An LSTM-based approach to precise landing of a UAV on a moving platform

Wei Luo | Henrik Ebel | Peter Eberhard

Institute of Engineering and Computational Mechanics, University of Stuttgart, Stuttgart, Germany

Correspondence
Prof. Peter Eberhard, Institute of Engineering and Computational Mechanics, University of Stuttgart, Stuttgart, Germany.
Email: peter.eberhard@itm.uni-stuttgart.de

Funding information
Deutsche Forschungsgemeinschaft, Grant/Award Numbers: 433183605, EXC 2075-390740016; China Scholarship Council, Grant/Award Number: 201808080061

Abstract
A machine learning-based method for the precise landing of an unmanned aerial vehicle on a moving mobile platform is proposed. The proposed approach attempts to predict the mobile platform's future trajectory based on the past states of the mobile platform. To that end, it combines a long short-term memory-based neural network with a Kalman filter. Hence, it aims at combining the advantages of a machine learning method with those of a state estimation method from established control theory. Based on the predicted trajectory, the unmanned aerial vehicle attempts to land precisely on the moving mobile platform. The experiment is conducted in the Gazebo simulation platform with a quadrotor and an omnidirectional mobile robot, and the proposed method is compared with the single-method approaches of using only either the Kalman filter or the machine learning method alone.

KEYWORDS
Kalman filter, long short-term memory (LSTM), trajectory prediction, unmanned aerial vehicle (UAV)

1 | INTRODUCTION

Nowadays, unmanned aerial vehicles (UAVs) are utilized in various applications, including aerial surveillance, object grasping, and object transportation. Due to their limited payload capacity and current battery density, UAVs might benefit from active collaboration with unmanned ground vehicles (UGVs), forming a heterogeneous robotic system capable of performing some complicated tasks. During their collaboration, the UAV may seek assistance from the UGV, for example, to deliver the carried object or to replace or to recharge its onboard battery. As a result, landing precisely, smoothly, and safely on a moving platform is an exciting and challenging task for UAVs. Solving this problem will significantly increase the capabilities of mobile robot-based transportation systems that, as of now, often operate without aerial support.

Compared to a fixed point landing, landing on a moving platform requires a dynamic interaction between the UAV and the mobile landing platform. However, direct control of the landing platform by the UAV is impractical or even impossible in many application scenarios. Thus, here, we assume that the UAV is unaware of the future motion of the moving platform. In some previous work, a polynomial-based trajectory is used to guide the UAV to approach the ground vehicle. However, this trajectory provided only a connection between the UAV's current position and its landing platform, and the UAV can only follow the mobile robot's trace. To further improve the performance of this kind of landing task, it is critical to make a reasonable prediction of the trajectory of the landing platform. Without sufficiently accurate prediction, the UAV may make significant landing errors or even collide with the landing platform. Thus, past states of the landing
platform must be gathered and time-series prediction techniques must be utilized. This study assumes that the mobile landing platform follows a certain intention (which is, however, completely unknown to the UAV). Thus, the platform is following a reasonable, although maybe complicated, path rather than moving aimlessly or randomly. This is meaningful for real-world applications.

Numerous strategies are investigated for forecasting the future states of an observed target, such as ensemble methods,\textsuperscript{9} Gaussian processes,\textsuperscript{10} and restricted Boltzmann machine-based neural networks.\textsuperscript{11} Furthermore, according to Ref. 12, even a constant velocity model can quite accurately predict pedestrian movements compared to state-of-the-art approaches. Apart from these approaches, deep neural networks have demonstrated remarkable achievements in these types of applications. Recurrent neural networks (RNNs) show the ability to handle time-series problems, for example, for language translation and speech recognition.\textsuperscript{13} However, when their structure is vast and complicated, RNNs suffer from vanishing gradients in back-propagation. To address this issue, the long short-term memory (LSTM)\textsuperscript{14} and the gated recurrent unit\textsuperscript{15} methods have been proposed, both of which have an information control mechanism to manage the flow of information from the past inputs. Even these deep neural network-based approaches have their advantage, in that they can learn the observed pattern efficiently based on collected data, but the results predicted directly by these approaches lack information about kinematics, which, however, are essential in real-world robotic applications.

