Guidance, Navigation and Control for Interception of Non-Cooperative UAVs*

Marina Moreira[1,2], Endre Papp[1] and Rodrigo Ventura[2]

***Abstract*— The widespread use of unlicensed UAVs has been posing serious safety and security concerns, e.g., the risk of overflying restricted airspaces, such as airports. Therefore, there is a need for the neutralisation of rogue UAVs that are found violating airspace restrictions. This paper proposes a solution for neutralising non-cooperative UAVs based on high-speed interception using another UAV. The task of intercepting a non-cooperative Target while minimising the time to interception is challenging, as it demands both Target tracking while pushing the actuation envelope of the Interceptor to its limits. We approach this problem using a navigation, guidance and control architecture. For navigation, we use a Kalman Filter, in order to estimate both the Target and Interceptor's relevant states. For guidance, we adopt a Proportional Navigation control method, feeding the desired references into a Thrust Vectoring Controller algorithm. The proposed solution was validated using a real UAV intercepting virtual Targets and quantitatively evaluated using a precise Motion Capture System.**

I. INTRODUCTION

Due to the increasing number of Unmanned Aerial Vehicles (UAV) users, these are gradually becoming an increasing threat to the Public Safety. Learning to operate a small scale UAV only takes a few hours, imposing the risk that novice operators are not sufficiently aware of the airspace regulations. Threats can range from uninformed amateur operators breaching restricted airspaces, to deliberately planned breaches with the intent of causing harm.

Several solutions already exist to cope with this problem: a jammer system [1], an Octocopter carrying a large net [2], a ground-based net launcher [3]. Ultimately, none of these solutions is autonomous, since they still rely on human skill. Autonomous solutions are expected to be more efficient, reproducible, and cost-efficient. Moreover, a better solution should be able to deploy in a wide range of airspaces, therefore, not relying on countermeasures which could affect other lawful air traffic, e.g. using electromagnetic waves.

A. Problem Statement

This paper goal is to develop an UAV Interceptor, I, capable of detecting and intercepting the threatening UAV Target, T, in order to neutralise it. The interception should occur using the onboard Flight Controller, Sensors and Actuators of I in Real-Time. This paper also approaches the problem where the goal is to reduce the distance between I and T, controlling I, minimising the trajectory time and maximising speed, so as to provoke maximum damage to T.

B. State-of-the-Art

Guidance Navigation and **Control** [4], are functions that: define the future desired state; determine the current estimated state from the measured state; and derive control commands to match the current estimated state with the desired state, respectively. For a pursuit mission, the Guidance and Navigation execution is swapped: given the state from the Navigation, the Guidance will produce a reference.

Navigation: For the state estimation of the UAV, a Particle Filter (PF) method is a possibility [5]. Its main benefit is that no linearisation is needed at the expense of increased required computational power. For systems in which computational power is a scarce resource, an efficient solution is the Extended Kalman Filter (EKF). EKF solutions are already being used commercially [6], with new methods arising such as [7]. Nevertheless, the EKF has the constraint that it is not optimal for non-linear systems and there is no stability guarantee. If the initial state estimate is not correct, the filter can diverge, by underestimating the covariance matrices [8].

Having stated this, the extended Kalman filter can give a reasonable performance and is arguably the *de facto* standard in autonomous aeronautical navigation.

Guidance: Air-to-air interception problems were firstly considered when deriving methods for Homing Missile Systems, self-propelled missiles guided while on flight. One widely implemented Guidance System is Proportional Navigation (PN) [9]. The reliable and robust PN methods are still nowadays used for such missile applications [10]. Augmented Proportional Navigation (APN), a derivative of PN, includes the Target acceleration information, in order to allow interception of manoeuvring Targets [11]. However, the acceleration estimation will induce delay, and amplify measurement noise, leading to a drop in the accuracy of the output, potentially performing worse under uncertainty [12].

Other, more recent techniques can be implemented. Using Pontryagin's minimum principle [13] or polynomial restricted trajectories [14], interception trajectories are generated. However, these do not consider unpredictable Targets, as they assume model knowledge. Game Theory has also been considered regarding the Single-Pursuer Single-Evader problem (SPSE) [15], possibly being adapted to UAVs such as performed in [16], in simulation. In the research conducted by [17], it has been shown that the solution of the SPSE problem equals the output of PN under the conditions:

*This work was coordinated and funded by DLR with support from ISR

[1]DLR, German Aerospace Center, Braunschweig, Germany, e-mail: Endre-Aladar.Papp@dlr.de

[2]ISR, Institute for Systems and Robotics, Instituto Superior Técnico, Lisboa, Portugal, e-mail: marina.moreira@ist.utl.pt, rodrigo.ventura@isr.ist.utl.pt

constant flow of Target position; linear dynamics; defined final time; small deviations from zero; unbounded actuation.

The fact that PN is proven robust and still used nowadays, the previous statement gives it a strong edge.

Control: UAV systems are often controlled using PID controllers [18]–[20], where dynamic characteristics of the quadrotor are disregarded, rejecting the disturbances with an error integral term. As the gains are fixed, it lacks robustness against external disturbances and nonlinear dynamics [21].

