Coordinated Control Concept for Recovery of a Fixed-Wing UAV on a Ship using a Net Carried by Multirotor UAVs

Kristian Klausen*, Jostein Borgen Moe*, Jonathan Cornel van den Hoorn†, Alojz Gomola‡,
Thor I. Fossen*, Tor Arne Johansen*
*Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics,
*NTNU, Norwegian University of Science and Technology, Norway
†Delft Centre for Systems and Control, Delft University of Technology, the Netherlands
‡Department of Electrical and Computer Engineering, University of Porto, Portugal
Email: *kristian.klausen@ntnu.no

Abstract

Ship-based Unmanned Aerial Vehicle (UAV) operations represent an important field of research which enables a large variety of mission types. Most of these operations demand a high level of endurance which normally requires the use of a fixed-wing UAV. Traditionally, a net located on the ship deck is used for recovering the fixed-wing UAV. However, there are numerous challenges when attempting autonomous landings in such environments. Waves will induce heave motion, and turbulence near the ship will make approaches challenging. In this paper, we present a concept using multirotor UAVs to move the recovery operation off the ship deck. To recover the fixed-wing UAV, a net is suspended below two coordinated multirotor UAVs which can synchronize the movement with the fixed-wing UAV. The approach trajectory can be optimized with respect to the wind direction, and turbulence caused by the ship can be avoided. In addition, the multirotor UAVs can transport the net at a certain speed along the trajectory of the fixed-wing UAV, thus decreasing the relative velocity between the net and fixed-wing UAV to reduce the forces of impact. This paper proves the proposed concept through a simulation study and a preliminary control system architecture.

I. Introduction

An increased effort has been made to enable autonomous operations with UAVs in marine environments. Here, UAVs are typically used for surveillance, data-acquisition or communication relaying.

Fixed-wing UAVs are often launched with a catapult-like device, powered by either pneumatics, springs or rubber bands. These devices have a reasonable footprint, and are quite popular. For recovery a fixed net can be used. To accommodate this however, a relatively large part of the ship needs to be set aside for UAV operations. This has a large footprint, and for safety reasons a large part of the deck needs to be emptied for both equipment and other personell. Depending on the construction of the net, the UAV has a risk for damage due to the impact with the fixed net. More importantly, waves will induce oscillatory heave motion on the ship, and the nets location and attitude may not always be optimized with respect to wind and turbulence near the ship due to the requirements of other ship operations.

On the topic of recovering fixed-wing UAVs, autonomous landing based on e.g. Global Navigation Satellite Systems (GNSS) [1] or visual servoing [2] have been studied in the literature. For ship-based operations, several methods including nets, hooks and wires have been pursued. Most notably is the SkyHook system developed by Insitu [3]. It consists of two components; a vertical wire attached to a mobile or fixed structure, and a hook on the wing tip of the fixed-wing UAV. When it is passing the wire, the wing gets hooked on tight. This system is commercially available today, with the specially designed ScanEagle UAV. This concept has been expanded by having the wire suspended in air by a heavy-duty multirotor [4]. Further, a concept for landing larger UAVs by using a horizontal wire and a hook, is presented in [5]. In stead of using nets or hooks, the high braking capabilities of a fixed-wing UAV in *deep stall* [6], [7] can be utilized for landing. In [8], the non-linear dynamics of a fixed-wing UAV during deep stall is analyzed and controlled using model predictive control. However, autonomous landing by a deep stall maneuver requires accurate models and high accuracy, and are especially prone to changing wind conditions.

In this paper, we present an approach for landing a small fixed-wing UAV in a net suspended by two powerful ($>$ 10 kg *maximum takeoff weight*) multirotor UAVs. The fixed-wing UAV is equipped with hooks so that after impact with the net, it will be arrested by the net to be transported back to the ship, see Figure 1. Key benefits of such an approach include

This work was partially supported by the Research Council of Norway through its Centers of Excellence funding scheme, grant number 223254 (NTNU Centre for Autonomous Marine Operations and Systems).

