensor configuration. A neural network estimator is proposed to perform accurate real-time payload position estimation. A novel proprioceptive feedback measurement method is proposed, and a neural network has been trained with domain randomization. The network shows accurate zero-shot estimation, even with excitations never seen by the system before. This preliminary work has been tested in a simulated environment and aims to show that only onboard inertial sensing is enough to achieve the sought task. The presented work may open new applications for drone transportation in real environments subject to several perturbations.

1 INTRODUCTION

For aerial vehicles, there is a weight obsession such that, especially for transportation tasks, each saved gram on the platform is a gram for the payload (Anderson and Gaston, 2013). Thanks to its simplicity, a cable-suspended payload under an aerial vehicle remains an elegant way to move any object through the air (see Fig. 1). Compared to a mechatronic arm, this system is lightweight and easy to install, utilizing a single passive cable fixation point. (Suarez et al., 2020). Despite new aerial platforms like omnidirectional drones, quadrotors have demonstrated a decade-long reliability (Bodie et al., 2019). A quadrotor with suspended load is comparable to helicopters achieving object transport but with a difference in agility and sensory system (Wendel et al., 2006). Nevertheless, research tends to increase flight accuracy by embedding as many sensors as possible at the expense of lightness and agility (Lanegger et al., 2 06), (Panetsos et al., 2022).

This preliminary study presents an approach that simplifies suspended payload position estimation in a standard quadrotor, eliminating the need for exteroceptive sensor processing. Hence, only internal state configuration at a high temporal rate is employed instead of cameras or any position sensor. Such systems are robust to environmental variations. The disturbances induced by the payload on the system are used to let a specially trained network find a correlation with the payload position. To the best of the authors' knowledge, no prior work achieved such accuracy in estimating suspended payload position in minimal aerial transportation setups.

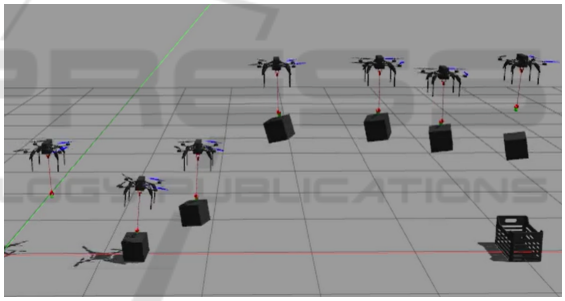

Figure 1: Pick and place tasks of the cable-suspended system. In the first frame, the drone reaches the package. The grasp of the payload is done in the second frame. From frame three to six, the load is transported. The last frame shows the packet's release, which is dropped into its target location.

1.1 Related Works

Conventional approaches to cable-suspended payload transportation use external sensors such as cam-

[a] https://orcid.org/0009-0007-8386-0012
[b] https://orcid.org/0000-0002-1639-5655
[c] https://orcid.org/0000-0001-7539-9157
[d] https://orcid.org/0000-0002-6089-2333

Mellet, J., Cacace, J., Ruggiero, F. and Lippiello, V.
Neural-Network for Position Estimation of a Cable-Suspended Payload Using Inertial Quadrotor Sensing.
DOI: 10.5220/0012204100003543
In *Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2023) - Volume 1*, pages 80-87
ISBN: 978-989-758-670-5; ISSN: 2184-2809

era (Tang et al., 2018), (Guo and Leang, 2020). Getting a direct measurement of the payload position is a reasonable option. Even with a relatively low measurement rate compared to the low-level controller, it allows making aggressive flights (Tang et al., 2018). However, cameras suffer from several problems like brightness (close to shadow), flickering (while flying over the sea), or fogging (in a humid area), making their use standalone unsafe for reliable industrial applications. The issue has also been revealed by (Lee and Kim, 2017), where the proposed solution consists of adding force sensors to the system. Equivalent sensing solutions have been adopted by (Lv et al., 2021), with the addition of a universal joint between the drone and the cable for the second one. Recently, (Panetsos et al., 2022) used four different sensors to get accurate cable state estimation, while (Outeiro et al., 2023) proposed an adaptive geometric control method with asymptotic tracking stability.

A similar approach to ours is described by (Kaufmann et al., 2020) and (Cioffi et al., 2022) to perform agile maneuvers. The use of internal sensing in those papers is called sensorimotor, but for the unification of the terms in robotics, we will call it proprioception, like it has been done in (Lee et al., 2020). Even if we do not have access to direct measurement of the motor speed, the PWM (Pulse-Width Modulation) command gives a fair proportional estimation. A neural network (NN) is implemented to control the drone. In particular, the work done by (Cioffi et al., 2022) learns inertial odometry and gets accurate position estimation without using any visual perception. However, the positioning tracking has been done with previously known trajectories. In contrast, for safety reasons, we implemented a neural network trained by supervised learning on the estimation stage of the tethered payload controller.