The Feedback Linearization and Integral-Backstepping techniques have been also implemented. as in [22], [23], taking in factors such as dynamic effects. However, linearized systems are only valid for conditions near hover state.

In Model Predictive Control solutions [24]–[26] both the Guidance and Navigation portion of the problem can be incorporated. MPC algorithms can however be computationally intractable on the embedded hardware typically available for UAVs, and usually use simplifications such as linearisation [24], or offline simulation computations [25], [26].

Finally, in [27]–[29], where Thrust Vectoring Control (TVC) is applied, the non-linearities, dynamic effects, and other disturbances such as wind, can be considered in a computationally lightweight manner. It even improves the image quality for any type of sensing system [21]. In [29], it is proven that TVC results in a faster response than classic PID's, by considering aerodynamics effects directly and direct actuation control.

The TVC is therefore a compromise between fidelity of the UAV model and computation complexity of the algorithm.

C. Contributions

This paper will approach the entire Navigation, Guidance and Control process. Firstly, an EKF will estimate the relevant Interceptor and Target states. Then, the PN algorithm will be adapted, adjusting to the UAV specific dynamics. A TVC will make sure that the UAV follows reference commands, while maintaining independent yaw control, allowing camera tracking. Finally, experimental results are acquired.

D. Outline

The remainder of this paper is structured as follows. In Section II the System Architecture will be presented revealing the inner workings of the System. In Section III the EKF, PN and TVC algorithms will be explored, providing definitions and improvements towards Literature. The experimental results obtained using the Motion Capture System will be exposed and analysed in Section IV. Finally, conclusions will be drawn in V.

II. SYSTEM ARCHITECTURE

The general System Architecture can be seen in Figure 1.

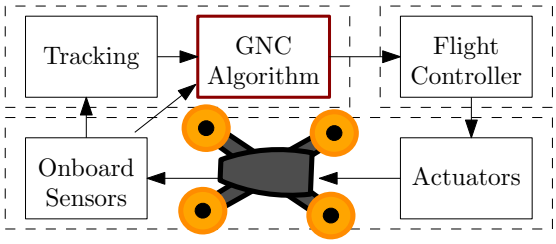

Fig. 1: General System Architecture

The paper algorithms are in the "GNC Algorithm" block. The "Tracking" block obtains an estimation of the relative Target's position, not in this scope. The "Flight Controller" adopted is the *INAV*, which converts the GNC commands into motor input, available open-source [30]. In Figure 2 the GNC Architecture is exposed, where each block element runs an independent algorithm explored further in detail.

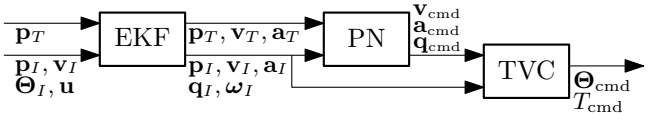

Fig. 2: NGC Sub-System Architecture

The position and velocity of the Interceptor and Target w.r.t. the inertial frame are $\mathbf{p}_I, \mathbf{v}_I$ and $\mathbf{p}_T, \mathbf{v}_T$. The Euler angles ϕ_I, θ_I, ψ_I w.r.t the inertial frame are bundled into $\boldsymbol{\Theta}_I$, with $\mathbf{q}_I$ the same in quaternion units. Finally, $\boldsymbol{\omega}_I$ denotes the Interceptor's angular velocity w.r.t. the body-fixed frame

III. APPROACH

A. Navigation - Extended Kalman Filter

The State model of the EKF is based on the Newton-Euler equations given by [31], modified to use the Pulse Width Modulation (PWM) instead of rotor angular velocity, since PWM is available. The State model is

$$\dot{\mathbf{x}} = \frac{d}{dt}\begin{bmatrix}\mathbf{p}_I\\ \mathbf{v}_I\\ \mathbf{q}_I\\ \boldsymbol{\omega}_I\\ \mathbf{p}_T\\ \mathbf{v}_T\\ \mathbf{a}_T\end{bmatrix} = \begin{bmatrix}\mathbf{v}_I\\ \mathbf{q}_I \otimes T/m \otimes \mathbf{q}_I^* + \overline{g}\\ (1/2)(\mathbf{q}_I \otimes \mathbf{w}_I)\\ \mathbf{J}^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega}_I \times \mathbf{J}\boldsymbol{\omega}_I)\\ \mathbf{v}_T\\ \mathbf{a}_T\\ 0\end{bmatrix}, \tag{1}$$

where T and $\boldsymbol{\tau}$ define the thrust and torque generated by the motors applied in the body-fixed frame, m and J the Interceptor's mass and inertia matrix, and $\overline{g}$ the gravity vector. Using [32] and parametric fitting from data of the Interceptor UAV, we have $T_i = C_T \mathrm{u}_i$ and $\tau_{z_i} = C_M \mathrm{u}_i$, with C_T and C_M the adapted thrust and momentum constants, where input $\mathbf{u}$ is a linear function of the PWM value. The thrust and torque are defined as

$$\begin{bmatrix}T\\ \tau_x\\ \tau_y\\ \tau_z\end{bmatrix} = \begin{bmatrix}C_T & C_T & C_T & C_T\\ -l_y C_T & -l_y C_T & l_y C_T & l_y C_T\\ l_x C_T & -l_x C_T & l_x C_T & -l_x C_T\\ C_M & -C_M & -C_M & C_M\end{bmatrix}\begin{bmatrix}\mathrm{u}_1\\ \mathrm{u}_2\\ \mathrm{u}_3\\ \mathrm{u}_4\end{bmatrix}, \tag{2}$$

where l_x, l_y are the motor's distance to the rotation center.