The purpose of this study is to develop a method for guiding a UAV onto a moving platform. Because localizing of the landing platform can be impeded by obstacles in the environment or by the effective sensor range, the robust prediction of the mobile landing platform’s trajectory is critical to ensure a safe and precise landing. To make a more reasonable guess of the future trajectory of the moving platform, rather than solely training a neural network to achieve a precise landing maneuver, the proposed method combines an LSTM network and a Kalman filter (KF) together, where the KF uses the LSTM network’s prediction results as the future observation of the moving platform to make a more acceptable prediction. Furthermore, the performance of the proposed method is compared to that of two other approaches in three different scenarios. To the best of our knowledge, this is the first application of this method for guiding a UAV landing on a dynamic landing platform. Additionally, in contrast to our previous work,\textsuperscript{16} which predicts a long-term trajectory for a platform based on its past route history and the guessed intended destination, this study focuses on the observed platform’s motion pattern to forecast a short-term trajectory and achieve a precise landing maneuver.

The paper is organized as follows: Section 1 describes the methods utilized for the trajectory prediction. In Section 3, the experimental environment and the control scheme are introduced. Subsequently, in Section 4, several scenarios and the experimental results are presented. Finally, conclusions are presented in Section 5.

2 | METHODS

2.1 Target modeling and the KF

In this study, an omnidirectional mobile ground robot is utilized as the moving landing platform. Compared with differential-drive mobile robots, omnidirectional robots have more moving flexibility, and are widely used in industry and research.\textsuperscript{17,18} Due to the fact that an omnidirectional mobile robot can move in any direction without turning, in the following, it is assumed that the robot maintains its orientation constant. Then, its orientation is irrelevant during landing, and its motion can be described using the state vector $x_k = [x, y, \dot{x}, \dot{y}]^T$ without rotation, where $x$ and $y$ denote the positions of the robot in the $xy$-plane. The dynamical model of the landing platform can be written as

$$x_{k+1} = Ax_k + Bu_k + w_k,$$

where the time-invariant system and input matrices are defined by $A$ and $B$, and the control input for the landing platform and Gaussian process noise are denoted as $u_{k-1}$ and $w_{k-1}$, respectively. The system matrix is defined as

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where the time difference between two consecutive states is denoted as $\Delta t$. Besides, in this study, since the quadrotor cannot directly affect or control the mobile landing platform, the input matrix $B$ is set to be a zero matrix and the input of the landing platform $u$ is neglected. The relationship between the state vector and the measurements from sensors can be described as

$$y_{k-1} = Cx_{k-1} + v_{k-1},$$

where the measurement at time step $k - 1$ is denoted as $y_{k-1}$ with corresponding Gaussian measurement noise $v_{k-1}$, and the output matrix $C$ maps the state of the mobile landing platform to the observed output accordingly.

Because the landing platform is not influenced by the UAV, the landing platform’s control input cannot be known in advance, and both process and measurement noise are also unknown to the UAV. As a result, Equations (1) and (3) cannot be used by the UAV for simulating the behavior of the moving platform. It must be understood that the moving platform is moving due to its own intentions and control, whereas these quantities are not available to the UAV. Thus, the UAV can only estimate the future motion of the moving platform.

