

Teaching a Drone to Accompany a Person from Demonstrations using Non-Linear ASFM

Anaís Garrell, Carles Coll, René Alquézar and Alberto Sanfeliu

Abstract—In this paper, we present a new method based on the Aerial Social Force Model (ASFM) to allow human-drone side-by-side social navigation in real environments. To tackle this problem, the present work proposes a new nonlinear-based approach using Neural Networks. To learn and test the rightness of the new approach, we built a new dataset with simulated environments and we recorded motion controls provided by a human expert tele-operating the drone. The recorded data is then used to train a neural network which maps interaction forces to acceleration commands. The system is also reinforced with a human path prediction module to improve the drone’s navigation, as well as, a collision detection module to completely avoid possible impacts. Moreover, a performance metric is defined which allows us to numerically evaluate and compare the fulfillment of the different learned policies. The method was validated by a large set of simulations; we also conducted real-life experiments with an autonomous drone to verify the framework described for the navigation process. In addition, a user study has been realized to reveal the social acceptability of the method.

I. INTRODUCTION

In recent years, the field of mobile robotics has seen an increased interest in the deployment of Unmanned Aerial Vehicles (UAV) in many domains of application, motivated by the recent technological advances. Considerable success has been achieved in many purposes, such as search and rescue [11], exploration [1], or mapping [13]. However, obstacle avoidance and navigation are still crucial hurdles. Concretely, the interest in human-robot side-by-side formation, in order to allow robots to accompany people, is increasing [4], [5].

This kind of functionality requires robots to interact with humans, as well as certain degree of knowledge about people’s intentions. Furthermore, as in any other mobile robot application, collision avoidance is another important requirement. This particular problem has been addressed by several authors in the case of ground robots [8]. However, in the aerial robots field, this problem is mostly unexplored.

Moreover, there has been a general shift in the nature of the proposed solutions to the problem of robot navigation, from hand-crafted engineered solutions [12] to data based solutions [6]. This trend is motivated by the recent advances

in Machine Learning, which have produced remarkable achievements in other fields, e.g. Computer Vision and Natural Language Processing [10]. Thus, it is natural to find an increasing number of solutions to the navigation task that integrate the last advances in machine learning.

In the present work, we propose a new method, called Non-linear Aerial Social Force Model, that allows autonomous flying robots to accompany humans in urban environments in a safe and comfortable manner. The approach is completely based on experimental data that is used to learn a motion controller. The system is based on an extension of the Aerial Social Force Model paradigm [5] and it also includes two additional features: a human path prediction module and a collision detection module.

Following the ASFM, the interaction of the different elements in the environment, i.e. obstacles, main human and other pedestrians, are modeled as forces applied to the aerial robot. But instead of computing the resulting force as a linear combination governed by a set of hand crafted weights, an interaction force model is learned from data. This data consists of many recorded trajectories provided by a human expert, so that after training, the learned model is able to mimic the control signals provided by a human.

Furthermore, a human path prediction module is integrated into the system to provide the motion controller some notion about the human’s position in the near future. This information allows the controller to anticipate to human’s trajectory and results in a more proactive drone’s behaviour. A collision detection module based on neural networks is also integrated into the system.

Finally, the learned motion controller is evaluated on simulated environments and real-life experiments. Since the verification of man-in-the-loop systems is fuzzy, a new quantitative metric is defined to asses the level of performance of the learned policy.

In Fig. 1 a schematic description of the different modules of the new model is presented.

The remainder of the paper is organized as follows. Section II introduces the new Non-Linear Aerial Social Force Model (ASFM) and its main components. In Section III, the Data collection is described. Section IV introduces the exhaustive validation of the system in diverse simulations and real-life experiments. Finally, Sections V and VI provide the user study and conclusions, respectively.

Work supported by the Spanish Ministry of Science and Innovation under project ColRobTransp (DPI2016-78957-RAEI/FEDER EU), and European Project GAUSS (H2020-GALILEO-GSA-2017-1-776293) and by the Spanish State Research Agency through the Maria de Maeztu Seal of Excellence to IRI (MDM-2016-0656).

