Design of dynamics invariant LSTM for touch basedhuman multirotor interaction

First Author[1,2*], Second Author[2,3†] and Third Author[1,2†]

[1*]Department, Organization, Street, City, 100190, State, Country.
[2]Department, Organization, Street, City, 10587, State, Country.
[3]Department, Organization, Street, City, 610101, State, Country.

*Corresponding author(s). E-mail(s): iauthor@gmail.com;
Contributing authors: iiauthor@gmail.com; iiiauthor@gmail.com;
†These authors contributed equally to this work.

Abstract

We propose a method to detect force signature from a human interaction with a multi rotor and use it as a touch based communication mode with the multi rotor. It has been shown through an experiment that if the drone seems safe to touch by the participant, the 58% of the participants will prefer to communicate with the drone by touch. Abtahi, Zhao, E., and Landay (2017) Our proposed work will allow us to detect human interaction using only the sensors that are already on board a multi rotor and without requiring explicit force estimation. The detection is done using LSTM based deep learning algorithms. We were able to achieve an accuracy of 96% with our model. The detection is demonstrated in a handover experiment where the multi rotor must handover a payload to the human if there is a human interacting with the multi rotor. The experiment can be extended to carry any payload and can be exploited in applications where multi rotor and humans have to work together.

Keywords: physical human-robot interaction, LSTM, Deep Learning, Payload delivery

1 Introduction

There is an interest in using alternative ways of communicating with an unmanned aerial vehicle (UAV). Traditionally, a radio controller was used to control a UAV. Improvements in sensors and controllers of UAVs have allowed us to have autonomous UAVs that can achieve tasks without the intervention of humans. Even autonomous UAVs need to have a human in the loop for high level task allocation, Once the task is allocated, then the platform will autonomously achieve the task. Recently, alternative modes of communication, where humans can control the UAV with gestures and voice commands, have been explored Abtahi et al. (2017); Delfa et al. (2020). New modes of communication with the UAV platform will allow us to use the best applicable mode for the scenario. In this paper, we will discuss a multirotor planar vertical take-off and landing (VTOL) UAV, that can communicate with human physical interaction for task allocation. The task that is allocated can be anything that requires close proximity to the user.

Physical interaction is explored widely for robotic arms. There are already instances where humans work with robotic arms for strength augmentation and assistance Santis, Siciliano, Luca, and Bicchi (2008). On Robotic arm, physical interaction is detected using sensors like tactile sensors in Singh et al. (2018) or vibrations passing through a payload in Eguíluz, Rañó, Coleman, and McGinnity (2019). Due to the mobility offered by a multirotor platform compared to a robotic arm, we could see applications that require physical interaction with a multirotor platform. Since a physical interaction will exert some force on a robotic platform, some robotic arms also have inbuilt load cells to determine the forces and torques at the joints Company (n.d.). On a multicopter platform, however, this force might destabilize the platform.

We can use the methods mentioned above on a multirotor platform as well (Yüksel, Secchi, Bülthoff, and Franchi (2019)), but there are other methods of force and torque estimation that are employed on multirotors that do not require additional sensors. In (McKinnon and Schoellig (2016)), external force and torque is estimated using the states such as the position and the attitude of the mulitrotor. In this paper, we will use the force profile to do high level task allocation for the multirotor. Based on the profile of the external force that is acting on the multirotor, it will achieve different tasks.

There is a risk factor in making use of physical interaction with a multirotor. There are multiple propellers that move at high rpm that puts the user under risk. In Abtahi et al. (2017), it is suggested that user will physically interact with a covered platform without hesitation. In this paper we will demonstrate this by handing over a payload to a human that is physically interacting with the multirotor to indicate that the reciever is awaiting delivery. multirotors have been used for payload delivery in many cases (Jackson et al. (2020); Lee and Son (2020)). In (Jackson et al. (2020)), payload delivery to an accuracy of 5m is claimed. We want to deliver a payload to the hands of the human reciever, while this does not need an accurate delivery as the human can move his hand around, the physical contact with the human might destabilize the multirotor. stability of a helicopter when it acquires a payload is explored and a robust PID controller is demonstrated in (Pounds, Bersak, and Dollar (2011)).