One of the most advanced research on payload state estimation using quadrotor proprioception is for parameter estimation (Prkačin et al., 2020). Recently, such a work has been improved by (Prkačin et al., 2021) with the implementation of an extended Kalman filter (EKF). However, estimation of load parameters remains challenging with the employed fast Fourier transform technique, getting only off-line results. Unfortunately, even if the research looks in an interesting direction with a minimal drone setup, real-time performance has not been reached. Real-world experiments showed poor results in tracking the angles of the payload. The system is non-linear, and the IMU is noisy while drones fly at high velocity. This makes classical EKF implementation unusable for real flights application. To tackle real-time, we use the concept of a neural observer (Chen et al., 2018) with a recurrent NN to access indirectly measured data (Habtom and Litz, 1997).

1.2 Contribution

In this paper, we prove the feasibility of real-time position estimation of a cable-suspended end-effector using only inertia sensors onboard a standard quadrotor. Contrary to the classic EKF approach, our method does not need to define any parameter a priori. We used a supervised learned network, making the position estimation of the suspended load for a standard quadrotor attitude controller. The neural estimator is trained in a simulated environment, with domain randomization, and runs the software controller in the loop. We reached zero-shot generalization of the network for load position estimation. After being trained on our dataset, the network has the capability to predict previously unseen perturbations, in particular, direct injection of energy into the end-effector. This shows the consistency of the estimator implementation, as well as its generalization capabilities.

2 MATERIALS AND METHODS

Making payload pose estimation is not a trivial task, and the classical filter approach showed accuracy limitations (Prkačin et al., 2021). In this section, we first define our model, which is needed for the simulation, and implement an attitude controller. Then, we focus on data processing to measure our features to train the network. Finally, to smooth the training, we gradually increase the measurement domain exploration with progressively more sophisticated trajectories.

2.1 Model

For the quadrotor, we consider the dynamic model with the Euler angles (Ollero and Siciliano, 2019). Here we use FLU (Front-Left-Up) convention to define the body frame B with axes $\{x,y,z\}$, and the world frame W with axes $\{x_w,y_w,z_w\}$. We define the drone position and its attitude as $p_b = [x_b,y_b,z_b]^\top \in \mathbb{R}^3$ and $\eta_b = [\phi_b,\theta_b,\psi_b]^\top \in \mathbb{R}^3$, respectively, with their time derivatives $\dot{p}_b$ and $\dot{\eta}_b$. The attitude can also be defined with $R_b \in SO(3)$, the rotation matrix from B to W, the special orthogonal group of dimension three, from which the roll-pitch-yaw angles ϕ_b, θ_b, and ψ_b, respectively, can be extracted. These give the linear and angular accelerations of the base as,

$$\ddot{p}_b = g\,e_3 + \frac{1}{m}\,u_T\,R_b\,e_3, \tag{1}$$

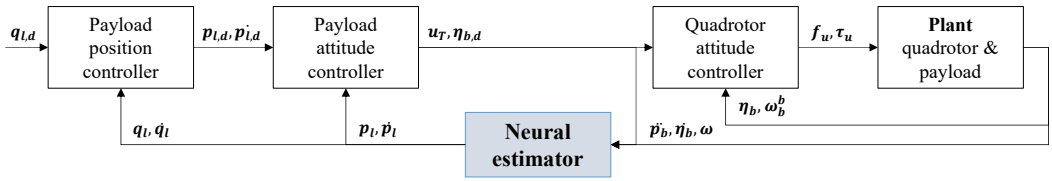

Figure 2: Control layout architecture with neural network state estimation.