The authors are with the Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens Artigas 4-6, 08028 Barcelona, Spain. {agarrell, ccoll, ralquezar, sanfeliu}@iri.upc.edu

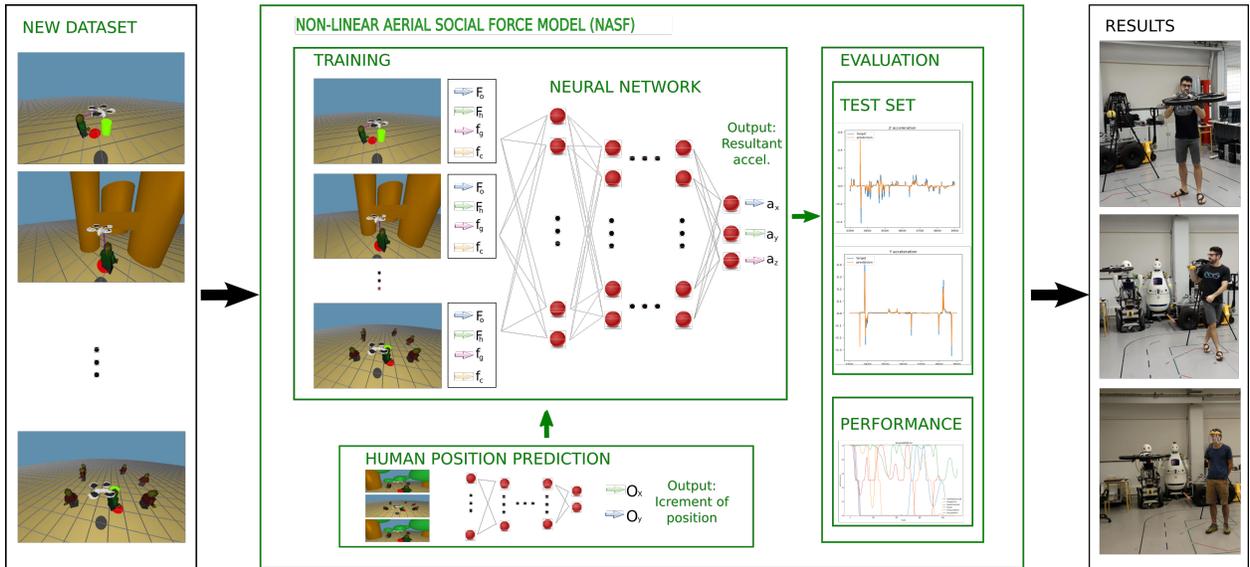


Fig. 1: **Non-linear Aerial Social Force Model (NASF)**: Diagram of the proposed method and its stage, (i) creation of the new dataset to train the model, (ii) components of the NASF model and its evaluation: Non-trained Aerial Social Fore model and Human path predictor (iii) obtained results.

II. NON-LINEAR AERIAL SOCIAL FORCE MODEL

In this section, we proceed to describe our method. The proposed system aims to allow human-drone side-by-side social navigation in real environments, and we base our approach on a 3D extension of the Social Force Model presented in [2], [5]. As a result, all drone's interactions with the different elements of the environment are integrated into four different forces. These forces represent four of the five features that the learning model uses as input in order to predict the acceleration commands that must be applied to the drone and are defined in detail in the following subsection. The remaining subsections are dedicated to explaining all the components that form the presented model so-called Non-linear Aerial Social Force Model (NASF): the non-linear regressor that predicts acceleration commands, the quantitative metric of performance, the human path prediction module and the collision detection module.

A. Feature Definition

Social Force Model states that changes in behavior/intentions (trajectory) can be explained in terms of social fields or forces, and the final motion can be expressed through a function of the pedestrians positions and velocities [2]. The total force applied to the robot comes from three separate components: the robot-humans and robot-objects interaction forces, which are repulsive, and the goal attraction force, which make the robot stay closer to the human being accompanied. In the Non-Linear Social Force Model, these forces are incorporated to the set of input features which also include two new features: a human feature and the instantaneous drone velocity. It is important to note that now all features are expressed as 3D vectors because the drone can move freely in all three dimensions.

1) *Static Object Repulsive Feature*: The static object repulsive feature provides information about the relative location of static objects with respect to the drone, and it also offers information about their proximity. It takes the form of a global force that aggregates all the individual interaction forces

$$\mathbf{F}_o = \sum_{o=1}^O \mathbf{f}_o \quad (1)$$

where O is the set of detected objects and \mathbf{f}_o is defined as a non-linear function of the distance of each detected static objects in the environment.

$$\mathbf{f}_o = \frac{\mathbf{P}_o - \mathbf{P}_R}{\|\mathbf{P}_o - \mathbf{P}_R\|} A_{R,o} e^{\left(\frac{d_{R,o} - d_{R,o}}{B_{R,o}}\right)} \quad (2)$$

where \mathbf{P}_o and \mathbf{P}_R are the positions of the object and the drone, respectively, and $d_{R,o}$ is the distance between the robot and the object. $A_{R,o}$, $B_{R,o}$ and $d_{R,o}$ are fixed parameters that govern the module of the resulting force, the value of the parameters were previously described in [5].