In the studies that are surveyed above, do not show a clear waay of classifying different types of forces that are exerted on a multirotor platform. Singh et al. (2018), classifies to some extent but it still requires a setup consisting of special sensors A conscious force exerted by a human on the UAV platform is different from force exerted on it by collision, or force exerted due to continuous wind. Getting the information on the type of force that is being exerted on the UAV platform will help us design the behaviour of the platform when it encounters each of these force signatures. For example, detecting a human interaction may command the UAV to deliver the Payload to the human. Currently, drone delivery is explored, but is limited as it drops the payload near the receiver. Wing (n.d.) There is a gap in the literature where studies agree that a touch based interaction is useful, but the actual implementation of a touch based communication system is lacking.

1.1 Contribution

In this paper, we address the need for a touch based communication system using LSTM. We can find differences in force signature based on the states of the drone and the output given by the drone.

The contribution of this paper is a deep learning method to classify different types of forces without explicitly estimating the force acting on the drone. Deep learning techniques have been applied in other fields to detect anomalies in time series data. We explore LSTMs to be used for this purpose. The force classification is demonstrated in a real life application where the drone is expected to release a payload when it detects a particular type of force signature. We also show that the technique can be quickly implemented on any quadcopter platform regardless of the difference in the dynamics of the platforms. This force classification method can offer a natural way of providing this feedback.

2 Structure of the paper

The paper is structured as follows. The model is presented in section (3). The DNN model used for detecting the human interaction is presented in section (4.3). An application for the presented method in the form of payload delivery is presented in section (4.4).

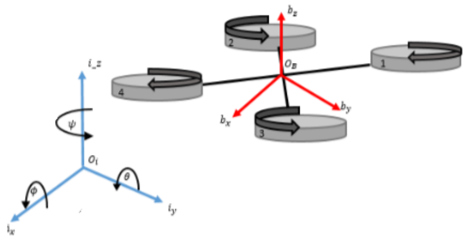

Fig. 1: Coordinate system

3 Modelling, Control and identification

In this section, we will present a non linear model for the quadrotor. We consider the actuator dynamics and the time delays. Then we present a linear model that can be used for system identification. This method is used for any multi rotor platform, but since we are using a Qdrone, we have derived it for a quad copter.

3.1 Reference frames and coordinate system

we define a right handed, Earth fixed inertial coordinate frame $\mathcal{F}_i$ with the z direction pointing upwards(opposite of the gravity). We define the position vector with reference to this frame. ${}^I\mathbf{P} = [x\ y\ z]^T$ is in the inertial frame of reference. The rotations around the inertial axes is given by Euler angles $\eta = [\phi\ \theta\ \psi]$ which represent roll, pitch and yaw. We have the body-fixed reference frame $\mathcal{F}_B$, where B_z is pointing in the same direction as the thrust force created by the propellers. We also define a horizon frame $\mathcal{F}_H$. This is an earth fixed reference frame which is always yaw aligned with $\mathcal{F}_B$. This is convenient because a command to pitch the multi rotor will move it in the x direction of the horizon frame regardless of the multi rotor yaw.

3.2 Non-linear Multirotor model

The thrust force by a propeller is described by:

$$f_i = k_f \Omega_i^2 \tag{1}$$

The pitching and rolling torque offered by the propellers are given by:

$$\tau_{\phi,\theta_i} = k_f l_{\phi,\theta} \Omega_i^2 \tag{2}$$

The yawing torque produced by the propellers is given by:

$$\tau_\psi = (-1)^{i+1} k_\tau \Omega_i^2 \tag{3}$$

So we can relate the forces and the torques with the angular speed of each propeller:

$$\begin{bmatrix} f_T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} k_f & k_f & k_f & k_f \\ k_f l_\phi & -k_f l_\phi & -k_f l_\phi & k_f l_\phi \\ k_f l_\theta & k_f l_\theta & -k_f l_\theta & -k_f l_\theta \\ k_\tau & -k_\tau & k_\tau & -k_\tau \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{4}$$

We assume that the platform is symmetric about all axes and that the center of gravity of the platform coincides with the origin of $\mathcal{F}_B$. with these assumptions, the product of inertia of the platform is zero. Therefore, the inertia matrix is $\mathbf{J} = diag(J_x, J_y, J_z)$. The newton Euler equations for the platform is given by:

$$\begin{bmatrix} m\mathbf{I}_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \boldsymbol{J} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{V}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} {}^B\boldsymbol{F} \\ {}^B\boldsymbol{\tau} \end{bmatrix} \tag{5}$$

$${}^B\boldsymbol{F} = f_T \boldsymbol{b_z} - {}^B_I R m g \boldsymbol{i_z} - \boldsymbol{\alpha} \tag{6}$$

$${}^B\boldsymbol{\tau} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} - \boldsymbol{\lambda} \tag{7}$$