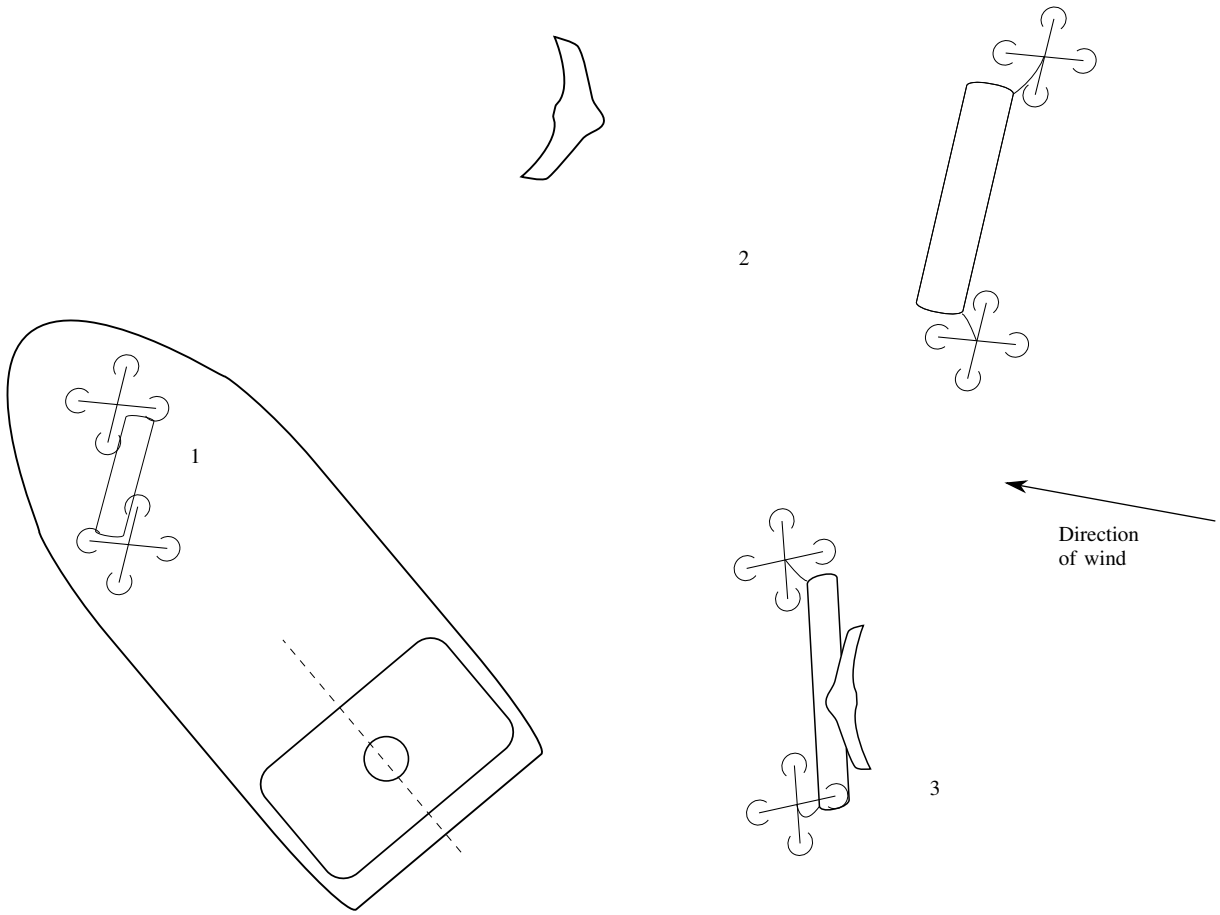

Fig. 1. The figure illustrates the recovery of a fixed-wing UAV. (1), the multirotors take off from the ship. (2), the fixed-wing UAV moves against the wind direction, while the multirotors position the net along its trajectory and accelerate to a prescribed velocity in order to catch the incoming fixed-wing UAV. (3), the multirotors are transporting the fixed-wing UAV back to the ship.

- *Operational flexibility:* When recovering a fixed-wing UAV, it is crucial to travel against the wind to minimize the ground speed, and thus a fixed net needs to be aligned with this path. Even in vessels equipped with Dynamic Positioning (DP) systems, turning the ship can be undesired as it may interfere with operations. The multirotors can however quickly react to changing wind conditions, and align the net against the wind without interfering with other ship operations.
- *Not affected by waves and turbulence:* Since the net is suspended free from the ship, heave motion induced by waves on the ship will not affect the landing. Also, there is no impact from turbulence caused by the ship super-structure.
- *Safety:* By having the net suspended by two multirotor UAVs, the recovery operation can be moved off ship. Thus, no operators or staff risk coming in contact with the incoming UAV.
- *Smaller impact force:* By having the two multirotors move against the wind with the fixed-wing UAV, the relative speed difference between it and the net can be made smaller, thus decreasing the structural load on the fixed-wing body during impact.
- *Smaller footprint:* By moving the landing operation off ship, operations with UAVs can be conducted from smaller ships, not needing a large open deck with a net to support the mission. Launch and recovery of the multirotors are still required.

Recovery with nets suspended by multirotors have been attempted in various settings. Due to the popularity of consumer-type multirotor UAVs, there is an increased interest in the ability to safely remove such vehicles from restricted airspaces. In [9], a multirotor is equipped with a net gun, capable of incapacitating smaller multirotors by shooting a net at them to disable the rotors on the target. A similar experiment was conducted in [10], where the target multirotor stays attached to the larger multirotor after the net is fired, see also [11], [12]. But to the best of the authors knowledge, no attempts to recover fixed-wing UAVs in a net suspended between multirotor UAVs have yet been published.