2) *Pedestrian Repulsive Feature*: Pedestrians can be considered as dynamic obstacles, but humans have personal space, and intrusions into this area by external objects may cause discomfort to the person, especially if the object is a flying robot. Thus, we define a separate feature to model, the repulsion exerted on the drone from pedestrians. The pedestrian repulsive feature is again a force defined similarly to the static object repulsive force,

$$\mathbf{F}_h = \sum_{h=1}^H \mathbf{f}_h \quad (3)$$

where H is the set of detected pedestrians and \mathbf{f}_h is defined as,

$$\mathbf{f}_h = \frac{\mathbf{P}_h - \mathbf{P}_R}{\|\mathbf{P}_h - \mathbf{P}_R\|} A_{R,h} \frac{v_h}{t_h} e^{\left(\frac{d_{R,h} - d_{R,h}}{B_{R,h}}\right)} \quad (4)$$

\mathbf{P}_h and \mathbf{P}_R are the positions of the pedestrian and the drone respectively, and $d_{R,h}$ is the distance between the robot and the object. $A_{R,h}$, $B_{R,h}$ and $d_{R,h}$ are fixed parameters that govern the module of the resulting force. Furthermore, the term v_h/t_h is introduced to modulate the force in such a way that the pedestrian h is able to stop in time t_h , according to [14].

3) *Goal Feature*: In order to obtain a flying behaviour in which the drone is moving in a side-by-side formation with the human, and it does not navigate behind the person, the drone needs some notion of the path the human is following. The people path prediction module is responsible for computing a predicted future human's position, which is used to build the goal attractive feature. This feature is defined as the vector from the drone position to the goal (the predicted future human position),

$$\mathbf{f}_g = \mathbf{P}_g - \mathbf{P}_R \quad (5)$$

where P_R is the drone's position and P_g is the estimated human position for one second into the future. This feature is not expressed as a force, but still it provides the necessary information to allow the drone to fly next to the main human.

4) *Human Feature*: Finally, a new feature is defined, it provides information about the current position of the human. This feature combined with the goal feature encode information about the current position of the human and the expected direction of movement, which allows the drone to place itself in the right spot for a natural accompaniment configuration. The human feature is defined similarly to the goal feature, as a vector from the drone position P_R to the main human position P_h .

$$\mathbf{f}_c = \mathbf{P}_c - \mathbf{P}_R \quad (6)$$

B. Non-Linear Regressor

Previous research works on Social Navigation based on Artificial Potential Fields [2], [5] extract forces from the elements present in the environment, and linearly combine them to obtain a resulting force, that is applied to the robot, and, thus, determines its acceleration. In this work, we attempt to substitute this linear approach by a non-linear model learned from data, which is composed of recordings of expert trajectories. We want to teach a dense neural network to imitate the flying controls of a human expert, so that the resulting behaviour is more accurate and natural from a human perspective. By learning the model from data, we also avoid to tune the parameters present in previous approaches.

The non-linear model used to learn the expert plying policy is a fully connected 5 layer dense neural network. The input layer has 15 neurons, one for each input dimension, and the output layer has 3 neurons, one for each component of a linear acceleration in a 3D space. Dropout regularization technique is used to avoid overfitting. Dropout works by

randomly canceling neurons in a layer (governed by a probability parameter) during training to make the network less dependent on certain neurons or inputs. The probability values defined for the network are, from input to output layer, 0.1 and 0.1. Regarding the parameters used during the training phase of the networks, the mean squared error was the chosen loss function and Adam was used as the gradient descend optimizer.

We desire our model to be able to profit from non-linear relations among the forces but also consider the temporal correlation between consecutive samples, as predicted the acceleration at a certain point has an effect on future samples. We estimate the intention to react to a certain event must happen during the second before the reaction event. Therefore, we arrange consecutive samples into 1s windows.

C. Quantitative Metric of Performance

To properly evaluate the learned control policies, the Non-linear Aerial Social Force Model defines a new quantitative metric of performance inspired by the metric presented in [5], which is based on the idea of proxemics [7].