α, and λ are arbitrary constants that depend on the translational and rotational drag on the platform. From eqn (5),(6) and (7), if we neglect the cross coupling, we can simplify the model.

$$\begin{aligned} {}^I\ddot{p}_x &= \frac{1}{m}\left((c_\psi s_\theta c_\phi + s_\psi s_\phi) f_T - \alpha_x({}^I\dot{p}_x, \eta, \Omega)\right), \\ {}^I\ddot{p}_y &= \frac{1}{m}\left((s_\psi s_\theta c_\phi - c_\psi s_\phi) f_T - \alpha_y({}^I\dot{p}_y, \eta, \Omega)\right), \\ {}^I\ddot{p}_z &= \frac{1}{m}\left(c_\phi c_\theta f_T - g - \alpha_z({}^I\dot{p}_z, \eta, \Omega)\right), \\ \ddot{\phi} &= \frac{1}{J_x}\left(\tau_\phi - \lambda_\phi(\omega, \Omega)\right), \\ \ddot{\theta} &= \frac{1}{J_y}\left(\tau_\theta - \lambda_\theta(\omega, \Omega)\right), \\ \ddot{\psi} &= \frac{1}{J_z}\left(\tau_\psi - \lambda_\psi(\omega, \Omega)\right). \end{aligned} \tag{8}$$

3.3 Actuator Dynamics

We use a brushless DC Motor(BLDC) as the actuator. The electronic speed controller(esc) recieves a command u_i and produces a thrust of f_i. the relation between the command and the actual thrust is approximated by a first order plus time delay Cheron, Dennis, Semerjyan, and Chen (2010).

$$G_{prop}(s) = \frac{K_{prop}e^{-\tau_{act}s}}{T_{prop}s+1} \tag{9}$$

3.4 Attitude and altitude dynamics

The Altitude and the attitude of the quad copter can be modeled as first order plus integrator system, So it is given by the following transfer function:

$$G_{att,alt} = \frac{K_{eq}}{s(T_1 s+1)} \tag{10}$$

from fig 2. We can get the inner loop dynamics of the multi rotor by cascading (10) with (9):

$$G_{inner} = \frac{k_{eq}e^{-\tau_{in}s}}{s(T_1 s+1)(T_{prop}s+1)} \tag{11}$$

3.5 Side motion dynamics

A quad copter is an under actuated system, we we cannot directly command it to to move laterally. It moves laterally by changing its attitude. The model for this dynamics is given by:

$$G_{outer}(s) = \frac{X(s)}{\phi(s)} = \frac{K_{eq}e^{-\tau_{out}s}}{s(T_2 s+1)} \tag{12}$$

using eqn 11, we can find the side motion dynamics:

$$G_{side}(s) = \frac{K_{eq}e^{-(\tau_{in}+\tau_{out})s}}{s^2(T_{prop}s+1)(T_1 s+1)(T_2 s+1)} \tag{13}$$

3.6 yaw dynamics

The yaw dynamics is different from the inner loop dynamics of the system. it is modeled as a second order system given by:

$$G_{yaw} = \frac{K_{eq}e^{-\tau s}}{s(T_{prop}s+1)} \tag{14}$$

3.7 System Identification

System identification is done using DNN-MRFT described in (Ayyad, Chehadeh, Awad, and Zweiri (2020)). DNN-MRFT identifies the system parameters in real time. This method was also shown to be applicable in many different platforms. Exciting the system in particular periodic motion, we can reveal the unknown system dynamics. The output from the system is fed to a DNN classifier which classifies the parameters and provides a near optimal tuning. The Period motion is motion is excited by MRFT given by (Haekal and Boiko (2014)):

$$u_M(t) = \begin{cases} h & : e(t) \geq b_1 \lor (e(t) > -b_2 \land u_M(t-) = \ \ h) \\ -h & : e(t) \leq -b_2 \lor (e(t) < b_1 \land u_M(t-) = -h) \end{cases} \tag{15}$$

Where $b_1 = -\beta e_{min}$ and $b_2 = \beta e_{max}$. e_{max} and e_{min} are the last maximum and minimum values of the error signal after it crosses the zero level. Initially, they are both set to be 0. β is a tune able parameter which decides the phase of the excited oscillations.