A KF is utilized in this study to estimate the state of the landing platform.\textsuperscript{19} In the prediction phase, the predicted state $\hat{x}_{k|k-1}$, the predicted measurement $\hat{y}_{k|k-1}$, and the a priori estimate covariance $P_{k|k-1}$ are calculated by
\[ \begin{align*}
\dot{x}_{r,k|k-1} &= Ax_{r,k-1|k-1} \\
\dot{y}_{r,k|k-1} &= Cx_{r,k-1|k-1} \\
P_{r,k|k-1} &= AP_{r,k-1|k-1}A^T + Q,
\end{align*} \]

(4)

where the process noise covariance is denoted by Q. The estimated state \( \hat{x}_{r,k|k} \) and the a posteriori estimate covariance \( P_{r,k|k} \) at time step \( k \) can be obtained by

\[ \begin{align*}
L_k &= P_{r,k|k-1}C^T(R + CP_{r,k|k-1}C^T)^{-1} \\
\hat{x}_{r,k|k} &= \hat{x}_{r,k|k-1} + L_k(y_k - \hat{y}_{r,k|k-1}) \\
P_{r,k|k} &= (I - L_k C)P_{r,k|k-1},
\end{align*} \]

(5)

where the measurement from sensors at time step \( k \) is denoted by \( y_k \), and the matrices \( L_k \) and \( R \) are the Kalman gain and the covariance matrix of the measurement noise, respectively. Finally, the estimated state \( \hat{x}_{r,k|k} \) is utilized as the state of the mobile landing platform at time step \( k \) for further computations.

### 2.2 Prediction based on LSTM

To land precisely on the moving platform, one must be able to predict its future trajectory, allowing the UAV’s approach trajectory to be optimized in some sense and the landing planner to adjust the approaching position and speed in advance. However, a KF can only utilize past measurements to estimate the current state of the platform, as no future information is available. For the time horizons \( [k+1, k+2, \ldots] \), the states can be integrated only through the model as shown in Equation (1), but without regard for influences of the unknown future control input.

A method for predicting the trajectory of the observed platform can be based on an RNN, in which both the input data and the estimated results are organized in a time sequence. Thus, the temporal relationship between each state in the trajectory can be investigated through the neural network. In this study, a variation of an RNN, that is, LSTM, is utilized to forecast the moving platform’s trajectory over a constant time horizon into the future, as it avoids the gradient vanishing problem that occurs with standard RNNs.

Figure 1 shows a single cell in the LSTM network. In each cell, two states are inherited from the previous cell, namely, the hidden state \( h_{m-1} \) and the cell state \( c_{m-1} \). The hidden state is concatenated with the current input state \( x_m \) and processed by the designed structure. On the contrary, the cell state has received only a slight influence from the current input, which helps retain some more information from previous states. After processing the whole cell structure, the newly generated hidden state \( h_m \) and cell state \( c_m \) are transferred into the next LSTM cell.

The designed stacked structure of the LSTM network is comprised of three segments, as shown in Figure 2. In segment \( A \), the previous observed \( N^* \) states of the mobile robot are collected in an input matrix \( X_{r,k}^{N^*} \). In this study, this matrix is defined by a length of \( N^* = 20 \), \( X_{r,k}^{20} = [x_{r,0}, \ldots, x_{r,19}] \in \mathbb{R}^{20} \). Additionally, the dimensionality of the input state \( x_{r,m} \) at the \( m \)th time step is chosen to be two, namely, \( x \) and \( y \), since for the omnidirectional mobile robot, the heading of the robot has no significant influence on the motion tendency. Following the segment \( A \), a similar structure is stacked in the segment \( B \) for the purpose of predicting the potential state of the observed mobile robot. To extract the data features and improve the prediction, two fully connected layers with the Mish activation function \(^{20}\) are introduced in segment \( C \). Finally, the number of the prediction time horizon steps is also set as \( N^+ = 20 \) in this study, and the predicted trajectory of the near-future position is given by \( X_{r,k}^{N^+} \). Note that the length of the observation/prediction can be customized.
individually to meet the requirements of the application. They should, however, be tuned carefully, particularly for the length of the prediction time horizon $N^\ast$. Given a much higher $N^\ast$, the prediction time would be large and the prediction is unreliable. On the contrary, for a much shorter $N^\ast$, the prediction would be quite simple, but the resulting action would require a very short reaction time of the UAV. In this study, with the time difference $\Delta t$ between states set to 0.1 s, the selected length $N^\ast = N^* = 20$ corresponds to a reasonable observation/prediction time horizon of 2 s.