To define the metric used in the present work, four different areas have been defined: (i) People's Personal space \mathcal{C}_i , it is required the drone does not perturb the human's personal spaced, in order to obtain a socially accepted navigation by the person being accompanied. (ii) Social distance area \mathcal{A} , the drone must navigate considering an accepted social distance. (iii) During the accompanied task, the drone should be in the human's field of view, the drone should navigate side-by-side with the person being accompanied, \mathcal{B} . (iv) Finally, robot should consider other pedestrians in the environment p_j , the drone is not allowed to perturb pedestrians' personal space $\bigcup_{p_j} \mathcal{C}_j$.

$$\begin{aligned} \mathcal{A} &= \{x \in \mathbb{R}^3 \setminus (\mathcal{B} \cup \mathcal{C}) \mid d(x, p_i) < 3\} \\ \mathcal{B} &= \{x \in \mathbb{R}^3 \setminus \mathcal{C} \mid d(x, p_i) < 3\psi(\varphi_{p_i}, \theta_{p_i})\} \\ \mathcal{C} &= \{x \in \mathbb{R}^3 \mid d(x, p_i) < \psi(\varphi_{p_i}, \theta_{p_i})\} \end{aligned} \quad (7)$$

where $\psi(\varphi_{p_i}, \theta_{p_i})$ represents the anisotropic factor in the 3-dimensional space, it is defined by:

$$\psi(\varphi_{p_i}, \theta_{p_i}) = w(\varphi_{p_i}) \cos(\theta_{p_i})(h + \xi_{p_i} w(\varphi_{p_i}) \eta) \quad (8)$$

it depends on φ_{p_i} , an angle formed between the desired velocity of the pedestrian p_i and the vector \mathbf{r}_{R_j} , indicating the distance between the robot and the pedestrian p_i and pointing to him, and θ_{p_i} , which is the angle between the x and z coordinates, an angle formed between the position of the robot and the pedestrian. Moreover, h is the height of the pedestrian, η is a constant defined by us as 1.25, after performing experiments with non-trained volunteers.

Furthermore, we should define $w(\varphi_{p_i})$:

$$w(\varphi_{p_i}) = \lambda_{p_i} + (1 - \lambda_{p_i}) \left(\frac{1 + \cos(\varphi_{R_j})}{2} \right) \quad (9)$$

Where λ_{p_i} defines the strength of the anisotropic, and $\cos(\varphi_{R_j})$ is calculated as:

$$\cos(\varphi_{Rj}) = -\mathbf{n}_{Rj} \cdot \mathbf{e}_{p_i} \quad (10)$$

using \mathbf{n}_{Rj} , the normalized vector pointing from the robot to p_i , it describes the direction of the force, and e_{p_i} is the desired motion direction of the pedestrian p_i (which is pointing to the goal).

Moreover, the flying robot has been represented as $\psi(\varphi_{Rj}, \theta_{Rj})$, its anisotropic factor:

$$\psi(\varphi_{Rj}, \theta_{Rj}) = w(\varphi_{Rj}) \cos(\theta_{Rj}) \quad (11)$$

Thus, we can define the following function which evaluates the behavior of the drone considered the volumes described above:

$$\rho(r, p_i) = \int_{(\mathcal{B} \setminus \cup_{p_j} \mathcal{C}_j) \cap \mathcal{R}} \frac{d\mathbf{x}}{|\mathcal{R}|} + \int_{(\mathcal{A} \setminus \cup_{p_j} \mathcal{C}_j) \cap \mathcal{R}} \frac{d\mathbf{x}}{2|\mathcal{R}|} \in [0, 1] \quad (12)$$

Where $\mathbf{x} \in \mathbb{R}^3$. If the complete area of the robot is allocated in zone \mathcal{B} , then, performance would get the maximum value. As the robot moves far from the person and enters to zone \mathcal{A} , the performance decreases until 1/2. Finally, the performance in zones \mathcal{C}_i is 0, as it is not allowed that the drone enters in people's personal space.

D. Human Path Prediction Module

The Non-linear Aerial Social Force Model includes a module that predicts the future position of the main human, similarly to [5]. The main difference lies in the use of a five layer neural network, which takes the last ten known positions of the main human, and it infers the increment in the current position one second into the future. The predicted increment in position is then used to compute the Goal Feature, which provides a notion of the path the human is following, and allows the drone to fly in a side-by-side formation with the human.