Using the describing function method, the MRFT achieves oscillations at a specified constant phase by satisfying the harmonic balance equation:

$$N_d(a_0)G(j\Omega_0) \tag{16}$$

Where N_d is the describing function and G is the process that is under test. a_0 and Ω_0 are the amplitude and the frequency of the steady state oscillations. The Describing function is given by:

$$N_d(a_0) = \frac{4h}{\pi a_0}(\sqrt{1-\beta^2} - j\beta) \tag{17}$$

The parameters of the inner loop is discretized in a space where the parameters are likely to lie. This space is determined by the common range of these parameters on similar quad copter models. The DNN acts as a classification for the discretised system parameters (Ayyad et al. (2020)).

4 hdi detection using lstms

4.1 Sensors

We use the Quanser Qdrone(fig 5) as our vehicle of choice. people would be more likely to interact with a drone if it is perceived as safe to touch. (Abtahi et al. (2017); Cauchard, E, Zhai, and Landay (2015)) The Qdrone already has protected

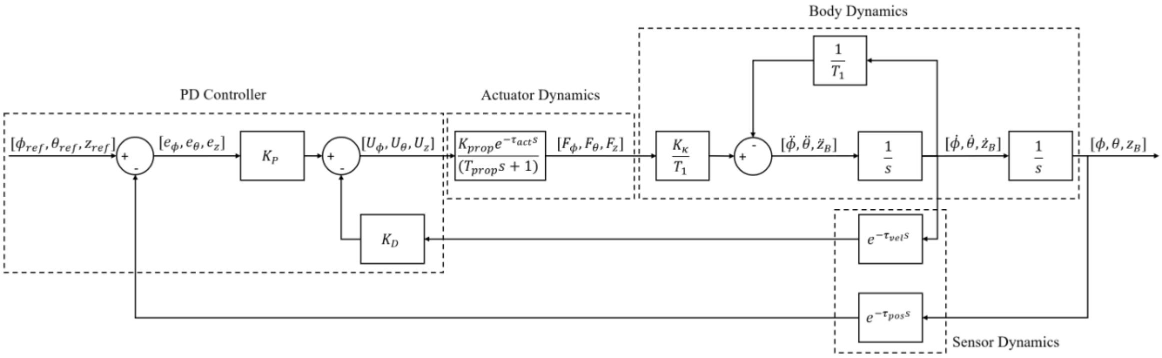
Body Dynamics
PD Controller
Actuator Dynamics
$\frac{1}{T_1}$
$[\phi_{ref}, \theta_{ref}, z_{ref}]$
+
-
$[e_\phi, e_\theta, e_z]$
K_P
$[U_\phi, U_\theta, U_z]$
$\frac{K_{prop}e^{-\tau_{act}s}}{(T_{prop}s+1)}$
$[F_\phi, F_\theta, F_z]$
$\frac{K_\kappa}{T_1}$
$[\ddot{\phi}, \ddot{\theta}, \ddot{z}_B]$
$\frac{1}{s}$
$[\dot{\phi}, \dot{\theta}, \dot{z}_B]$
$[\phi, \theta, z_B]$
K_D
$e^{-\tau_{vel}s}$
$e^{-\tau_{pos}s}$
Sensor Dynamics

Fig. 2: Generalized model used for attitude and altitude (ϕ, θ, z_B) dynamics with PD feedback controller.