To train the neural network, the mobile landing platform is driven along several preprogrammed trajectories within the simulation environment for generating training data. These trajectories are expected to be typical trajectories for the mobile landing platform. For training a more general model, the training data are normalized before they are fed to the neural network. Furthermore, all parameters in the proposed network structure are initialized with small random numbers before the training phase, and the stochastic optimization method uses the Adam optimizer$^{21}$ with a learning rate of 0.001. The mean square error is taken as the loss function for comparing the predicted trajectory to the ground truth. After the training procedure is completed, all parameters in the proposed network are frozen, and the trained network is prepared to predict the position trajectory for the near future on the basis of the recorded $N^\ast$ previous steps of the past trajectory.

### 2.3 Combination of the LSTM network and the KF

Although the LSTM algorithm can be used directly to predict the trajectory of the moving platform, the most straightforward prediction approach is to use the current state to make an integration for the future. Rather than setting up a fixed velocity model in Ref. 12, in this paper, we estimate the current state using a KF and to calculate estimated future states by integrating the system dynamics with control input set to zero. The KF with subsequent time integration (KF + TI) and the LSTM method perform very similarly in a straight-line trajectory, as shown in Figure 3A. The LSTM method outperforms the KF + TI method during turning; see Figure 3B. It is worth noting that the prediction by KF + TI can only use the past measurements indicated by gray plus markers to obtain the KF results, which are marked by orange stars. By contrast, the tendency of the LSTM prediction marked with blue triangles corresponds to the target’s actual movement, and it performs better than the prediction marked with brown circles using the KF + TI method.

However, because the LSTM approach does not explicitly consider any a priori knowledge about the dynamics of the robotic platform, it is more likely to generate nonphysical trajectories. For instance, it may introduce unphysical discontinuities, jumping from the original trajectory; see Figure 3. To enhance the prediction performance, this study uses the predicted trajectory from the LSTM model as the incoming measurement resource for the KF. Based on Equation (5), one can update each estimated state of the mobile robot by

$$\hat{x}_{r,k} = \hat{x}_{r,k-1} + L_k (y_{LSTM,k} - \hat{y}_{r,k-1}).$$

where $y_{LSTM,k}$ contains the predicted positions of the LSTM network.

In Figure 3, green squares represent the predicted trajectory based on the proposed combination of the KF and the LSTM network (LSTM + KF). When compared to the results obtained directly from the LSTM network, it has a lower initial error and a smoother predicted trajectory. Additionally, the KF algorithm is capable of not only correcting the predicted position trajectory from the LSTM network but also forecasting the velocity at each time step. The predicted velocity at each time step aids in the design of the landing trajectory and the control of the UAV during the landing procedure. Using this prediction result, one can design an appropriate controller that minimizes both the landing position error and the velocity error simultaneously.
3 | EXPERIMENTAL SETUP

To verify the performance of the proposed algorithm, an experiment is set up in the simulation platform Gazebo\textsuperscript{22} and the sensor data are exchanged within the robot operating system\textsuperscript{23} environment. Figure 4 shows the UAV and the landing platform mounted on top of a UGV. The UAV is equipped with a downward-facing camera that enables it to detect the marker on the landing platform before initiating the landing procedure. The marker used in the experiment is an ArUco-type marker,\textsuperscript{24} which can simultaneously provide position and orientation relative to the marker center. The moving landing platform is 1 m × 1 m in size.