The contributions of this paper is twofold. First, it presents a controller structure for the net recovery concept, where the main contribution consists of how to combine existing control methodologies to a complete system. Next, we present the results from numerical simulations, which gives insight into the dynamics during the recovery maneuver. In addition, we

give an overview of an implementation of the controller.

A. Organization

This paper is organized as follows. Section II gives an overview of the maneuver and proposed control structure, followed by more details of each part of the controller in Section III. This section also introduces the necessary notation and dynamical models. In Section IV, the multi-body dynamics of the suspended net is discussed to create a simulator with 6 degrees of freedom (DOF) of all involved objects. A model of the impact dynamics is developed in Section IV-B. The results of the simulation is presented in Section V, which also shows the results of the proposed controller. Section VI gives an overview of the systems architecture of the proposed design, including necessary hardware. We also present the results of a Software-In-the-Loop (SIL) simulation, to verify the implementation. This is the same setup to be used in experiments. Section VIII gives a brief summary and concludes the paper.

II. Autonomous Net Recovery Concept

Autonomous recovery of a fixed-wing UAV in a suspended net is a complex task, so the functionality is split into several key components. The overall mission is executed in the following fashion:

- The fixed-wing UAV is instructed to follow a path against the wind, with the minimal airspeed required for safe flying. This is called the *virtual runway*, and the path is transmitted to the multirotor UAVs.
- Both multirotors are equipped with coordinated controllers that keep the inter-formation of the two intact, while lifting the suspended net.
- The current position and the velocity of the fixed-wing UAV is transmitted to a coordination controller in one of the multirotors, which sends desired setpoints to the formation controllers according to the phases of the mission, as to catch the fixed-wing UAV.

Although using two multirotor UAVs instead of one increases the complexity of the system, it has several practical advantages. First, by distributing the load, each multirotor can be physically smaller than a single with the combined lift capacity. Further, the two multirotors can spread the net without a support structure (top beam), giving reduced weight of the net.

Precise navigation is crucial for precision landing of UAVs. In this work, we utilize *Real-Time Kinematic* (RTK) *Global Navigation Satellite System* (GNSS). This is a navigation technique using the carrier wave of the incoming signals from the satellites, and comparing the signals to that received by a base station. By computing the phase shift between the signals at the UAV (rover) and the base, the location can be locked in at centimeter-level accuracy. Such a system was used in [13] for landing a fixed-wing UAV in a stationary net, which also contains more detailed information about RTK GNSS systems.

III. Control design

This section introduces the control design, and gives details about each of the different parts. The overall structure can be seen in Figure 2. As can be seen, there are three distinct control modules, which are detailed next. We also introduce the concept of the *virtual runway*.

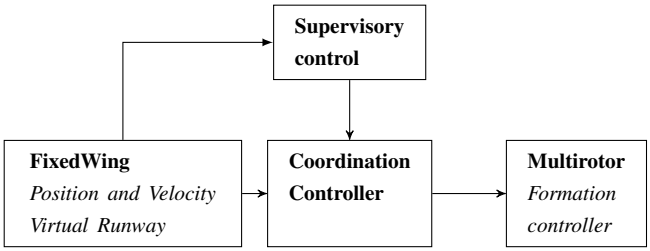

Fig. 2. Information flow in the controller structure. Based on the current position of the fixed-wing UAV, the supervisor starts the net-recovery maneuver. The coordination controller guides the two multirotors along a recovery path to intercept the fixed-wing UAV.

Note that the fixed-wing UAV acts as a reference generator (master) in the proposed control scheme, as it is not affected by the current position of the multirotors. Depending on the type of fixed-wing UAV used, it is preferred to keep a steady flight envelope, rather than correcting minor deviations from the net position. This is much better handled by the agility of the multirotors.

We assume that the fixed-wing UAV moves with a constant course and altitude along a virtual runway, and its position and velocity is communicated to the other vehicles. This is controlled by an on-board autopilot.

In the next sections, let $\mathbf{p}_i^n \in \mathbb{R}^3, i \in \{1, 2\}$ be the position of multirotor i in the inertial frame $\{n\}$. Further, we define the position $\bar{\mathbf{p}}^n$ as the centroid of the two multirotors plus an height offset to compensate for the position of the net. Further, the states of the fixed-wing UAV is denoted with subscript $\cdot_f$.

A. Virtual runway

Figure 3 illustrates the *virtual runway* (VR). The virtual runway defines a path frame $\{p\}$ at constant altitude, which is defined by an origin $\mathbf{p}^n_{p/n}$ and a rotation ψ around the $\{n\}$ z-axis such that $\mathbf{R}^n_p = \mathbf{R}_z(\psi)$. Then a position $\mathbf{p}^n$ can be decomposed in $\{p\}$ by the transformation $\mathbf{p}^p = (\mathbf{R}^n_p)^\top(\mathbf{p}^n - \mathbf{p}^n_{p/n})$. By dividing the path frame into a *cross-track* plane and an *along-track* distance, we can design controllers for each part separately.