In the Measurement model of the EKF, the measurements $\hat{\boldsymbol{\Theta}}_I, \hat{\mathbf{p}}_\mathbf{I}, \hat{\mathbf{v}}_\mathbf{I}, \hat{\boldsymbol{\omega}}_\boldsymbol{I}$ are given by the onboard "Flight Controller"

block, and the measurement $\hat{\mathbf{p}}_\mathbf{r}$ is given by the "Tracking" block, which relate to the considered states as

$$\mathbf{z} = \begin{bmatrix} \hat{\phi}_I \\ \hat{\theta}_I \\ \hat{\psi}_I \\ \hat{\mathbf{p}}_I \\ \hat{\mathbf{v}}_I \\ \hat{\boldsymbol{\omega}}_I \\ \hat{\mathbf{p}}_r \end{bmatrix} = h(\mathbf{x}) = \begin{bmatrix} \tan^{-1}\left(\dfrac{2q_w q_x + q_y q_z}{1-2(q_x^2+q_z^2)}\right) \\ \sin^{-1}(2(q_w q_y - q_z q_x)) \\ \tan^{-1}\left(\dfrac{2q_w q_z + q_1 q_y}{1-2(q_y^2+q_z^2)}\right) \\ \mathbf{p}_I \\ \mathbf{v}_I \\ \boldsymbol{\omega}_I \\ \mathbf{q} \otimes (\mathbf{p}_T - \mathbf{p}_I) \otimes \mathbf{q}^* \end{bmatrix} \tag{3}$$

B. Guidance - Proportional Navigation

Proportional Navigation is a Guidance law in which the Interceptor is commanded to turn at a rate proportional to the angular velocity of the line-of-sight line [9], Figure 3.

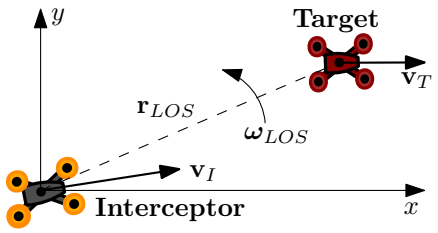

Fig. 3: Proportional Navigation schematics

The desired Interceptor's acceleration is defined such that it is proportional and perpendicular to its current velocity as

$$\mathbf{a}_{PN} = N_{PN}\mathbf{v}_I \times \boldsymbol{\omega}_{LOS}, \tag{4}$$

where N_{PN} is the PN constant, $\mathbf{v}_I$ is the Interceptor's velocity, and $\boldsymbol{\omega}_{LOS}$ is the line-of-sight rate defined as

$$\boldsymbol{\omega}_{LOS} = \frac{\mathbf{r}_{LOS} \times \mathbf{v}_r}{\mathbf{r}_{LOS} \cdot \mathbf{r}_{LOS}}, \tag{5}$$

where $\mathbf{v}_r = \mathbf{v}_T - \mathbf{v}_I$ is the relative velocity of the Target w.r.t the Interceptor in the Inertial reference frame, and $\mathbf{r}_{LOS} = \mathbf{p}_T - \mathbf{p}_I$ is the relative position vector. The vector cross product is defined by $\times$.

In Missile Guidance Systems, defining the desired velocity is not an issue due to its dynamics where the acceleration commands change the velocity direction by default. However in an UAV System, the velocity direction needs to be defined. Therefore, to maintain consistency between a missile and the UAV model we define the desired Interceptor's velocity as

$$\mathbf{v}_{PN_k} = V_{PN}\frac{\mathbf{v}_{PN_{k-1}} + \Delta t \mathbf{a}_{PN_{k-1}}}{\left\|\mathbf{v}_{PN_{k-1}} + \Delta t \mathbf{a}_{PN_{k-1}}\right\|}, \tag{6}$$

where the initial velocity is defined using

$$V_{PN} = \left\| b\frac{\mathbf{r}_{LOS_0}}{\|\mathbf{r}_{LOS_0}\|} + \mathbf{v}_{T_0} \right\| \Rightarrow \mathbf{v}_{PN_0} = b\frac{\mathbf{r}_{LOS_0}}{\|\mathbf{r}_{LOS_0}\|} + \mathbf{v}_{T_0}, \tag{7}$$

where V_{PN} is fixed, and should be higher than $\|\mathbf{v}_T\|$. The b constant can be easily calculated, using the solution for a second degree equation.