To train the model, a dataset of human paths was built from all the recorded expert trajectories. Each feature vector of each sample in the dataset is made of 10 consecutive positions of the human, expressed as increments, so that they belong to the same distribution. The corresponding target of the feature vector is the increment in position observed in the consecutive 30 frames of simulation. The global dataset is then split into three subsets: the train set, which contains 46,468, the validation set, containing 11,592 samples, and the test set, composed of 12,056 samples. The validation set is used to tune the hyperparameters of the neural network. After training the model on the described dataset, the model is able to predict the position of the main human one second into the future with an mean squared error over the test set of 0.01969.

E. Collision Detection Module

The collision detection module was not present in the original Aerial Social Force Model, so it is a new addition in the Non-linear Aerial Social Force Model. The main reason

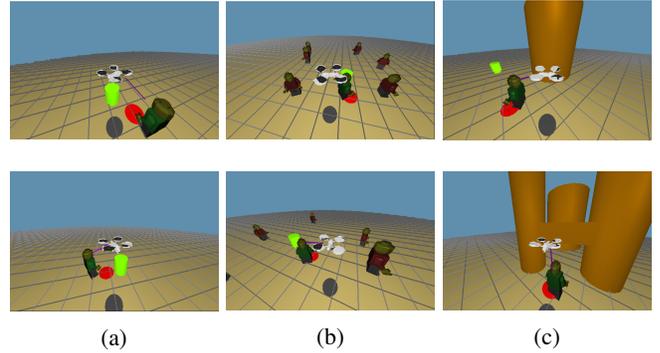


Fig. 2: **Simulated expert trajectories** (a) shows two pictures of ‘Open environments’, column (b) shows type ‘Crowded environments’ and column (c) shows type ‘Cluttered Environments’ environments.

behind the integration of such a mechanism into the system is to avoid potential collisions that may happen at run time, especially if the drone reaches states that are not seen during training. Following the principles of the present work, and supported by the evidence provided in [3], we decided to learn the collision detection model from data.

A two layer neural network was trained with data collected from collision trajectories. We created 600 simulated environments densely populated with obstacles and configured the drone to fly in a straight line until a collision happened. We recorded all the data from the simulated trajectories and built a dataset by labeling the samples automatically. Each sample in the dataset corresponds to a given frame in a given recorded trajectory, and it contains the static object repulsive feature and the drone velocity. The labeling procedure works as follows: the set of samples that are less than one second distant to a collision are labeled with a 1, and the remaining samples are labeled with a 0.

III. DATA COLLECTION

In the Non-linear Aerial Social Force model, we aim to learn a motion controller for a drone that behaves similarly to the controls provided by a human expert. Therefore, in order to teach our model to behave similarly to a human, we need a significant amount of expert demonstrations that will be later used to build a dataset by extracting the relevant features, this dataset will be then used to train the non linear model, and hopefully the trained model will fulfill our requirements. Thus, the first step to learn a flying control policy is to obtain human expert demonstrations, and due to the lack of this kind of data, we obtain the demonstrations ourselves. It has to be mentioned, the information that we desire to capture with the tele-operated behavior by an expert is the intention of the robot's movement, not the instantaneous changes, that is why a time window is used in the training stage.

Simulated environments have been used to generate the expert demonstrations because they are safe, easy to run and allow complete freedom of environment design, compared to gathering the data from a real drone. The software that

was used to build and run the simulator is FreeGLUT¹. The simulator has an update frequency of 30Hz and each cycle it generates an output file that contains the position of all the elements in the simulated environment, as well as, the velocity and acceleration of the drone. In each simulated environment, the human expert controls the acceleration of the drone with a PS3 controller and tries to fly next to the main human, which is autonomously controlled. Therefore, each recorded trajectory corresponds to a simulation run, and is composed of the set of output files generated in the episode.

Regarding the configuration of the simulated environments that are used to record expert trajectories, we defined three different types of environments, which belong to different levels of complexity. The simplest type of environments are named ‘Open Environments’ and only contain the drone and the main human. The second type of environment called ‘Crowded environments’ contains the drone, the main human and several pedestrians. Finally, the set of environments called ‘Cluttered Environments’ contain the drone, the main human and static obstacles, like pillars & bridges. 260 expert trajectories were recorded in type ‘Open Environments’, 180 were recorded in type ‘Crowded Environments’ and 270 were recorded in type ‘Cluttered Environments’ environments. All the recorded expert demonstrations account for 326 minutes of simulation. Fig. 2 shows several pictures for the different types of simulated environments.

In order to build the final dataset that is used to train the neural network, the defined features are extracted for each simulation cycle in each recorded trajectory and put together into a single dataset. The target associated to each feature vector corresponds to the acceleration of the drone at the simulation cycle the feature vector belongs to.