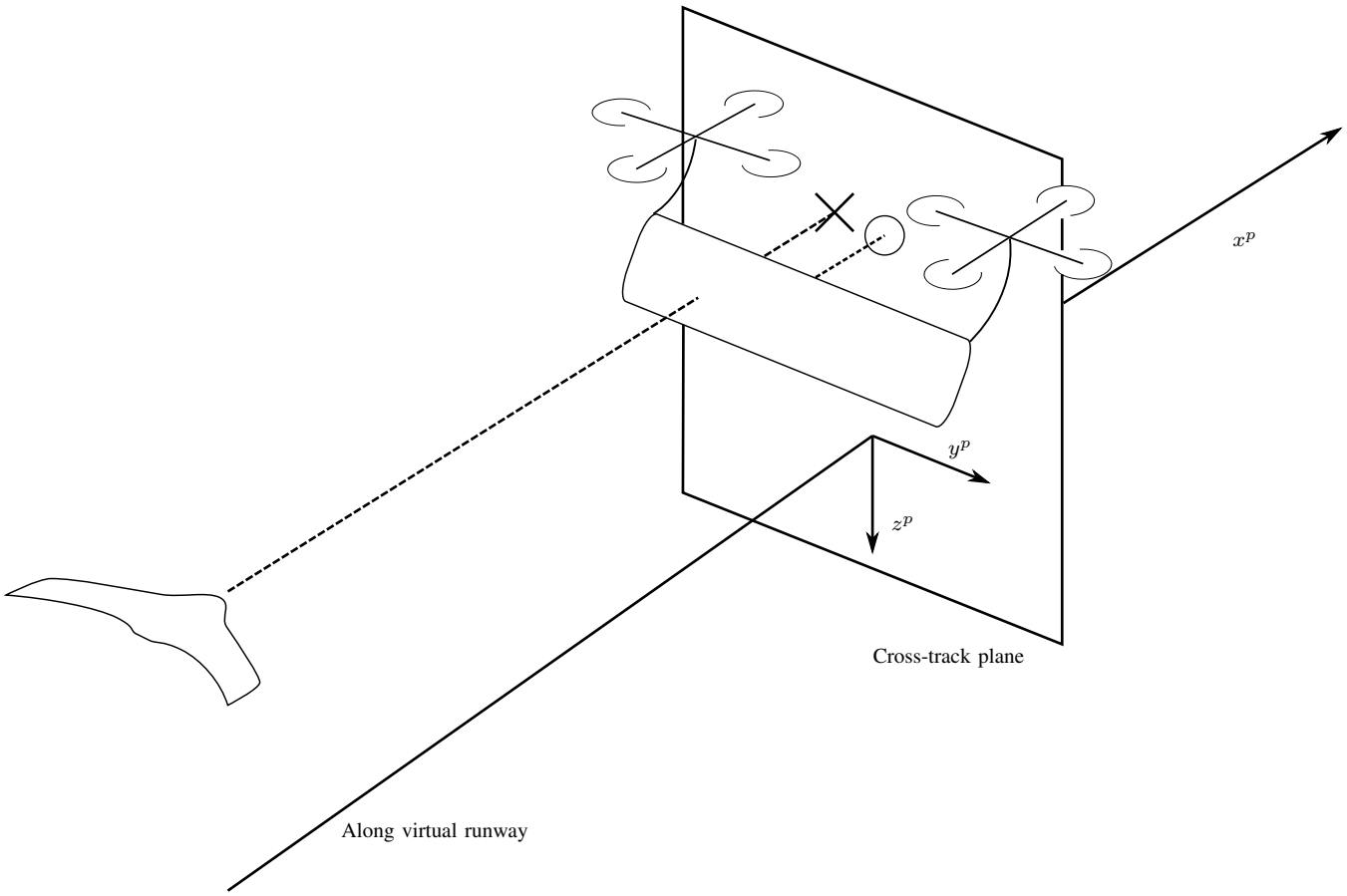

Fig. 3. Illustration of the *virtual runway*. The runway defines a *path frame* $\{p\}$, and can be divided into a cross-track plane ($y^p z^p$) and an along-track distance x^p. The position of the net on the cross-track plane is marked with a circle, while the intersection of the cross-track plane and the path of the fixed-wing UAV is marked with a cross.

B. Supervisor

The supervisor monitors the position and velocity of the fixed-wing UAV relative to the virtual runway in order to switch between the different modes in the maneuver. Each mode enables a certain controller and reference which gives a desired velocity setpoint. Figure 4 gives an overview of the different states of the supervisor. As can be seen in Figure 5, the supervisor also controls when to activate the two parts of the coordination controller.
In addition, the supervisor monitors the maneuver as it is progressing. If, because of wind or other factors, the fixed-wing UAV misses the net, it instructs the vehicles to try the maneuver again. Further, if the projected proximity of the fixed-wing UAV and multirotors are to small, the supervisor can abort the operation. Depending on the situation, an abort can involve the multirotors to climb and reposition for a retry, or releasing the net and abort the mission entirely.