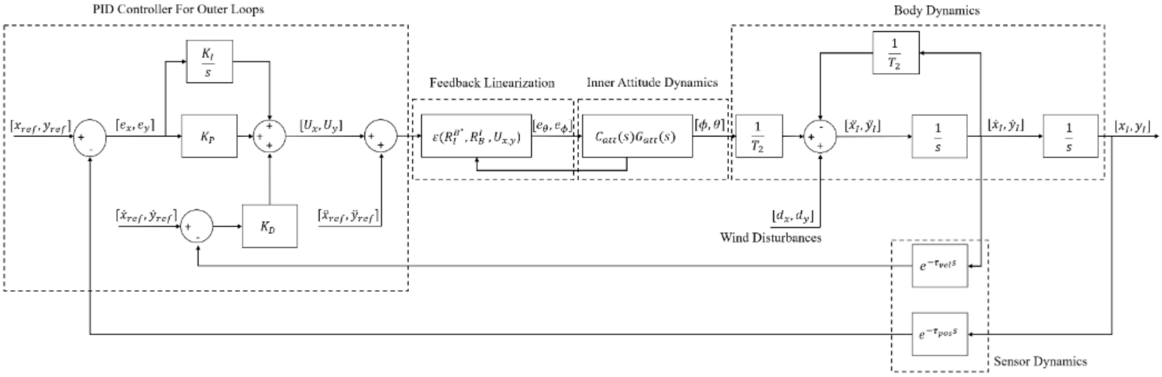

Fig. 3: Generalized model used for horizontal motion (x_I, y_I) dynamics with PID feedback controller.

propellers that make it relatively safe to use. We make it safer by covering the whole drone in a wire mesh.fig **??**

We use a Madgwick filter(Madgwick (2010)) to estimate the attitude of the Qdrone using the BMI160 onboard IMU. An optitrack system is used at 250Hz to find the position and yaw estimates.

4.2 Data gathered

we use the following data for HDI detection:

- e_z
- ϕ
- θ
- ψ
- a_x
- a_y
- a_z
- U_i where $i \in [1:4]$

Where U_i is the motor command given to each of the motors. From these data, we extract the features that are used to train the LSTM model. The features that are used to detect HDI are:

- L2 norm of the acceleration ($\|a\|_2$)
- L2 norm of the orientation($\|\eta\|_2$)
- angle difference between acceleration and orientation($dist$)
- error_yaw (e_ψ)
- clockwise motor input (Ω_{cw})
- counterclockwise motor input (Ω_{ccw})

$$\|a\|_2 = \sqrt{a_{xi}^2 + a_{yi}} \tag{18}$$

$$\|\eta\|_2 = \sqrt{\phi_B^2 + \theta_B} \tag{19}$$

$$dist = \arctan(\tfrac{a_{xi}}{a_{yi}}) - \arctan(\tfrac{\theta_B}{\phi_B}) \tag{20}$$

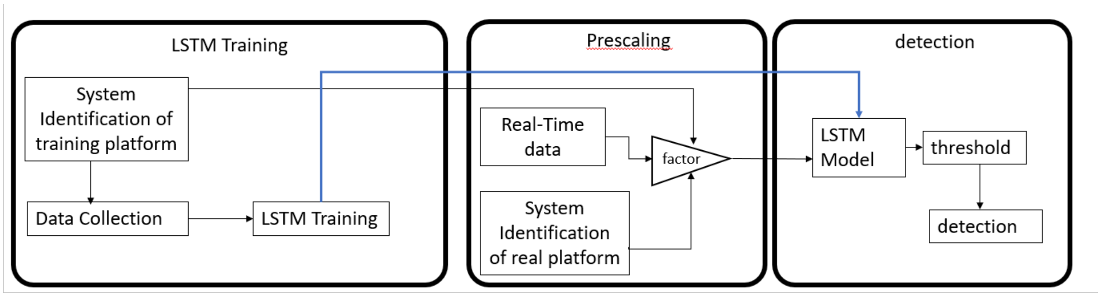

Fig. 4: Proposed system

Fig. 5: Qdrone

Parameter	search space
number of layers	$\{1, 2, 3\}$
neurons per layer	{100, 200, 250}
activation function	ReLu
optimizer	{rmsprop, adam}
features	{2, 3, 4, 6}