IV. EXPERIMENTS

The previously defined dataset was split into train, validation and test sets. This division was required because we initially proposed three different model, a linear regressor, a five layer neural network and a three layer neural network. We trained all the models on the training set, then, we used the validation set to adjust hyperparameters of the neural networks and, finally, we tested the models’ average mean squared error on the test set. The results are listed in table I. These results provide an indicator of how well the models predict a control action given a certain state of the environment. However, this is not a good indicator of how well the learned policy behaves in reality, because the problem of learning a control policy from demonstrations violates the i.i.d. assumption, in the sense that the model’s prediction modify the distribution of the input, as new states of the environment depend on previous control actions. Therefore, to really asses the quality of the learned policies, we must test them on simulated environments using the performance metric described in section II-C.

¹<http://freeglut.sourceforge.net>

	Train MSE	Validation MSE	Test MSE
Linear Regression	0.7297e-3	0.6969e-3	0.6918e-3
3-Layer NN	0.7025e-3	0.6812e-3	0.6770e-3
5-Layer NN	0.7017e-3	0.6796e-3	0.6764e-3

TABLE I: MSE achieved by the models on different sets.

	Pillars & Bridges	Forest
5-Layer NN with CDM	0.7581	0.7256
5-Layer NN without CDM	0.5459	0.5196

TABLE II: Final performance average achieved by each model after simulating the learned policies on *Pillars and Bridges* environments and *Forest*, the enhancement of the system using the Collision Detection Module (CDM) is presented.

A. Simulations

The learned flying policies were tested on several test environments of two new unseen types, called *Pillars & Bridges* and *Forest*. These environments are more complex than those used to train the models, as Fig. 3 shows. To compute the performance of a model in a test environment, the test environment is simulated and the trained model is used to compute the drone intentions during the accompanied task. To obtain the prediction an input must be fed to the model, which is built from the data generated in the previous simulation cycle. This way, the model takes control of the motion of the drone, in the same way the human expert is controlling the drone when expert trajectories are recorded.

The performance metric is computed in each cycle of a test simulation, and thus, the final performance of a certain model in a certain environment is obtained by averaging the instantaneous measurements of performance along a test simulation run on that environment.

To further illustrate the behaviour of the learned policies, Fig. 3 shows the values of performance for each frame in a test simulation achieved by each model. The quantitative results allowed us to choose the five layer neural network model as the final controller for the drone, which was later implemented on a real drone for real life experiments. It can be seen, as the density of pedestrians is increased the performance of the system gets lower.

Finally, in order to evaluate if the collision detection module (CDM) enhances the performance of the system, we also compared the drone’s behavior using the collision detection or without using it, Table II introduces the obtained results (average), it can be seen the CDM increases the performances of the system.

B. Real-life Experiments

For the real experiments we have used a quadcopter built by Parrot, the AR.Drone 2.0, on which we tested the policy learned by the three layer neural network. We also have made use of the Optitrack Motion Capture system, created by NaturalPoint Inc, which provides all the necessary information about absolute positions of all the elements in the environment that are properly marked. The controller

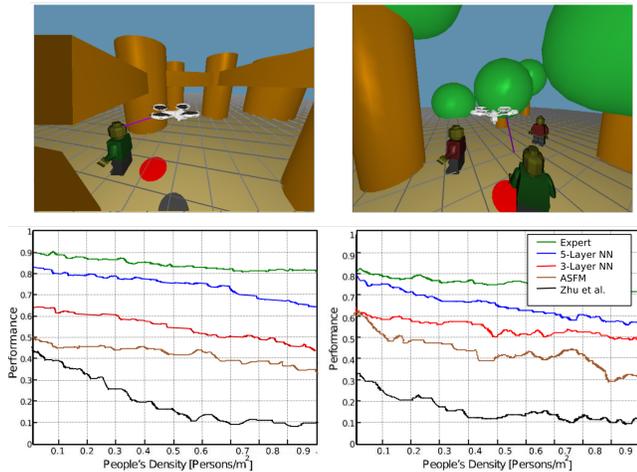


Fig. 3: **Synthetic experiments.** *Top row:* Test environments, the picture in the left belongs to a *Pillars & Bridges* environment, the picture on the right is a *Forest* environment. Note that the tree obstacles in the forest are not seen during the training stage. *Bottom row:* Performance presented previously. All results are function of the pedestrian density in the environment.

of the drone was implemented as ROS node. With the information provided by Optitrack, the controller node can build the inputs of the model to get the predicted acceleration commands for the drone. The drawback of using Optitrack to accurately locate all the elements in the environment is that the working area is limited to 5x5m, which imposes severe constraints to the movement of the drone and the main human.