To simulate the typical condition of searching the moving platform, in which one can only obtain the position using wireless signals, such as GPS, at the initial stage, the position information of the landing platform is disturbed by a significant amount of noise in the simulation, and the UAV cannot land precisely using this position information alone. Thus, before the UAV detects the marker, the UAV approaches the moving landing platform using the predicted trajectory of the landing platform based on the noisy position information, while simultaneously searching for the marker with its onboard camera. After detecting the marker, the UAV lowers its altitude and attempts to land on this platform using the predicted pose trajectory. However, due to the limited angle of view of the used onboard camera, the UAV is unable to update the platform’s position throughout the landing procedure, especially when the distance between the UAV and the landing platform is less than a predetermined value. As a result, it can only land using the previously calculated trajectory prediction. Such practical issues add additional complexity to the situation.

During the simulated experiment, the UGV follows the designed trajectory, and its position is communicated to the UAV with an additional Gaussian noise $\eta \sim \mathcal{N}(0.0, 0.1)$ as shown in Figure 5, corresponding to a direct measurement by the UAV. Note that in the experiment, the UGV may have a similarly shaped trajectory at the data collection phase for training the LSTM network; however, it is not expected to have totally exact trajectories. Meanwhile, the ArUco marker detector verifies the image received from the camera mounted onboard. When the marker on the landing platform is detected, the onboard camera’s relative position and rotation with respect to the marker $x_{\text{marker}} \in \mathbb{R}^2$ is obtained, and it is combined with the current pose of the UAV $x_q \in \mathbb{R}^2$ and sent to the trajectory prediction node; otherwise, the simulated noisy position of the UGV is utilized by the trajectory prediction node directly. Depending on the various trajectory prediction methods described in Section 2, the predicted trajectory $X_{\text{traj}}^{N+}$ is then generated for the next $N$ steps in the time horizon. For the method based on LSTM only, the neural network can generate only the position of the landing platform, and the predicted state at each time step is in $\mathbb{R}^2$. On the other hand, the KF + TI and the LSTM + KF models can predict both position and velocity concurrently, which are in $\mathbb{R}^4$. As the flight controller, a nonlinear model predictive controller (MPC) is utilized and implemented with ACADO,\textsuperscript{25} which is building upon the work from Ref. 26. Finally, the UAV’s internal attitude controller receives the optimized thrust and the desired attitude control input $u_{\text{command}} \in \mathbb{R}^4$. On a laptop equipped with an Intel i7-6700HQ and an NVIDIA Quadro M1000, the MPC runs at a frequency of 100 Hz, and every 0.1 s, the proposed algorithm calculates the potential trajectory of the observed mobile landing platform over the next 2 s.

4 | SIMULATION RESULTS

4.1 | Scenario I

In the first scenario, the UGV has an identical trajectory shape in each test that travels a straight line along the $x$-axis at varying...
The discussed methods should predict not only the trajectory shape but also the corresponding time for a precise landing. The performance of each discussed method is shown in Figure 6. When the UGV moves slowly at 0.1 m/s, the UAV can successfully land on the platform even when no trajectory prediction is enabled. When the UGV’s velocity is increased to 0.3 m/s, the UAV’s mean landing error is already 0.54 m. When considering the size of the landing platform, this error is already beyond a safe landing. Obviously, as the UGV’s velocity increases, the landing error will increase without the aid of the trajectory prediction. In contrast, the landing errors for all three prediction methods are less than 0.3 m. In comparison to the KF + TI method, the two LSTM-based methods outperform it at all test speeds.

![Figure 6](image1.png)

**FIGURE 6** Landing errors at different velocities in Scenario I. KF, Kalman Filter; LSTM, long short-term memory; UGV, unmanned ground vehicle

![Figure 7](image2.png)

**FIGURE 7** Landing approaches in Scenario II: (A) Landing approaches in 3D; (B) the touchdown positions for UAV (red) and UGV (blue) by KF + TI (♦), LSTM (□), and LSTM + KF (☆); and (C) landing trajectories in the x-direction and (D) landing trajectories in the y-direction. KF, Kalman filter; LSTM, long short-term memory; TI, time integration; UAV, unmanned aerial vehicle; UGV, unmanned ground vehicle
4.2 | Scenario II