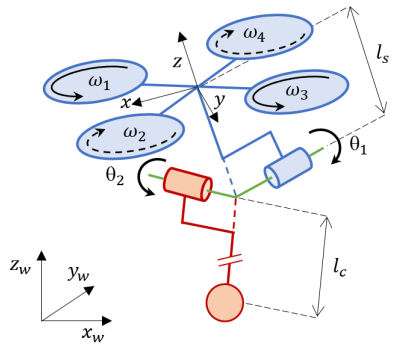

Figure 3: Schematic representation of the quadrotor (in blue) with the cable-suspended payload (in red). The terms $l_s, l_c \in \mathbb{R}$, with $l_c >> l_s$ are the support length and the cable length, respectively. Besides, $\omega_i \in \mathbb{R}, i = 1,\dots,4$ are the rotation speed of the motors, while $\theta_1, \theta_2 \in \mathbb{R}$ are the angles of the cable with respect to the drone.

$$\ddot{\eta}_b = M(\eta_b)^{-1}(-C(\eta_b,\dot{\eta}_b) + Q^\top(\eta_b)\tau_b), \quad (2)$$

where $g \in \mathbb{R}$ is the gravity, $m \in \mathbb{R}$ is the drone mass, and $e_3 = [0,0,1]^\top$. Moreover, $M(\eta_b) = Q(\eta_b)^\top I_b Q(\eta_b) \in \mathbb{R}^{3\times3}$ is the symmetric and positive definite (provided that $\theta \neq \pm\frac{\pi}{2}$) mass matrix, $I_b \in \mathbb{R}^{3\times3}$ is the drone inertia matrix, and $Q(\eta_b) \in \mathbb{R}^{3\times3}$ is the transformation matrix such that $\omega_b^b = Q(\eta_b)\dot{\eta}_b$, where $\omega_b^b \in \mathbb{R}^3$ is the angular velocity of the B with respect to the W expressed in B. Finally, $C(\eta_b,\dot{\eta}_b) = Q^\top(\eta_b)S(Q(\eta_b)\dot{\eta}_b)I_bQ(\eta_b) + Q^\top(\eta_b)^\top I_b\dot{Q}(\eta_b) \in \mathbb{R}^3$ is the Coriolis matrix, with $S(\cdot) \in \mathbb{R}^{3\times3}$ the skew-symmetric operator, $\tau_b = [\tau_x,\tau_y,\tau_z]^\top \in \mathbb{R}^3$ is the torque control vector, and $u_T \in \mathbb{R}^+$ is the total thrust. The physical control inputs to the system, that is the propeller velocities $\omega_i \in \mathbb{R}$, with $i = 1,\dots,4$ (see Fig. 3), can be retrieved from the torques and the total thrust through the allocation matrix as follows,

$$\begin{bmatrix} u_T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & lc_T & 0 & -lc_T \\ -lc_T & 0 & lc_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}, \quad (3)$$

where $c_T, c_Q \in \mathbb{R}^+$ are the thrust constant and drag factor, respectively.

For the suspended-cable, several models were proposed in the literature, such as finite element approximation (Goodarzi et al., 2014), elastic rope (Kotaru et al., 2017), or a rigid bar (Tang et al., 2018). The last model has been chosen for ease of simulation. It represents the most interesting case for transportation. Details of the model are presented by (Sreenath et al., 2013) and explained in Fig. 3. Recall that the quadrotor is a differentially flat system (Rathinam et al., 1995), while the quadrotor with a cable-suspended payload is a differentially flat hybrid system (Tang et al., 2018; Sreenath et al., 2013). Hence, all state and input variables are defined through nonlinear equations involving flat variables and their respective derivatives.

The load is considered as a point mass, the mass of the cable is neglected, and the cable is considered rigid. The variables are defined in Fig. 3. The position of the quadrotor in W can be retrieved from the load position as

$$p_b = q_l - l_c p_l, \quad (4)$$

where $q_l \in \mathbb{R}^3$ is the load position in W and $p_l \in \mathbb{S}^2$, with $\mathbb{S}^2$ the manifold of unit vectors in $\mathbb{R}^3$, is the unit vector pointing the load from the quadrotor's center of mass.

2.2 Control

The controller presented by (Sreenath et al., 2013) and applied by (Tang et al., 2018) is here briefly reported. The state and input are defined as, $x = [p_l^\top, \dot{p}_l^\top, p_b^\top, \dot{p}_b^\top, \dot{\eta}_b^\top, \omega_b^{b\top}]^\top$ and $u = [f_u^\top, \tau_u^\top]^\top$, respectively.

First, define the low level quadrotor attitude controller as,

$$\begin{aligned} \tau_b = &-K_R e_R - K_\omega e_\omega + S(\omega_b^b)I_b\omega_b^b \\ &-I_b(S(\omega_b^b)R_b^\top R_{b,d}\omega_{b,d}^{b,d} - R_b^\top R_{b,d}\dot{\omega}_{b,d}^{b,d}), \end{aligned} \quad (5)$$

where K_R, $K_\omega \in \mathbb{R}^{3\times3}$ are diagonal gain matrices, $e_R = \frac{1}{2}(R_{b,d}^\top R_b - R_b^\top R_{b,d})^\vee \in \mathbb{R}^3$ is the quadrotor attitude error, $e_\omega = \omega_b^b - R_b^\top R_{b,d}\omega_{b,d}^{b,d} \in \mathbb{R}^3$ is the angular velocity error, $(\cdot)^\vee$ is the *vee* operator mapping a skew-symmetric matrix into the vector that generated it, $R_{b,d} \in SO(3)$ is the desired quadrotor rotation matrix, $\omega_{b,d}^{b,d} \in \mathbb{R}^3$ is the desired angular velocity expressed in the desired body frame.