C. Coordination - Cross-track

The position of the net is controlled according to the fixed-wing UAV position in the cross-track plane along the virtual runway. A cross-track frame $\{p_*\}$ is defined as the yz-plane in the path frame $\{p\}$, such that there exist a mapping from a position $\mathbf{p}^p = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^\top$ to $\mathbf{p}^{p_*} = \mathbf{p}^p_{2:3} = \begin{bmatrix} p_y & p_z \end{bmatrix}^\top \in \mathbb{R}^2$. Then a modified pure-pursuit [14] scheme is introduced. Given a desired position $\mathbf{p}^{p_*}_d$ and the position error $\tilde{\mathbf{p}}^{p_*} := \mathbf{p}^{p_*}_d - \mathbf{p}^{p_*}$ the following controller is used

$$\mathbf{v}^{p_*}_d = \mathbf{K}_{p,p}\tilde{\mathbf{p}}^{p_*} + \mathbf{K}_{d,p}\dot{\tilde{\mathbf{p}}}^{p_*} + \mathbf{K}_{i,p}\int_0^t \tilde{\mathbf{p}}^{p_*}\,dt \tag{1}$$

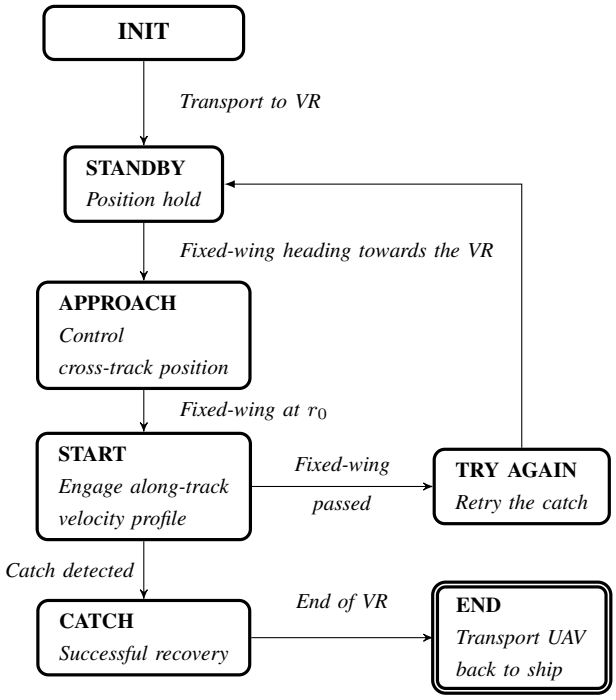

Fig. 4. Supervisor state-machine. The supervisor monitors the position of the fixed-wing UAV along the *virtual runway* (VR), and starts the coordination controller. When the along-track distance reaches r_0 (Figure 6), the START state is initiated.