C. Control - Thrust Vectoring Controller

The Thrust Vectoring Controller (TVC) law implemented is derived from [29], where we ignore the wind velocity. For the Drag terms, a Lumped Drag coefficient including Blade Flapping, and Induced Drag is used, as it is relevant for all speeds. Contrarily to [29] Parasitic Drag term cannot be neglected because it is relevant for velocities above $10[m/s]$ [33], above the target of the overall project. The TVC Equation is thus

$$\mathbf{F_T} = \mathbf{k}_p \circ \mathbf{e}_p + \mathbf{k}_{pi} \circ \int_0^{\Delta t} \mathbf{e}_p \mathrm{d}t + \mathbf{k}_v \circ \mathbf{e}_v + \mathbf{k}_a \circ \mathbf{e}_a + m\overline{g} + m\mathbf{a}_{PN} + T\mathbf{A}_{bla}\mathbf{v} + K_{par}\|\mathbf{v}_I\|\mathbf{v}_I, \tag{8}$$

where $\circ$ represents the Hadamard product, necessary due to the difference in dynamics between the axis, demanding different gains. The constants $\mathbf{k}_p$, $\mathbf{k}_{pi}$, $\mathbf{k}_v$, and $\mathbf{k}_a$ represent the gains of the position, position integral, velocity and acceleration error; $\mathbf{e}_p$, $\mathbf{e}_v$, and $\mathbf{e}_a$ represent the respective errors; $\mathbf{A}_{bla}$ is the blade flapping coefficient matrix; and $\mathbf{K}_{par}$ is the parasitic drag coefficient. All the variables are defined w.r.t. the inertial reference frame.

The desired set of body axes, can be defined using the Thrust vector and the yaw angle, ψ_I, as shown in [29]. The result can be displayed in a rotation matrix in the form

$$\mathbf{R} = \begin{bmatrix} \mathbf{x}_{cmd}^B & \mathbf{y}_{cmd}^B & \mathbf{z}_{cmd}^B \end{bmatrix}. \tag{9}$$

Considering the matrix rotation $\mathbf{R} = \mathbf{R_z}(\psi)\mathbf{R_y}(\theta)\mathbf{R_x}(\phi)$, one can obtain the desired Euler angles. The Thrust will be defined as to fulfil the z-component of the TVC under inclination, which differs from [29], showing better results. The desired outputs are defined in Equation (10).

$$\begin{bmatrix} \phi_{cmd} \\ \theta_{cmd} \\ T_{cmd} \end{bmatrix} = \begin{bmatrix} \tan^{-1}(\mathbf{R}_{32}/\mathbf{R}_{33}) \\ \tan^{-1}(-\mathbf{R}_{31}/\sqrt{\mathbf{R}_{32}+\mathbf{R}_{33}}) \\ F_{T_z}/(\cos\phi_I \cos\theta_I) \end{bmatrix} \tag{10}$$

Because the UAV has 4DOF, the desirable yaw angle, ψ_{cmd}, is independent and controlled via a PID controller. During interception stage there is no position control, thus $\mathbf{e}_p = 0$.

IV. EXPERIMENTAL RESULTS

To validate the proposed solution, the interception strategy proposed has been tested using virtual Targets in the Intelligent Robots and System Lab, at Instituto Superior Técnico, Lisboa [3]. Due to the Lab's reduced dimensions, high speeds manoeuvres were unfortunately not feasible.

A. Testbed

The System is based on a *Robocat 270mm* Racing Quadcopter UAV frame; a *Matek F722-STD* performing angle and thrust Control; a *Raspberry Pi 3 B+* running *ROS*, executing the interfaces and algorithms. Details can be seen in Figure 4, where the interfaces and algorithms were thoroughly profiled.

[3] www.isr.tecnico.ulisboa.pt

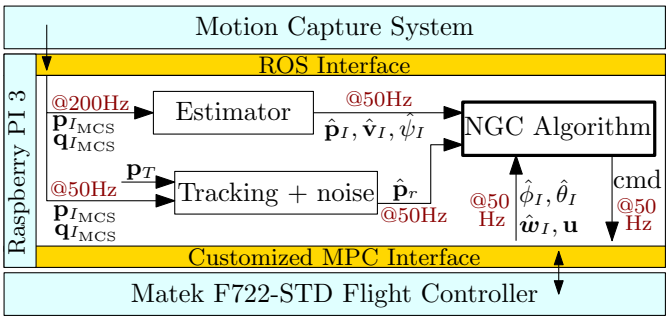

Fig. 4: Testbed schematics

B. Trajectories

The interception starts from the hover state and is considered to end, once the Interceptor is at a radius of $\|\mathbf{p}_r\| < 30[cm]$ of the Target and $\|\mathbf{p}_r\|$ has an inflexion point. The results will be evaluated for interception accuracy, velocity and robustness, through different trajectories.