Table 1: Search space of the neural network

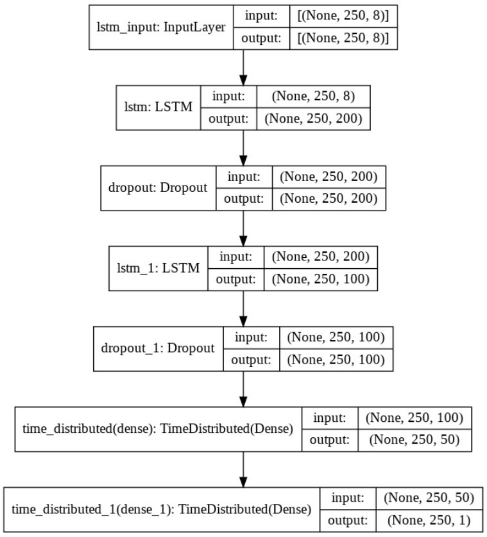

Fig. 6: The model architecture

$$\Omega_{cw} = \frac{U_1 + U_3}{2} \tag{21}$$

$$\Omega_{ccw} = \frac{U_2 + U_4}{2} \tag{22}$$

4.3 LSTM Model

We use an LSTM based machine learning model to detect the human interaction acting on the platform. We use the root mean square propagation algorithm for training the neural network. We determine the best size for the neural network and the set of hyperparameters that will give us the best results, we use an automated hyperparameter space searching. We choose the best performing neural network out of 55 different networks. The search space is demonstrated in (table 1).

After the hyperparameter search, we can choose the best neural network based on the prediction accuracy on the testing dataset. The architecture of the neural network that was chosen is shown in (figure 6).

The LSTM model will give us a detection result for every timestep, to check for sustained human interaction, we employ a threshold that will detect a human interaction if interaction is detected for n consecutive timesteps. We define a positive detection as the detection given after the threshold. This is done to minimize the false positives. Clearly, this will be slightly different from the accuracy given from keras since the threshold will filter out any lone positives or negatives. The LSTM based neural network was able to achieve an accuracy of 96%. In a similar project which explored the classification of different types of touches on a robotic arm, they were able to achieve an accuracy of 93%. (Singh et al. (2018)) We have also shown that we can detect different types of

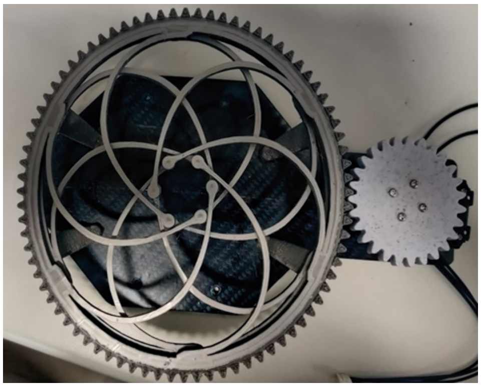

Fig. 7: Iris Gripper

force signatures. We can detect a direct downward pull, and we can detect any twisting of the drone. This is useful for high level task allocation.

4.4 Payload delivery

We demonstrate the touch based interaction by using it to achieve a payload delivery scenario. The objective is to handover a payload to a human receiver based on a command from the receiver. We provide a supplementary video. A slight pull from the receiver is detected by the drone and this tells the Qdrone to release the payload. For this demonstrated, we have chosen a cup to be delivered, but this method can be extended to other payloads. To hold the payload and to release it on command, we design a gripper.

4.4.1 Gripper

We designed the gripper based on the iris gripper (*Grippers* (n.d.)). This gives us the ability to have a compliant mechanism that offers just enough force to hold the cup up.

The gripper is controlled by a servo motor(dynamixel Ax-12a). This allows us to use the serial port of the Qdrone to release the payload. The mass of the gripper with the servo motor is 260 grams. It must be noted that the payload capacity of the the Qdrone platform is 300 grams. Since we have a payload that is close to the max payload, we will need to identify the system after the gripper is added to the platform since it might change the dynamics of the platform. This change in the dynamics between the platform that the model was trained on and the platform where the model is running on may not be effective for detecting human interaction without requiring retraining. To overcome this, we propose a tuning method to give similar response to human interaction as the platform on which it was trained on. Using the method in section 3.7, the change in the model parameters after adding the payload to the platform is given in (table: 2):

4.5 Data normalization

A human interaction that pulls the drone straight down in the z direction will depend mostly on the data in the z direction. The human interaction is encoded into the amplitude, and shape of the error in the z position of the platform. The amplitude and shape of the position error to a human interaction depends on the dynamics of the platform as well as the controller that is used by the platform. In subsection 4.4.1, we have demonstrated how the dynamics of the platform changes when the payload is added to the platform. If we hope to use the neural network on platforms with different dynamics, we must normalize the data in such a way that it retains the amplitude and shape of the position error. We identify the system of the platform that we used to train the LSTM model to obtain a parameter vector $P_T = [K_{eq_T}, K_{P_T}, K_{d_T}, T_T]$. After the payload is added, the dynamics of the system will change and so the response to an external human interaction will also change. We perform system identification on the platform with the payload to obtain a parameter vector $P_P = [K_{eq_P}, K_{P_P}, K_{d_P}, T_P]$. We obtain a normalizing factor G given by:

$$G = \frac{K_{eq_P} \times K_{P_P} \times K_{d_P} \times T_T}{K_{eq_T} \times K_{P_T} \times K_{d_T} \times T_P} \tag{23}$$

We have demonstrated in simulation that we can normalize the position error such that it retains the shape and amplitude of the response without payload by scaling the position error by G in (fig: 8).