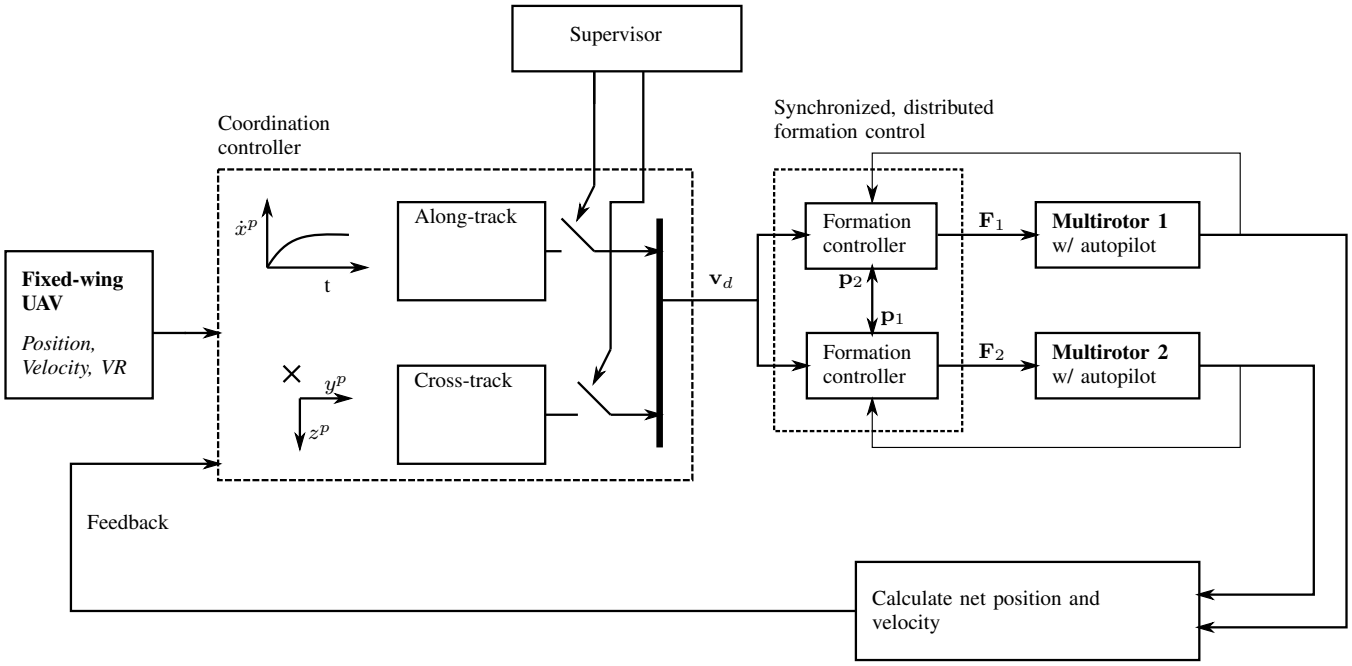

Fig. 5. Illustration of the control-structure. The cross-track controller gets feedback from the position of the fixed-wing UAV in a plane orthogonal to the virtual runway. The along-track controller is an open-loop controller, initiated by the supervisor when the fixed-wing UAV reaches the virtual runway.

where $\mathbf{K}_{j,p} \in \mathbb{R}^{2\times 2}$ for $j \in \{p, i, d\}$. The desired position $\mathbf{p}_d^{p_*} = \mathbf{p}_f^{p_*} = \mathbf{p}_{f,2:3}$ is defined as the current position of the fixed-wing UAV projected along the virtual runway to the cross-track plane.
It should be noted that the net position is not measured explicitly, and furthermore it is not a desirable control target as the net will swing during the transit. Therefore we seek to control the position $\bar{\mathbf{p}}$ as illustrated in Figure 3 as the circle in the cross-track plane. Hence, $\mathbf{p}^{p_*} = \bar{\mathbf{p}}_{2:3}$.

D. Coordination - Along-track

The relative velocity between the net and the fixed-wing UAV is reduced by accelerating the net to a desired velocity. In order to control the point of impact an open loop scheme is proposed.
Firstly, the along-track velocity of the fixed-wing UAV is assumed to be constant, then a desired along-track velocity profile is defined. By integrating these the along-track position profile can be found analytically for both vehicles. Finally the desired catch point $r_c = r_c(t_f)$ is given as a function of the desired start point $r_0 = r_0(t_0)$ as illustrated in Figure 6.