1) **Line Trajectory:** velocity 1 $[m/s]$. PN should predict the path performing small adjustments.
2) **Circle Trajectory:** velocity 0.5 $[m/s]$, radius 0.5 $[m]$. Defies the PN, as it does not consider acceleration.
3) **Random Trajectory:** random direction of velocity at velocity 0.4 $[m/s]$. Evaluates response to uncertainty.

Multiple Sets of several trials from different initial positions were recorded.

C. Navigation - Extended Kalman Filter

For Navigation, we present results for the Circle Trajectory, as it is the one that poses more challenges to the model assumed in the EKF, which is constant acceleration.

The position results are depicted in Figure 5, where we show that the position is correctly estimated. The Input noise is attenuated, with the estimated error mostly smaller than the noisy input error.

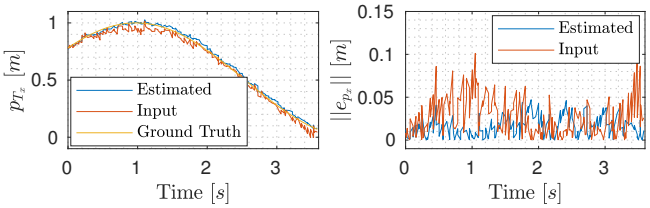

Fig. 5: Target Position and Position Error

In Figure 6, we show that the acceleration and velocity have increasing delays as a compromise between the model and reality because the trajectory deviates from the EKF assumed model.

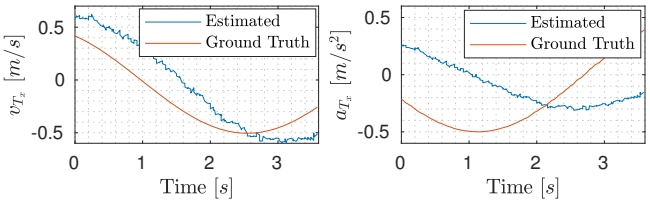

Fig. 6: Target Velocity and Acceleration

D. Guidance - Proportional Navigation

The PN results can be viewed in detail in Figures 8 to 22, where the typical interception paths can be viewed, alongside distance, velocity, acceleration, and the interception error.

Interception paths: in Figures 8, 13, 18, one can clearly distinguish the case where there is acceleration involved to where there is not. In the Line Trajectory, the Interceptor chooses right along the shortest interception path, and performs adjustments according to disturbances. In the Circle Trajectory, the Target fools the Interceptor, which has to do aggressive curves to intercept, which clearly is not the shortest route, but the one predicted considering the currently available data. The Figures include a $10[cm]$ radius circle around the interception point for readability.

Distance to Target: in Figures 9, 14, 19, the initial behaviour observed is a delay in the beginning, due to the starting hover position. Also, the slope of the distance decreases according to the relative velocity w.r.t the Target. In the Circle Trajectory, due to aggressive Target manoeuvres,

there are periods where the distance actually increases. One can note that for initial position C, there is a trial with a second interception, more successful that the remaining ones.

Velocity: in Figures 10, 15, 20, we show that despite the commanded velocity being $1[m/s]$ for all Trajectories, oscillations occur due to the acceleration effects of the PN algorithm. The biggest disturbances are located closer to the Target, because the PN algorithm output is proportional to ω_{LOS}, which, for the same linear velocity disturbance, increases with distance. Impact velocities are usually higher than the specified $1[m/s]$.

Acceleration: in Figures 11, 16, 21, disregarding the initial acceleration due to the transitory state, we observe the same behaviours as for the velocity. In Line Trajectories, even with a correctly predicted interception path, a small error near Target can produce an acceleration of $4[m/s]$. For the Circle trajectory, it can be as high as $6[m/s]$, which is extremely aggressive for our Lab's reduced dimensions.

Interception Error: in Figures 12, 17, 22, we see the Circle Trajectory error is the highest, followed by the Random and Line trajectories. The results validate the approach because, even due to high Target accelerations, the Interceptor is still capable of obtaining a reasonable interception distance. The algorithm is very precise for trajectories which do not contained acceleration, with maximum error under $10[cm]$. The error n the z-axis is always the smallest, as the Target height does not change.

Although not shown here, it was also verified that the error increases with the increase of noise in Target position.

E. Control - Thrust Vectoring Controller

Our main focus is to question if the changes made to the original TVC [29] deteriorate the overall solution, because position error is not used during the interception stage.

As one can see in Figures 10, 15, 20, the velocity stabilises around the $1[m/s]$ reference, with oscillations already mentioned due to the acceleration commands. Moreover, during the transition to interception, maintaining height is not a problem because the UAV departed from a hover state and maintains pre-existent position integral error. Naturally, gains could be tuned, and a solution is already being developed.

In Figure 7, one can observe the Yaw control for the worst case Circle trajectory. The yaw gains were set very low, and consequently create high delay, because safe landing after interception was a concern. These results are within acceptable Field of View range, and even though the design parameters were hand-tuned.