The desired $R_{b,d}$ and $\omega_{b,d}^{b,d}$ are retrieved from the position controller as explained by (Tang et al., 2018).

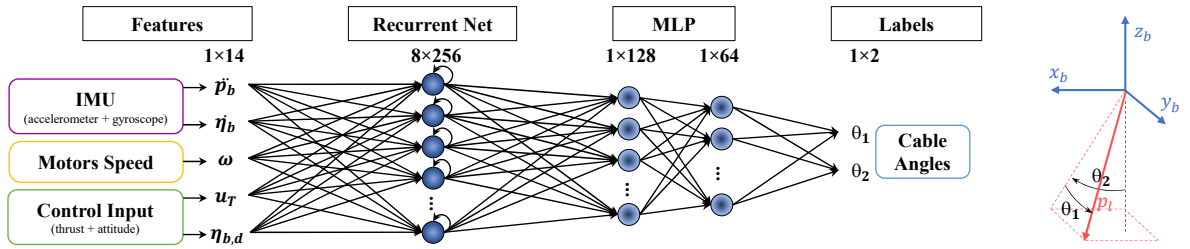

Figure 4: Neural network architecture. The NN keeps the history of the past data at the same rate. All features are processed through a recurrent neural network layer. Then data are processed with a fully connected multi-layer perception before getting the two angles of the cable.

The thrust input results from,

$$f_u = u_T R_b e_3, \tag{6}$$

where,

$$u_T = K_p e_p + K_{\dot{p}} e_{\dot{p}} + K_i \int e_p dt, \tag{7}$$

with $K_p, K_{\dot{p}}, K_i \in \mathbb{R}^3$ some diagonal and positive definite gain matrices, $e_p = S(p_l)^2 p_{l,d}$ the error function with $p_{l,d} \in \mathbb{R}^3$ the desired load position in W, and $e_{\dot{p}} = \dot{p}_l - S(S(p_{l,d})\dot{p}_{l,d})p_l$ the time derivative. Then, the desired quadrotor attitude is,

$$R_{b,d} = \left[S\left(\frac{S(b)c}{\|S(b)c\|}\right) b, \frac{S(b)c}{\|S(b)c\|}, c \right], \tag{8}$$

with $\|\cdot\|$ the Euclidean norm, $b = \dfrac{u_T}{\|u_T\|} \in \mathbb{R}$, and $c = [\cos(\psi_b), \sin(\psi_b), 0]^\top \in \mathbb{R}^3$.

Finally the payload position controller is defined as,

$$q_{l,d} = K_q e_q + K_{\dot{q}} e_{\dot{q}} + K_{q,i} \int e_q dt, \tag{9}$$

with $K_q, K_{\dot{q}}, K_{q,i} \in \mathbb{R}^3$ some diagonal and positive definite gain matrices, $e_q = q_l - q_{l,d} \in \mathbb{R}^3$ the error function, and $e_{\dot{q}} = \dot{q}_l - \dot{q}_{l,d} \in \mathbb{R}^3$ its time derivative.

The controller implementation is presented in Fig. 2 and the NN estimator is explained in the next section.

2.3 Neural Estimation

Through all the data provided by the IMU, the magnetometer has been found to add too much noise and randomness to the result. Then, only the accelerometer and the gyroscope are used.

Compared to (Kaufmann et al., 2020), we found that sampling measurements with synchronization of all the features were getting accurate results. Normalization is done with min-max normalization. Assuming $X \in \mathbb{R}$ a scalar, we use the unit-based normalization,

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}. \tag{10}$$

To avoid any overshoot on measurements $X_{meas} \in \mathbb{R}$, boundaries limits are fixed with saturation values such that,

$$X_{meas} \in [X_{min}, X_{max}]. \tag{11}$$

The main network architecture Fig. 4 remains condensed, with relatively few hidden layers. Keeping the network as small as possible is to implement it on a small computational unit and reduce the processing latency. It has three incoming features branches with the IMU $(\ddot{p}_b, \dot{\eta}_b) \in \mathbb{R}^6$ the motor speeds $\omega \in \mathbb{R}^4$, and the control inputs $(u_T, \eta_{b,d}) \in \mathbb{R}^4$. There is a total of fourteen scalar input features $[\ddot{p}_b, \dot{\eta}_b, \omega, u_T, \eta_{b,d}]$ preprocessed as mentioned above.