In order to enhance the database, we added some real-life trajectories, where the drone tele-operated by an expert accompanied different volunteers, we then, re-trained our 5-layer Neural Network, and we used this new model to perform the real experimentation.

In first place, we run several experiments in the simplest scenario, where only the main human and the drone are present. During the experiments we observed that the drone is able to approach the human if the take of point is far, and once it reaches a certain distance to the human it stops and maintains the position in the air. If the human starts moving, the drone is able to follow the person while keeping a safe distance.

In the second set of experiments a pillar obstacle was introduced into the scenario. We observed that the drone was still able to follow the main human maintaining a safe distance, and additionally it successfully avoided the pillar if it was found on the drone’s way. It is important to mention, however, that due to the limitations of the experiment setup it was not possible to fully replicate in reality the simulated results, and we had to run very simplified experiments. Fig. 4 shows the real-life experiments, the drone moved and was able to accompany a person.

We would like to address the reader to see videos and additional results of the real-life experiments, in the BRL, in the

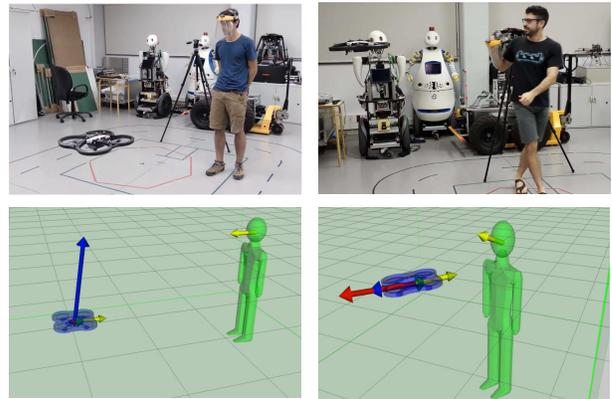


Fig. 4: **Real-life experiments:** Validation of the model in a real world environment.

following link <http://www.iri.upc.edu/people/agarrell/asfm-iros2019.html>

V. USER STUDY

The results presented in the previous section demonstrate that the drone is able to accompany people in different environments, and this behavior has been learned from expert trajectories. A user study was also conducted to determine whether the non-linear ASFM presented previously are perceived by people as socially appropriate. Finally, we concluded this section by studying how our social navigation enhances a follower approach, wherein the robot only follows the person’s trajectory, without considering any social conventions, and we should highlight that people perceived a difference between these two approaches.

The hypothesis we endeavored to test was as follows: “Participants will not perceive a difference between the Non-Linear ASFM and the tele-operated drone by and expert.”

For the experiments, we selected 35 people (20 men, 15 women) on the University Campus. Participants ranged in age from 21 to 53 years ($M=29.13$, $SD=15.23$), and represented a variety of university majors and occupations including computer science, mathematics, biology, finance and chemistry. For each individual selected, we randomly activated one of the two drone’s behaviors to accompany the volunteer. It should be mentioned that none of the participants had previous experience working or interacting with robots.

Participants were asked to complete a variety of surveys. Our independent variables considered whether the drone autonomous accompanied a person (using the Non-linear ASFM) or was tele-operated. The main dependent variables involved participants’ perceptions of the comfortableness and intellectual characteristics. Each of these fields, was evaluated by every participant using a questionnaire to fill out after the experiment, based on [9].

Participants were asked to answer a questionnaire, following their encounter with the drone in each mode of behavior. To analyze their responses, we grouped the survey questions into three scales: the first measured overall robot behavior, while the second and third evaluated more specific

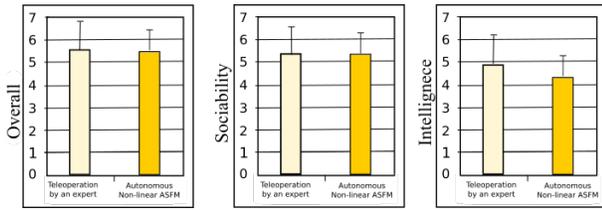


Fig. 5: **HRI Navigation Results.** Degree of acceptance of the robot navigation using Non-linear ASFM. **Left:** Global evaluation of the two navigation. **Center:** Robot’s sociability. **Right:** Robot’s intelligence, as perceived by the humans.

questions on the robot’s movement. Both scales surpassed the commonly used 0.7 level of reliability (Cronbach’s alpha).