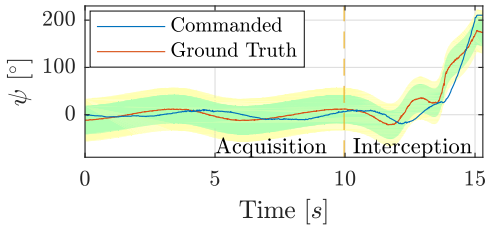

Fig. 7: Yaw Angle control for Circle Trajectory.
Color bands represent a Field of View of $\pm 30^{\circ}$ and $\pm 45^{\circ}$

1) Line Trajectory:

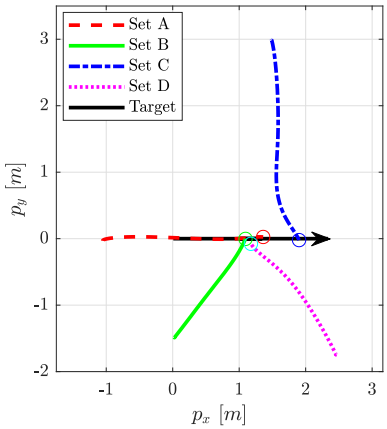

Fig. 8: Line - Interception path

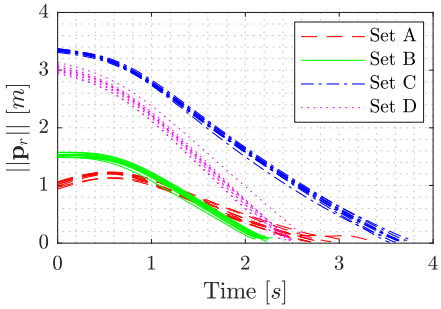

Fig. 9: Line - Distance to Target

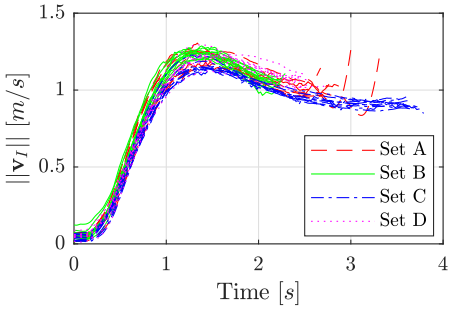

Fig. 10: Line - Interceptor Velocity

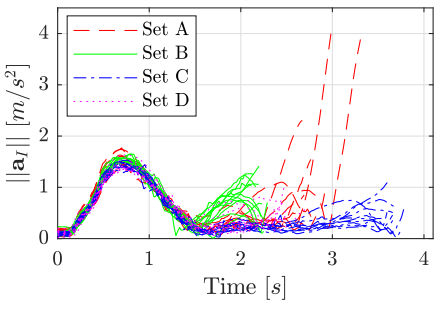

Fig. 11: Line - Interceptor Acceleration

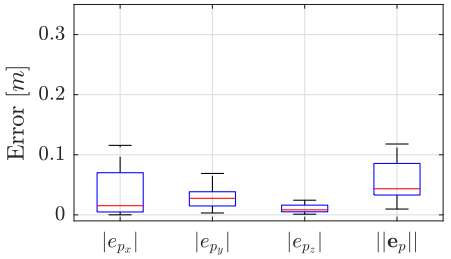

Fig. 12: Line - Interception Error

2) Circle Trajectory:

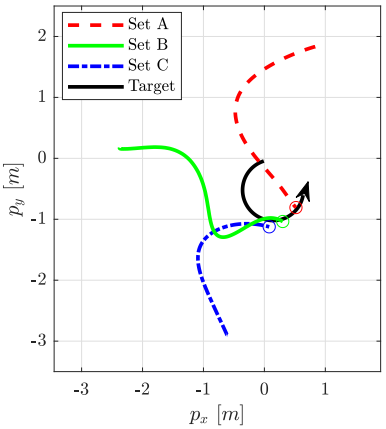

Fig. 13: Circle - Interception path

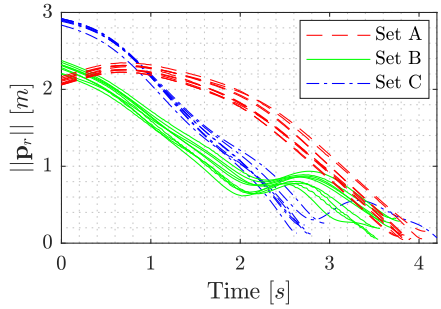

Fig. 14: Circle - Distance to Target

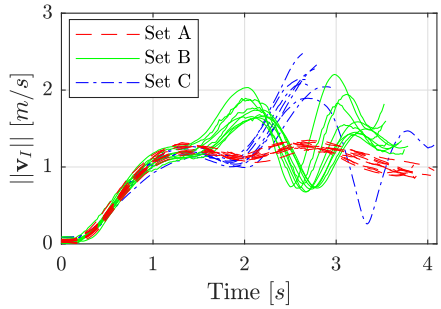

Fig. 15: Circle - Interceptor Velocity

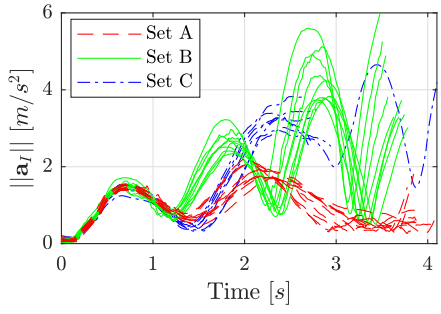

Fig. 16: Circle - Interceptor Acceleration

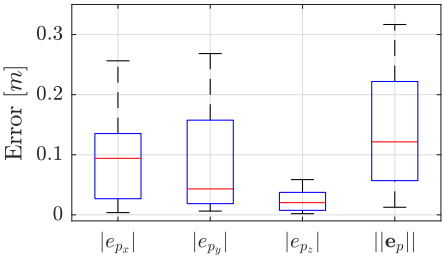

Fig. 17: Circle - Interception Error

3) Random Trajectory:

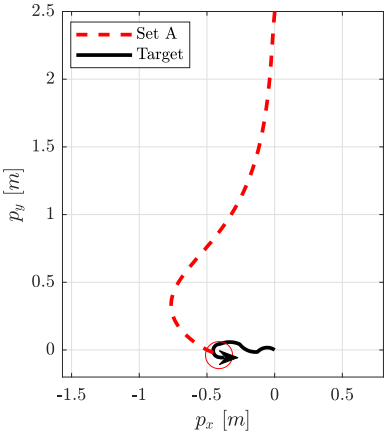

Fig. 18: Random - Interception path

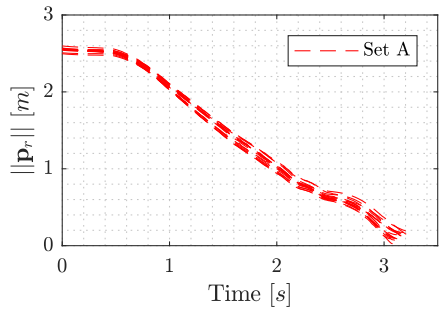

Fig. 19: Random - Distance to Target

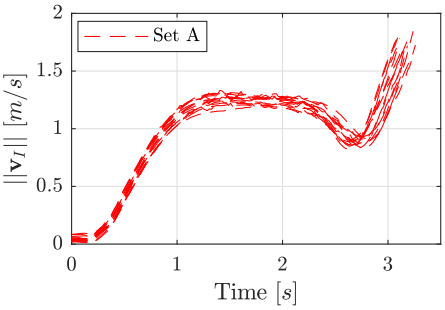

Fig. 20: Random - Interceptor Velocity

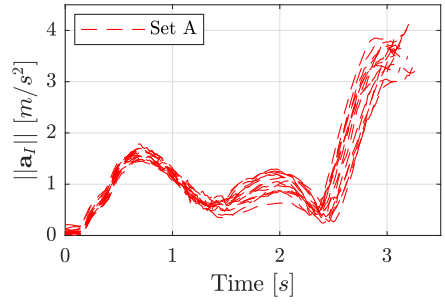

Fig. 21: Random - Interceptor Acceleration

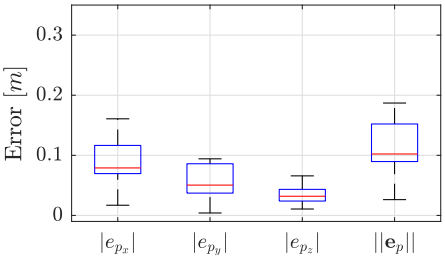

Fig. 22: Random - Interception Error

V. CONCLUSIONS

This paper proposes a solution for the interception of a non-cooperative Target drone by an autonomous UAV, where relative position of the Target UAV are fed to the Interceptor UAV. The solution has shown good results, even when subjected to an aggressive manoeuvring escaping Target.

Firstly, it was shown that a classic PN Missile Guidance System method can be successfully adapted to UAVs using small changes to the Literature solutions, mimicking the Missile Model. The Interceptor UAV can therefore correct its path and predict the Target's future path to minimize interception time, according to the current information.

Then, to estimate the Interceptor and Target's states, an EKF was developed where it was shown that the position, velocity and acceleration states can be successfully estimated from the Target's relative position. However, the results show that the velocity and acceleration estimation highly depends on the chosen Target's Model accuracy.

Finally, it has been shown that in the absence of position commands during the interception stage, the implemented adapted TVC is still capable of following references correctly. Moreover, the TVC improved greatly height control. It was also shown that the yaw control can be considered independent, leaving it as an extra degree of freedom. Eventually, this could be used to position a camera that captures images, autonomously obtaining the Target's position.

A. Future Work

The presented algorithms should be tested further to understand the minimum requirements that a Tracking strategy would impose, by including delay and increase the noise.

Moreover, the solution presented should be tested outdoors, subjected to wind perturbation, using GNSS methods, instead of the Motion Capture System. For validation purposes, an RTK GPS solution is advised.

For the Navigation, an Interactive Multiple Model solution could be developed, to combat the shortcomings of the EKF, where an incorrect model can damage the results.