For a later comparison study, we want to compare different promising NNs, namely, a time convolutional network (TCN) and two recurrent neural networks (RNN). For both cases, networks are trained on a history of N time past steps to the current measurement that are defined later. Even if TCN proved its efficiency in neural estimation for drones (Kaufmann et al., 2020; Cioffi et al., 2022), RNN looks to be a more appropriate option. These layers take advantage of processing sequential data to make the prediction. In this study, we used Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) (LSTM) hidden neurons that have also been used by (Peringal et al., 2022; Jung et al., 2022). We can notice that simple RNN and gated recurrent unit (GRU) have also been tested during this research, but the results were similar to LSTM with poorer prediction accuracy. For this reason, LSTM is the chosen recurrent net architecture.

The recurrent network trained with $N = 15$ timestamps of the past proprioceptive measurements follows. Even if the eight past measures were used by (Kaufmann et al., 2020), we got better performance with fifteen. Then, two fully connected layers of MLPs give the cable angles (θ_1, θ_2). We use differentiation to obtain the angular velocity $\dot{p}_l$ needed for the controller in Fig. 2.

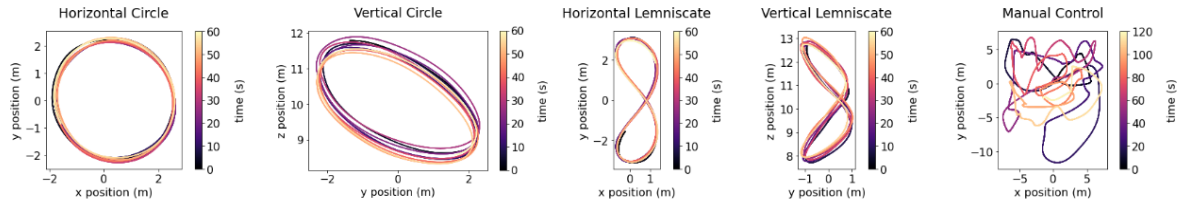

Figure 5: Sample trajectories used to train networks. From left to right, with trajectories getting more complex through the training: 1 min horizontal circle trajectory; 1 min vertical circle trajectory; 1 min horizontal lemniscate trajectory; 1 min vertical lemniscate trajectory; 2 min of manual control.

2.4 Training

Compared to (Lee et al., 2020), (Faust et al., 2017) that substantially used reinforcement learning to train their networks, we used supervised learning. However, we kept the principle of accelerating the training with a given policy. Thus, we proposed to begin the training with simple trajectories, letting the loss converge smoothly. Then, we complicated the trajectories to finish with erratic movements. Here, only state estimation is performed by the network. The advantage of virtual training, is to have an infinite flight time with variable parameters. The payload position controller is fed with circular, lemniscate, and random manual trajectories (see Fig. 5).

We trained the NN estimator through ten datasets with the five different trajectories, depicted in Fig. 5, and two different payloads of 200 g and 800 g. The sum of the datasets gives a total of 140.10^3 timestamped measurements for 50 min flight time. We split each trajectory dataset in two, with 20% for validation and the remaining 80% for training.

To predict the load position, p_l, a time window of size 16 has been used. Thanks to the 15 past measured features, the 16^{th} label is predicted. Because the regressive network has to keep consistency through time, no data shuffle was made. Nevertheless, each data set was split up into batches of size 256. Trying different batch sizes, it turned out this last was a good trade-off. For the simulation, we used Gazebo, a simulator with a physic engine (Smith, 2008). The propellers' drag, lift coefficients and IMU characteristics are simulated through the internal plugin. We simulate our quadcopter with a stick with a two-degree-of-freedom (DoF) universal joint, presented in Fig. 3. Compared to real flights, the simulator does not provide any aerodynamic effects. As a result, the payload is static under the drone while hovering. We then propose to make domain randomization through the excitation of the end-effector. It consists in applying a relatively small bounded random force on the end-effector. In our case, we selected a maximum force of 0.2 N. It is now about comparing the network's performances.

Table 1: Architectures of the three networks that are compared for the study. LSTM2 with two recurrent layers and an MLP before the output. TCN with a convolutional layer followed by an MLP. LSTM embedding a unique recurrent layer and two MLPs.