Each scale response was computed by averaging the results of the survey questions comprising the scale. ANOVAs were run on each scale to highlight differences between the three robot behaviors.

Below, we provide the results of comparing the two different behaviors. Human perception has been studied in the navigation skill. To analyze the source of the difference, three scores were examined: “overall”, “robot’s sociability” and “robot’s intelligence”, plotted in Fig. 5. For the global evaluation score plotted in Fig. 5-Left, pairwise comparison with Bonferroni demonstrate there were no difference between the two kind of navigation approaches, $p = 0.36$. In terms of robot’s sociability and intelligences the volunteers also did not perceived a difference between the two navigation, $p = 0.18$ and $p = 0.27$, respectively.

Therefore, after analyzing these three components in navigation terms, we may conclude that the drones navigation learned from expert trajectories was socially accepted by volunteers.

VI. CONCLUSIONS

In this paper, we present a new solution to the robot companion task for the case of a UAV that flies in a side-by-side formation with a human. The presented approach is inspired by the Aerial Social Force Model but it integrates novel techniques from the Machine Learning field to learn a flying control policy from expert demonstrations. A dataset is built by recording the control actions provided by a human expert in simulated environments. Three different learning models are trained on the dataset and their performance is assessed on simulated test environments in order to select the best one. Finally, the best model is implemented on a real drone and real-life experiments are performed.

The test simulations show that all the considered models are able to learn control policies that resemble the one provided by the expert, with slightly different degrees of accuracy. From the real experiments, we conclude that the learned policy has the potential to provide an accurate, safe and human-like drone motion, but due to the limitations of the setup, we cannot fully replicate the results from the test simulations. Finally, a user study showed the acceptance of the method by inexperienced people, the method was compared

with a teleop performance and they found no statistical difference between the two approaches.

REFERENCES

- [1] Kyle Cesare, Ryan Skeelee, Soo-Hyun Yoo, Yawei Zhang, and Geoffrey Hollinger. Multi-uav exploration with limited communication and battery. In *IEEE International Conference on Robotics and Automation*, pages 2230–2235, 2015.
- [2] Gonzalo Ferrer, Anaís Garrell Zulueta, Fernando Herrero Cotarelo, and Alberto Sanfeliu. Robot social-aware navigation framework to accompany people walking side-by-side. *Autonomous robots*, 41(4):775–793, 2017.
- [3] Dhiraj Gandhi, Lerrel Pinto, and Abhinav Gupta. Learning to fly by crashing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3948–3955. IEEE, 2017.
- [4] Anaís Garrell, Michael Villamizar, Francesc Moreno-Noguer, and Alberto Sanfeliu. Teaching robots proactive behavior using human assistance. *International Journal of Social Robotics*, 9(2):231–249, 2017.
- [5] Anaís Garrell Zulueta, Luís Alberto Garza Elizondo, Michael Alejandro Villamizar Vergel, Fernando Herrero Cotarelo, and Alberto Sanfeliu Cortés. Aerial social force model: A new framework to accompany people using autonomous flying robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 7011–7017, 2017.
- [6] Alessandro Giusti, Jérôme Guzzi, Dan C Ciresan, Fang-Lin He, Juan P Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen Schmidhuber, Gianni Di Caro, et al. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667, 2016.
- [7] Edward Twitchell Hall. *The hidden dimension*, volume 609. Garden City, NY: Doubleday, 1966.
- [8] Daniel Hennes, Daniel Claes, Wim Meeussen, and Karl Tuyls. Multi-robot collision avoidance with localization uncertainty. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 147–154, 2012.
- [9] Rachel Kirby. *Social robot navigation*. PhD thesis, Carnegie Mellon University, The Robotics Institute, 2010.
- [10] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5528–5531, 2011.
- [11] Yogianandh Naidoo, Riaan Stopforth, and Glen Bright. Development of an uav for search & rescue applications. In *AFRICON*, pages 1–6. IEEE, 2011.
- [12] Emrah Akin Sisbot, Luis F Marin-Urias, Rachid Alami, and Thierry Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5):874–883, 2007.
- [13] Christoph Strecha, Olivier Küng, and Pascal Fua. Automatic mapping from ultra-light uav imagery. In *EuroCOW 2012*, 2015.
- [14] Francesco Zanlungo, Tetsushi Ikeda, and Takayuki Kanda. Social force model with explicit collision prediction. *EPL (Europhysics Letters)*, 93(6):68005, 2011.