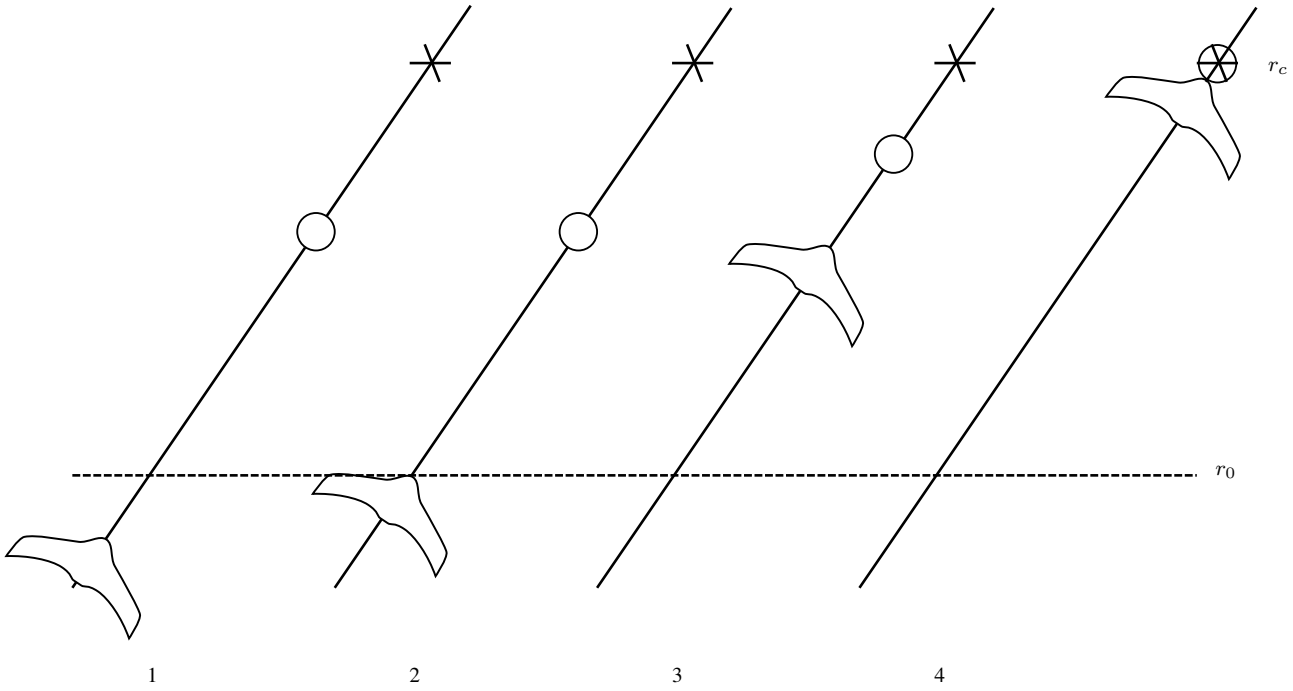

Fig. 6. Figure showing the timing of the along-track velocity, where the current position of the net is marked with a circle at different instances of time (1)-(4). When the fixed-wing UAV reaches r_0, the multirotors starts the velocity profile for forward flight as to intercept the fixed-wing UAV at r_c.

Different methods can be used to create a feasible velocity profile as the ultimate goal is to be able to calculate r_0.
These include, among others, ramps and reference models. Here a polynomial approach will be derived. First, we define the desired along-track velocity profile $v^p_{d,x}(t) := v(t)$ as an Nth-order polynomial

$$v(t) = \sum_{k=0}^{N} a_k (t - t_0)^k \tag{2}$$

Next, we seek to constrain the position, velocity, acceleration and jerk along that trajectory. This trajectory $\mathbf{q}(t) = \begin{bmatrix} p(t) & v(t) & a(t) & j(t) \end{bmatrix}^\top \in \mathbb{R}^4$ can be found by integrating and differentiating (2). Then, by defining the constraints $\mathbf{q}(t_0)$ and $\mathbf{q}(t_f)$, the N coefficients a_k can be found as derived in e.g. [15]. With a total of 8 constraints a polynomial of order 7 is required. In Figure 7 an example of such an trajectory is shown.
By combining the desired velocity from the cross-track and along-track control we get $\mathbf{v}^p_d = \begin{bmatrix} v^p_{d,x}(t) & (\mathbf{v}^{p_*}_d)^\top \end{bmatrix}^\top$, and the resulting desired velocity in $\{n\}$ can be found by the following transformation

$$\mathbf{v}^n_d = (\mathbf{R}^p_n)^\top \mathbf{v}^p_d \tag{3}$$

which gives the desired velocity for the two multirotors, and is applied to the formation in the next section.

E. Multirotor Modeling

In this section, we start by presenting the dynamical model of a multirotor UAV, as in [16]. The model can be derived by Newtonian or Lagrangian methods, and readers are referred to [17] for details on its derivation. By further assuming the presence of an internal attitude controller, the relevant dynamics for the control design is extracted.

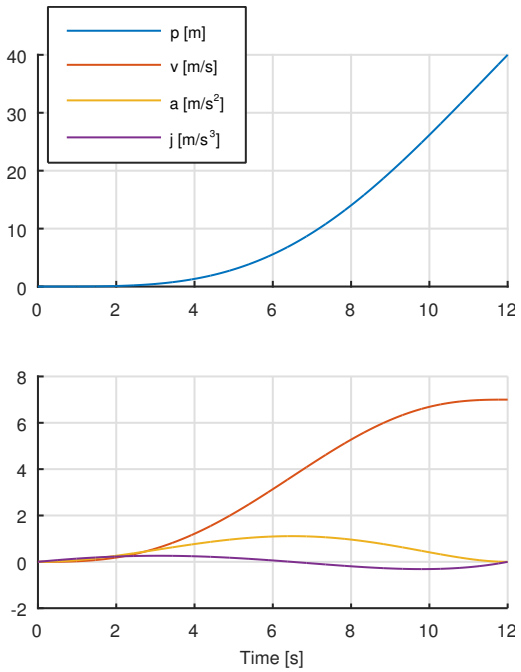

Fig. 7. Along-track polynomial velocity profile. Here the desired duration $t_f - t_0 = 12$, constraints $\mathbf{q}(t_0 = 0) = \mathbf{0}$ and $\mathbf{q}(t_f) = \begin{bmatrix} 40, & 7, & 0, & 0 \end{bmatrix}^\top$ are specified.

Let the dynamics of multirotor i be modeled by

$$\dot{\mathbf{p}}_i = \mathbf{v}_i \tag{4}$$
$$m_i \dot{\mathbf{v}}_i = m_i \mathbf{g} + \mathbf{R}_i f_i \tag{5}$$
$$\dot{\mathbf{R}}_i = \mathbf{R}_i \mathbf{S}(\boldsymbol{\omega}_i) \tag{6}$$
$$\mathbf{I}_i \dot{\boldsymbol{\omega}}_i = \mathbf{S}(\mathbf{I}_i \boldsymbol{\omega}_i)\boldsymbol{\omega}_i + \mathbf{M}_i \tag{7}$$