This was also demonstrated on the real platform, the position error is shown in (fig: 9). small changes in the shape as compared to the simulation is because in a real experiment, it is difficult to maintain the same force profile on different flights. In simulation, we can define a particular force profile that is exerted on all flights.

	K_{eq}	T_{prop}	T_1	τ
Without Payload	0.1415	0.0224	0.2776	0.0656
With Payload	0.2337	0.0177	0.5794	0.0526

Table 2: DNN-MRFT z channel identification results for the QDrone with and without the payload.

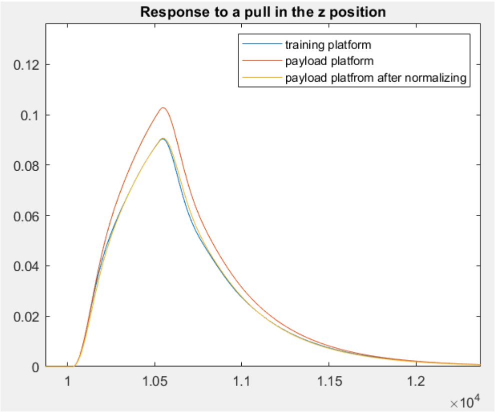

Fig. 8: response to a pull after normalizing with model parameters
flight 1(blue): Platform where the neural network was trained on
flight 2(orange): Platform with the payload
flight 2(yellow): flight 2 data after scaled by G

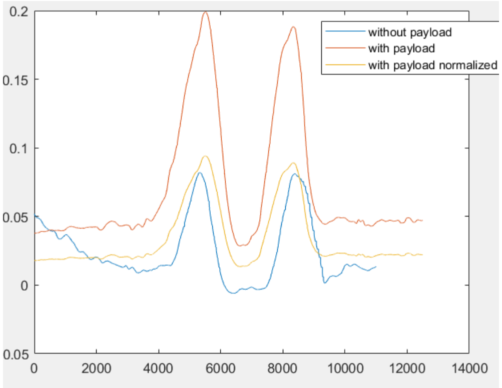

Fig. 9: real response to pull after normalizing

4.5.1 Compliance

We demonstrate this mode of communication with the multi rotor with another application. We show that we can apply compliance to the drone when there is a particular human interaction with the platform. When the Lstm model detects a human interaction, the platform will hold its position for some time. This is done so that interacting with the multi rotor will give the user a more natural experience. The benefit of this is that the Human receiver need not exert a continuous force When retrieving the payload.

4.6 Disturbance rejection

The model should reject any other type of interaction other than the particular force signature that it is looking for. This can be seen in the supplementary video. Wind of upto 3m/s is offered to the drone. the HDI detection is tested in this condition and the model is still able to reject the wind disturbance and detect the particular force signature. A video demonstration of the disturbance rejection is provided in the supplementory video.

5 conclusion

We have demonstrated that the UAV platform is able to detect A specific force signature acting on it and we have demonstrated that a command can be encoded to a particular force signature. This is shown by the payload release capabilities of the platform. Any number of commands may be encoded to any number of different force signatures to control the platform by touch. The demonstration of the payload delivery used here is done so because it is the most natural task that require direct contact with the platform.

Another useful task this method may be applied to is to command the platform to follow-me. This is a feature that is seen in some commercial platforms and since it requires the platform to be in close proximity to the human, this command may be given using the touch mode.