	LSTM2		**TCN**		**LSTM**	
Hidden layer 1	LSTM	256	Conv		LSTM	256
Hidden layer 2	LSTM	128	MLP	128	MLP	128
Hidden layer 3	MLP	64	X		MLP	64

2.5 Networks Comparisons

The first set of experiments compares the accuracy in the prediction of three promising neural architectures. Table 1 compares the architectures of the three networks used for the research. with two recurrent layers and an MLP before the output. TCN with a convolutional layer followed by an MLP. LSTM embeds a unique recurrent layer and two MLPs. Then, the training through the different trajectories is presented in Fig. 6.

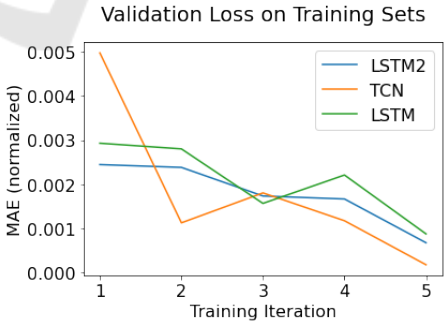

Figure 6: Validation loss after training on each trajectory dataset. From left to right, 1 corresponds to the horizontal circle, 2 corresponds to the vertical circle, 3 corresponds to the horizontal lemniscate, 4 corresponds to the vertical lemniscate, and 5 corresponds to manual control.

One can notice that the validation loss decreases after each trajectory train. This points out the improvement of the estimation prediction with more and

more sophisticated trajectories. Nevertheless, even if TCN showed a better accuracy on the validation set, RNN performs better with the test set (see Fig. 7).

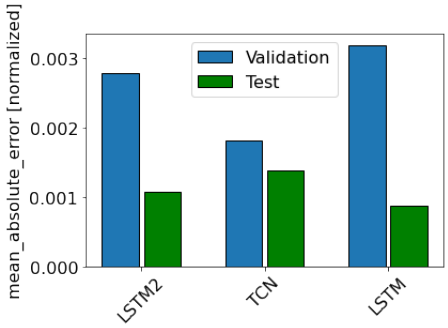

Figure 7: Mean absolute error of LSTM2, TCN, and LSTM after being trained on the different datasets.

The better performance of TCN in Fig. 7 is noticeable on the validation set, while LSTM is more accurate on the test set. LSTM2 and even more LSTM has thus a generalization capability. Both temporal convolution and recurrent networks result in the same order of magnitude, thanks to temporal measurements.

The following section uses only LSTM to understand what implies the change in payload mass on the NN estimator.

3 APPLICATIONS

NN estimation takes a certain time to process. We made an inference model of our network to accelerate pose estimation during the experiments. We improved the pose estimation processing from a mean response time of 4.10^{-2} s to an instant estimation of 10^{-4} s without losing accuracy.

3.1 Response to External Perturbations

Because the weight of the payload directly impacts drone perturbation, and by implication the inertial sensor measurements, we want to understand how the weight impacts pose estimation of the end-effector. To measure the accuracy of prediction, we have perturbed the load on the single x-axis with 40 N for the 200 g load (see Fig. 8) and with 100 N for the 800 g load (see Fig. 9).

For both plots, with light-weight (see Fig. 8) and heavy payload (Fig. 9), we notice that the estimation follows the ground truth through time with a drop in accuracy at the perturbation. Moreover, no perturbation is measured on θ_1, the non-perturbed axis, showing the independence of each angle estimation.

Another critical point is the noise reduction in payload state estimation with a heavier load. This phenomenon should be explained for two reasons: firstly, because the heavier load, with greater inertia, have slower dynamics, which is easier to predict; secondly, because the perturbation of the largest mass has more effect on the system and disturbance can be measured with higher intensity by the inertial unit.

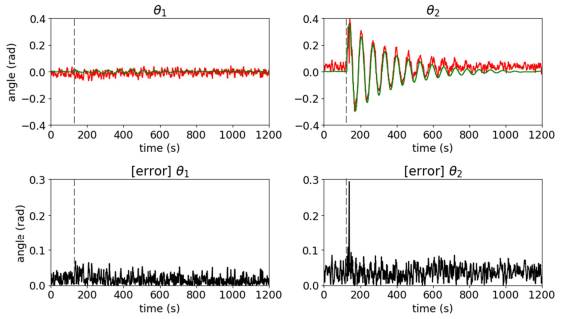

Figure 8: Cable orientation estimation for a 200 g load and 40 N force excitation. The mean absolute error (MAE) between ground-truth in green and estimated angle in red, has been measured at 0.05326 rad. Perturbation is done at $t = 126$ s.