REFERENCES

[1] IDS. (2016) Black Knight uav detection radar. [Online]. Available: https://www.idscorporation.com/pf/black-knight/

[2] T. Telegraph. (2016) Tokyo police are using drones with nets to catch other drones. [Online]. Available: https://www.telegraph.co.uk/technology/2016/01/21/tokyo-police-are-using-drones-with-nets-to-catch-other-drones

[3] O. W. Engineering. SkyWall100 handheld drone capture system. [Online]. Available: https://openworksengineering.com/skywall

[4] ECSS-E-60A, "Space Engineering – Control Engineering," *European Cooperation for Space Standardization: Space Engineering – Control Engineering*, no. September, 2004.

[5] Y. Dhassi, A. Aarab, and M. Alfidi, *IMM-PF visual tracking based on multi-feature fusion*.

[6] A. D. Team. Extended kalman filter (ekf). [Online]. Available: http://ardupilot.org/copter/docs/common-apm-navigation-extended-kalman-filter-overview.html

[7] G. Mao, S. Drake, and B. D. Anderson, "Design of an extended kalman filter for UAV localization," in *Conference Proceedings of 2007 Information, Decision and Control, IDC*, 2007.

[8] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended kalman filter based slam," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 473–479.

[9] J. H. Blakelock, *Automatic Control of Aircraft and Missiles*, 1991. [Online]. Available: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471506516.html

[10] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd, "Basic principles of homing guidance," *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, vol. 29, no. 1, pp. 25–41, 2010.

[11] V. Garber, "Optimum intercept laws for accelerating targets." *AIAA Journal*, vol. 6, no. 11, pp. 2196–2198, 1968.

[12] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd, "Modern homing missile guidance theory and techniques," *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, vol. 29, no. 1, pp. 42–59, 2010.

[13] M. Hehn and R. D'Andrea, "Real-time trajectory generation for interception maneuvers with quadrocopters," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4979–4984, 2012.

[14] M. Müller, S. Lupashin, and R. D'Andrea, "Quadrocopter Ball Juggling," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5113–5120, 2011.

[15] R. Isaacs, *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.

[16] M. Kothari and J. G. Manathara, "Cooperative Multiple Pursuers against a Single Evader," pp. 551–567, 2017.

[17] Y. C. Ho, A. E. Bryson, and S. Baron, "Differential Games and Optimal Pursuit-Evasion Strategies," *IEEE Transactions on Automatic Control*, vol. 10, no. 4, pp. 385–389, 1965.

[18] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed," *Control Engineering Practice*, 2011.

[19] S. Waslander and C. Wang, "Wind Disturbance Estimation and Rejection for Quadrotor Position Control," in *AIAA Infotech@Aerospace Conference*, 2009.

[20] P. Martin and E. Salaün, *The True Role of Accelerometer Feedback in Quadrotor Control*. [Online]. Available: www.xsens.com

[21] C.-H. Kuo, C.-M. Kuo, A. Leber, and C. Boller, "Vector thrust multirotor copter and its application for building inspection," in *Internat Conf. on Micro Aerial Vehicles (IMAV)*, 2013.

[22] N. Sydney, B. Smyth, and D. A. Paley, "Dynamic control of autonomous Quadrotor flight in an estimated wind field," in *Proceedings of the IEEE Conference on Decision and Control*, 2013.

[23] O. Araar and N. Aouf, "Quadrotor control for trajectory tracking in presence of wind disturbances," in *2014 UKACC International Conference on Control, CONTROL 2014 - Proceedings*, 2014.

[24] M. Bangura and R. Mahony, "Real-time model predictive control for quadrotors," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2014.

[25] M. Mueller and R. D'Andrea, "A model predictive controller for quadrocopter state interception," *Proceedings of the European Control Conference (ECC), 2013*, pp. 1383–1389, 2013. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6669415

[26] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "MPC with Nonlinear H-Infinity Control for Path Tracking of a Quad-Rotor Helicopter," 2008.

[27] S. Omari, M. D. Hua, G. Ducard, and T. Hamel, "Nonlinear control of VTOL UAVs incorporating flapping dynamics," in *IEEE International Conference on Intelligent Robots and Systems*, 2013.

[28] D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, Shanghai, 2011.

[29] J. P. Silva, C. De Wagter, and G. de Croon, "Quadrotor Thrust Vectoring Control with Time and Jerk Optimal Trajectory Planning in Constant Wind Fields," *Unmanned Systems*, 2017.

[30] I. D. Team. Inav - navigation capable flight controller. [Online]. Available: github.com/iNavFlight/inav

[31] J. Cariño Escobar, H. Abaunza González, and P. Castillo Garcia, *Quadrotor Quaternion Control*, jun 2015.

[32] J. G. B. F. Filho, C. E. T. Dórea, W. M. Bessa, and J. L. C. B. Farias, "Modeling, Test Benches and Identification of a Quadcopter," in *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*, 2016, pp. 49–54.

[33] M. Bangura and R. Mahony, "Nonlinear dynamic modeling for high performance control of a quadrotor," in *Australasian Conference on Robotics and Automation, ACRA*, 2012.