A smooth circular trajectory with a radius of 3 m is chosen as the second scenario. The UAV will land using one of the three trajectory prediction methods in this scenario. The approach trajectories of the UAV and the UGV are depicted in three dimensions in Figure 7A, and the touchdown positions of the UAV and the UGV for each method are indicated by an individual marker in Figure 7B, respectively. Each of the three methods successfully lands the quadrotor on the landing platform with an acceptable landing error, as shown in Table 1. Compared with the two LSTM-based approaches, the KF + TI method lands the UAV further away from the center of the landing platform. More details about the final landing trajectories in both x- and y-directions are shown in Figures 7C,D. If one contrasts the performance between the LSTM and the LSTM + KF methods, the UAV lands closer to the center of the landing platform with the LSTM + KF method, and it takes less time to approach the moving landing platform, since the LSTM + KF method predicts not only the predicted position but also its velocity.

4.3 | Scenario III

In the final scenario, an asymmetric eight curve is introduced to verify the performance of the three prediction methods. The trajectory is defined by

\[
\begin{align*}
x &= 4 + R \sin(f) \cos(f) \\
y &= 4 + R \sin(f)
\end{align*}
\]

where \( R = \begin{cases} 
4 & \text{for } f = \frac{t}{30}, \quad t \in [0, 30), \\
3 & \text{for } f = \frac{t-3}{25}, \quad t \in [30, 55].
\end{cases} \tag{7}
\]

The mobile landing platform starts from the crossing point in the center and follows the designed trajectory strictly as shown in Figure 8. The landing procedure will be triggered when the mobile landing platform reaches specified action points on its trajectory. In the comparison, six different action points are chosen to verify the performance of the proposed methods. The mobile landing platform's position at each action point is marked by red circles. When the landing platform reaches an action point, the landing procedure of the UAV is activated, and the UAV approaches the landing platform from its initial position \((0.0, 0.0, 0.3)^T\) m and attempts to land along the predicted trajectory. The landing performance of the different trajectory prediction methods is compared and shown in Table 2.

According to Equation (7), for \( t < 30 \), the designed trajectory has a larger radius, and the maximum velocity of the landing platform reaches 0.4 m/s. The KF + TI method shows a similar performance as in the previous scenarios, with the landing errors exceeding 0.3 m in the first three test phases. In comparison, the performance of the LSTM method is inconsistent, particularly when the approach direction of the UAV and the moving direction of the landing platform are significantly different, as in the third test phase, where it produces a slightly larger landing error than the KF + TI method. Throughout the rest of the designed trajectory, the radius is reduced to 3 m, and the landing platform's designed moving velocity is reduced. As a result, the KF + TI and the LSTM methods perform better in the final three test phases. Compared to the other two methods, in each test phase in Scenario III, the proposed LSTM + KF method shows the best performance, and the average landing error remains within 0.15 m.

5 | CONCLUSIONS

In this paper, a machine learning-based method is proposed for the precise landing of a UAV on a moving landing platform. Based on a trained LSTM network and a KF, the UAV attempts to predict the trajectory of the moving landing platform in real time. Then, it can land based on the prediction with the designed control scheme. The
The proposed method is evaluated and compared to conventional approaches in the simulation experiment conducted on the Gazebo platform. Three scenarios are set up, and the landing performance of each method is compared. The proposed method ensures that the UAV lands on the platform in all of the tested scenarios and shows relatively consistent performance. In all considered scenarios, the proposed combination LSTM + KF outperforms the other methods. In future work, one may apply moving horizon estimation, which will allow consideration of robotic platforms with nonlinear dynamics as well as constraints on the predicted trajectory. First hardware experiments have been prepared and are soon performed to get further insight into how the described approaches work in practice.

ACKNOWLEDGMENTS

This study was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant 433183605 and through Germany’s Excellence Strategy (Project PN4-4 Theoretical Guarantees for Predictive Control in Adaptive Multi-Agent Scenarios) under Grant EXC 2075-390740016. This study also benefited from the support of the China Scholarship Council (CSC, No. 20180800061) for Wei Luo.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

Research data are not shared.