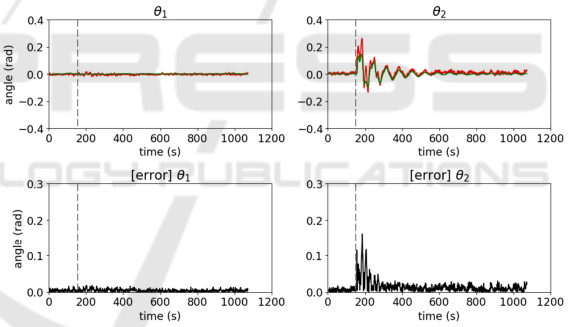

Figure 9: Cable orientation estimation for a 800 g load and 100 N force excitation. The mean absolute error (MAE) between ground-truth in green and estimated angle in red, has been measured at 0.01772 rad. Perturbation is done at $t = 155$ s.

3.2 Use Case Example

The proposed estimator was then applied in a pick-and-place scenario. The idea is to understand the estimation performance without training the network for this specific application. The drone picks up a 600 g package and transports it to a basket at the target location. Figure 1 presents the environment in which the system is used. The drone flies at a relatively slow speed, keeping the near-hovering assumption.

Compared to the system subjected to simple disturbances (see Figs. 8-9), we can notice a more significant mean absolute error (MAE) during the entire task in Fig. 10. The same comments as previously can

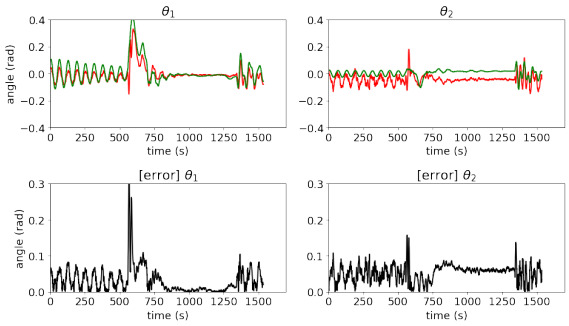

Figure 10: Cable orientation estimation during a pick and place task. The payload is grasped at $t = 580$ s, and released at $t = 1310$ s. The mean absolute error (MAE) between ground-truth in green and estimated angle in red, has been measured at 0.08365 rad.

be made, on the influence of weight on estimation. For θ_1, MAE is more significant with the unloaded end-effector, while precision increases with the payload transported. An estimation bias has to be noticed for θ_2 during the carriage. Likewise, the grasping action being similar to a prompt perturbation, the same result is observed as above, with a drop in accuracy. In contrast to the grab, the release is smoother, and the accuracy of the estimate changes from a lower to a higher MAE.

4 CONCLUSION AND FUTURE WORK

This preliminary study demonstrates that inertial measurements are sufficient for estimating payload position in cable-suspended drone systems, enabling their integration into the controller. Moreover, indirect neural pose estimation has been proven accurate in performing stable transportation tasks near hovering. The hardware complexity has thus decreased. This estimator suits cable-suspended underwater transport, free from sun reflection flickering on water waves.

Future work will focus on transferring the NN from simulation to reality thanks to domain randomization presented in this paper. The core idea is to test the neural estimator in challenging-to-simulate environments. On the other hand, we would like to take advantage of this estimator to perform aggressive maneuvers implementing model predictive control.

ACKNOWLEDGEMENTS

The research leading to these results has been supported by the AERIAL-CORE project (Horizon 2020 Grant Agreement No. 871479) and by the AERO-TRAIN Project, European Union's Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie Grant Agreement 953454. The authors are solely responsible for its content.

REFERENCES

Anderson, K. and Gaston, K. J. (2013). Lightweight unmanned aerial vehicles will revolutionize spatial ecology. *Frontiers in Ecology and the Environment*, 11(3):138–146.

Bodie, K., Brunner, M., Pantic, M., Walser, S., Pfndler, P., Angst, U., Siegwart, R., and Nieto, J. (2019). An Omnidirectional Aerial Manipulation Platform for Contact-Based Inspection. In *Robotics: Science and Systems XV*. Robotics: Science and Systems Foundation.

Chen, B., Zhang, H., Liu, X., and Lin, C. (2018). Neural Observer and Adaptive Neural Control Design for a Class of Nonlinear Systems. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):4261–4271.

Cioffi, G., Bauersfeld, L., Kaufmann, E., and Scaramuzza, D. (2022). Learned Inertial Odometry for Autonomous Drone Racing.