References

Abtahi, P., Zhao, D.Y., E., J.L., Landay, J.A. (2017, September). Drone near me. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *1*(3), 1–8. Retrieved from https://doi.org/10.1145/3130899

10.1145/3130899

Ayyad, A., Chehadeh, M., Awad, M.I., Zweiri, Y. (2020). Real-time system identification using deep learning for linear processes with application to unmanned aerial vehicles. *IEEE Access*, *8*, 122539–122553. Retrieved from https://doi.org/10.1109/access.2020.3006277

10.1109/access.2020.3006277

Cauchard, J.R., E, J.L., Zhai, K.Y., Landay, J.A. (2015). Drone & me. *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing - UbiComp '15.* ACM Press. Retrieved from https://doi.org/10.1145/2750858.2805823 10.1145/2750858.2805823

Cheron, C., Dennis, A., Semerjyan, V., Chen, Y. (2010, July). A multifunctional HIL testbed for multirotor VTOL UAV actuator. *Proceedings of 2010 IEEE/ASME international conference on mechatronic and embedded systems and applications.* IEEE. Retrieved from https://doi.org/10.1109/mesa.2010.5552032 10.1109/mesa.2010.5552032

Company, T.K. (n.d.). *Optoforce.* Retrieved from https://www.knottsco.com/optoforce

Delfa, J.L., Baytas, M.A., Patibanda, R., Ngari, H., Khot, R.A., Mueller, F.'. (2020, April). Drone chi: Somaesthetic human-drone interaction. *Proceedings of the 2020 CHI conference on human factors in computing systems.* ACM. Retrieved from https://doi.org/10.1145/3313831.3376786 10.1145/3313831.3376786

Eguíluz, A.G., Rañó, I., Coleman, S.A., McGinnity, T.M. (2019, January). Reliable robotic handovers through tactile sensing. *Autonomous Robots*, *43*(7), 1623–1637. Retrieved from https://doi.org/10.1007/s10514-018-09823-2

10.1007/s10514-018-09823-2

(n.d.). Retrieved from https://www.flxsys.com/grippers

Haekal, M., & Boiko, I. (2014, June). MRFT based identification of process dynamics. *2014 13th international workshop on variable structure systems (VSS).* IEEE. Retrieved from https://doi.org/10.1109/vss.2014.6881128 10.1109/vss.2014.6881128

Jackson, S.W., Riccoboni, N.A., Rahim, A.H.A., Tobin, R.V., Bluman, J.E., Kopeikin, A.N., ... Prosser, E.M. (2020). Autonomous airborne multi-rotor uas delivery system. *2020 international conference on unmanned aircraft systems (icuas)* (p. 702-708). 10.1109/ICUAS48674.2020.9214011

Lee, S., & Son, H. (2020). Antisway control of a multirotor with cable-suspended payload. *IEEE Transactions on Control Systems Technology*, 1-9.

10.1109/TCST.2020.3035004

Madgwick, S. (2010). An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, *25*, 113–118.

McKinnon, C.D., & Schoellig, A.P. (2016, October). Unscented external force and torque estimation for quadrotors. *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS).* IEEE. Retrieved from https://doi.org/10.1109/iros.2016.7759831 10.1109/iros.2016.7759831

Pounds, P.E.I., Bersak, D.R., Dollar, A.M. (2011). Grasping from the air: Hovering capture and load stability. *2011 ieee international conference on robotics and automation* (p. 2491-2498). 10.1109/ICRA.2011.5980314

Santis, A.D., Siciliano, B., Luca, A.D., Bicchi, A. (2008, March). An atlas of physical human–robot interaction. *Mechanism and Machine Theory*, *43*(3), 253–270. Retrieved from https://doi.org/10.1016/j.mechmachtheory.2007.03.003

10.1016/j.mechmachtheory.2007.03.003

Singh, H., Controzzi, M., Cipriani, C., Caterina, G.D., Petropoulakis, L., Soraghan, J. (2018, September). Online prediction of robot to human handover events using vibrations. *2018 26th european signal processing conference (EUSIPCO).* IEEE. Retrieved from https://doi.org/10.23919/eusipco.2018.8553474 10.23919/eusipco.2018.8553474

Wing, P. (n.d.). *Google drones will deliver chipotle burritos at virginia tech.* Retrieved from https://money.cnn.com/2016/09/08/technology/google-drone-chipotle-burrito/index.html

Yüksel, B., Secchi, C., Bülthoff, H.H., Franchi, A. (2019, March). Aerial physical interaction via IDA-PBC. *The International Journal of Robotics Research*, *38*(4), 403–421. Retrieved from https://doi.org/10.1177/0278364919835605

10.1177/0278364919835605