ORCID

Wei Luo http://orcid.org/0000-0003-4016-765X
Henrik Ebel http://orcid.org/0000-0002-2632-6960
Peter Eberhard http://orcid.org/0000-0003-1809-4407

REFERENCES

1. Semsch E, Jakob M, Pavlicek D, Pechoucek M. Autonomous UAV surveillance in complex urban environments. IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Milan, Italy, September 15-18, 2009:82-85.

2. Zhang G, He Y, Dai B, et al. Grasp a moving target from the air: system control of an aerial manipulator. Paper presented at: IEEE International Conference on Robotics and Automation (ICRA), Munich, Germany; July 3-7, 2017:1393-1398.

3. Ardanaki ES, Ebel H, Eberhard P. Transporting an elastic plate using a group of swarm mobile robots. Paper presented at: IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Munich, Germany; July 3-7, 2017:1393-1398.

4. Ebel H, Eberhard P. Optimization-driven control and organization of a robot swarm for cooperative transportation. IFAC-PapersOnLine. 2019;52(15):115-120.

5. Ebel H, Luo W, Yu F, Tang Q, Eberhard P. Design and experimental validation of a distributed cooperative transportation Scheme. IEEE Trans Autom Sci Eng. 2021;18(3):1157-1169.

6. Acuna R, Zhang D, Willert V. Vision-based UAV. Vision-based UAV landing on a moving platform in GPS denied environments using motion prediction. Paper presented at: Latin American Robot Symposium, Brazilian Symposium on Robotics (SBR) and Workshop on Robotics in Education (WRE), Joao Pessoa, Brazil; November 6-10, 2018:515-521.

7. Allende H, Valle C. Ensemble methods for time series forecasting. In: Seising R, Allende-Cid H, eds. Claudio Moraga: A Passion for Multi-Valued Logic and Soft Computing. Springer International Publishing; 2017:217-232.

8. Yan W, Qiu H, Xue Y. Gaussian process for long-term time-series forecasting. Paper presented at: International Joint Conference on Neural Networks, Atlanta, USA; June 14-19, 2009:3420-3427.

9. Luo W, Xiao Z, Ebel H, Eberhard P. Stereo vision landing on a moving platform in GPS denied environments using deep RNN odels and WFST-based decoding. Paper presented at: IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Scottsdale, USA; December 13-17, 2018:167-174.

10. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735-1780.

11. Dey R, Salem FM. Gate-variants of gated recurrent unit (GRU) neural networks. Paper presented at: 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, USA; August 6-9, 2017:1597-1600.

12. Luo W, Eberhard P. Optimization-based trajectory prediction enhanced with goal evaluation for omnidirectional mobile robots. Paper presented at: 18th International Conference on Informatics in Control, Automation and Robotics (ICINCO); Online Streaming; July 6-8, 2021:263-273.

13. Ebel H, Eberhard P. Distributed decision making and control for cooperative transportation using mobile robots. Paper presented at: 9th International Conference Advances in Swarm Intelligence (ICSI); Shanghai, China; July 18-20, 2018:89-101.

14. Luo W, Xiao Z, Ebel H, Eberhard P. Stereo vision-based autonomous target detection and tracking on an omnidirectional mobile robot. Paper presented at: 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO); Prague, Czech Republic; July 29-31, 2019:268-275.

15. Simon D. Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. Wiley-Interscience; 2006.
25. Houska B, Ferreau H, Diehl M. ACADO toolkit—an open source framework for automatic control and dynamic optimization. Optim Control Appl Methods. 2011;32(3):298-312.

26. Kamel M, Stasny T, Alexis K, Siegwart R. Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system. In: Koubaa A, ed. Robot Operating System (ROS): The Complete Reference. Vol 2. Springer International Publishing; 2017:3-39.