Faust, A., Palunko, I., Cruz, P., Fierro, R., and Tapia, L. (2017). Automated aerial suspended cargo delivery through reinforcement learning. *Artificial Intelligence*, 247:381–398. Special Issue on AI and Robotics.

Goodarzi, F. A., Lee, D., and Lee, T. (2014). Geometric stabilization of a quadrotor UAV with a payload connected by flexible cable. In *2014 American Control Conference*, pages 4925–4930.

Guo, D. and Leang, K. K. (2020). Image-Based Estimation, Planning, and Control of a Cable-Suspended Payload for Package Delivery. *IEEE Robotics and Automation Letters*, 5(2):2698–2705.

Habtom, R. and Litz, L. (1997). Estimation of unmeasured inputs using recurrent neural networks and the extended Kalman filter. In *Proceedings of International Conference on Neural Networks (ICNN'97)*, volume 4, pages 2067–2071 vol.4.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780.

Jung, J., You, S., Kim, D., and Park, J. (2022). Variable Stiffness Control via External Torque Estimation Using LSTM. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8325–8330.

Kaufmann, E., Loquercio, A., Ranftl, R., Müller, M., Koltun, V., and Scaramuzza, D. (2020). Deep Drone Acrobatics.

Kotaru, P., Wu, G., and Sreenath, K. (2017). Dynamics and control of a quadrotor with a payload suspended through an elastic cable. In *2017 American Control Conference (ACC)*, pages 3906–3913.

Lanegger, C., Ruggia, M., Tognon, M., Ott, L., and Siegwart, R. (2022-06). Aerial layouting: Design and control of a compliant and actuated end-effector for precise in-flight marking on ceilings. In *Proceedings of Robotics: Science and System XVIII*, page p073, s.l. Robotics Science & Systems Foundation.

Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., and Hutter, M. (2020). Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47).

Lee, S. J. and Kim, H. J. (2017). Autonomous swing-angle estimation for stable slung-load flight of multi-rotor UAVs. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4576–4581.

Lv, Z.-Y., Li, S., Wu, Y., and Wang, Q.-G. (2021). Adaptive Control for a Quadrotor Transporting a Cable-Suspended Payload With Unknown Mass in the Presence of Rotor Downwash. *IEEE Transactions on Vehicular Technology*, 70(9):8505–8518.

Ollero, A. and Siciliano, B. (2019). *Aerial Robotic Manipulation*. Springer.

Outeiro, P., Cardeira, C., and Oliveira, P. (2023). Control architecture for a quadrotor transporting a cable-suspended load of uncertain mass. *Drones*, 7(3).

Panetsos, F., Karras, G. C., Aspragkathos, S. N., and Kyriakopoulos, K. J. (2022). Precise Position Control of a Multi-rotor UAV with a Cable-suspended Mechanism During Water Sampling. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1780–1786.

Peringal, A., Chehadeh, M., Azzam, R., Hamandi, M., Boiko, I., and Zweiri, Y. (2022). Design of Dynamics Invariant LSTM for Touch Based Human–UAV Interaction Detection. *IEEE Access*, 10:116045–116058.

Prkačin, V., Palunko, I., and Petrović, I. (2021). Extended Kalman filter for payload state estimation utilizing aircraft inertial sensing. In *2021 Aerial Robotic Systems Physically Interacting with the Environment (AIRPHARO)*, pages 1–6.

Prkačin, V., Palunko, I., and Petrović, I. (2020). State and parameter estimation of suspended load using quadrotor onboard sensors. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 958–967.

Rathinam, M., Murray, R. M., and Sluis, W. M. (1995). Differential Flatness of Mechanical Control Systems: A Catalog of Prototype Systems.

Smith, R. (2008). Open Dynamics Engine. http://www.ode.org/.

Sreenath, K., Lee, T., and Kumar, V. (2013). Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load. In *52nd IEEE Conference on Decision and Control*, pages 2269–2274.

Suarez, A., Vega, V. M., Fernandez, M., Heredia, G., and Ollero, A. (2020). Benchmarks for Aerial Manipulation. *IEEE Robotics and Automation Letters*, 5(2):2650–2657.

Tang, S., Wüest, V., and Kumar, V. (2018). Aggressive Flight With Suspended Payloads Using Vision-Based Control. *IEEE Robotics and Automation Letters*, 3(2):1152–1159.

Wendel, J., Meister, O., Schlaile, C., and Trommer, G. F. (2006). An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter. *Aerospace Science and Technology*, 10(6):527–533.