where $\mathbf{p}_i \in \mathbb{R}^3$ is the UAV position in the inertial frame $\{n\}$, $\mathbf{v}_i \in \mathbb{R}^3$ the translational velocity in $\{n\}$, $\mathbf{R}_i \in \mathcal{SO}^3$ a rotation matrix from the body-fixed frame $\{b_i\}$ to the inertial frame $\{n\}$, $\omega_i \in \mathbb{R}^3$ the angular velocity of the UAV, represented in $\{b_i\}$. Further, the operator $\mathbf{S}(\cdot)$ is the skew-symmetric transformation, such that $p \times q = \mathbf{S}(p)q$. m_i is the mass of the multirotor, and I_i the body-fixed inertia matrix. f_i is upwards thrust directed along the negative body-aligned z-axis, $\mathbf{M}_i$ are applied moment about the multirotor centre of gravity, and $\mathbf{g} = [0\ 0\ g]^\top$ where g is the gravitational constant.
Consider now the net being suspended in the centre of gravity of the UAV. This will affect the translational motion (5) by a force $\boldsymbol{\tau}_{L,i}$, given by the load dynamics, but the rotational motion (7) is unaffected. As control of the attitude of the multirotor is not considered in this paper, the model considering the translational motion is now

$$m_i \dot{\mathbf{v}}_i = m_i \mathbf{g} + \mathbf{R}_i f_i + \boldsymbol{\tau}_{L,i} \tag{5b}$$

Further, assume now that a sufficiently fast attitude controller is present. The direction of the applied force for translational motion (5) is given by $\mathbf{R}_i$, and by manipulating the roll and pitch of the UAV we can apply force in a desired direction. An example of such a controller is given in [17], and a similar controller is assumed present in the multirotor autopilot. Thus, the term $\mathbf{R}_i f_i$ can be replaced by an inertial control force $\mathbf{F}_i \in \mathbb{R}^3$, resulting in the linear dynamics

$$m_i \dot{\mathbf{v}} = m_i \mathbf{g} + \mathbf{F}_i + \boldsymbol{\tau}_{L,i} \tag{5c}$$

F. Formation Control of Two Multirotors

The formation controller is designed in two steps, following the procedure outlined in [18]. This is a passivity-based approach, where an inner loop controller takes a velocity setpoint from an outer controller, and the stability of the cascaded structure is proved by passivity theory. While the inner controller uses only its own measurements, the outer uses available information from the other multirotor to reach the desired formation.
First, let

$$\mathbf{F}_i = \boldsymbol{\tau}_{L,i} + m_i \mathbf{g} - \mathbf{K}_i(\mathbf{v}_i - \mathbf{v}_d) + m_i \dot{\mathbf{v}}_d + \mathbf{u}_i \tag{8}$$

where $\mathbf{v}_d$ is the desired common velocity from the coordination controller, known to both vehicles. $\mathbf{u}_i \in \mathbb{R}^3$ is the input from the outer loop formation controller, which acts as an injection to achieve a desired formation, to be specified later. Note that we assume we can measure the disturbance force $\boldsymbol{\tau}_{L,i}$ of the suspended net, so it can be compensated using feed-forward by the controller as discussed in Section VI-D.

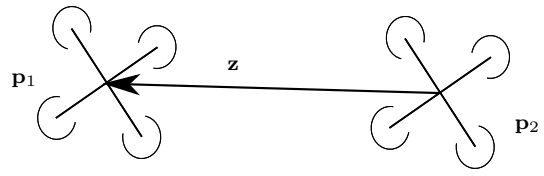

Fig. 8. Illustration of the vector between the two multirotors.

Next, let $\mathbf{z} := \mathbf{p}_1 - \mathbf{p}_2$ be the vector between the two multirotors in $\{n\}$, as in Figure 8. Let the desired value $\mathbf{z}_d$ be given by a length l and rotation ψ_n about $\vec{z}$ as

$$\mathbf{z}_d = \begin{bmatrix} l\cos(\psi_n) & l\sin(\psi_n) & 0 \end{bmatrix}^\top \tag{9}$$

The standard linear consensus protocol ([18]) can now be applied as

$$\mathbf{u}_i = d_i \mathbf{K}_p(\mathbf{z} - \mathbf{z}_d) \tag{10}$$

where $d_1 = 1, d_2 = -1$. We can now state the main result of this section:

Proposition 1. *The inner control loop* (8) *combined with the consensus protocol* (10) *ensures that the equilibrium point*

$$\mathbf{v}_1 = \mathbf{v}_2 = \mathbf{v}_d \tag{11}$$

$$\mathbf{z} = \mathbf{z}_d \tag{12}$$

is globally asymptotically stable.

Proof. Direct application of Corollary 2.2 in [18]. □

IV. Multi-body Simulation Model

This section discusses the dynamical models of the combined multirotor-net system. To provide a thorough understanding of the dynamical motion during the recovery maneuver, we developed a simulator that includes the full 6-DOF dynamics of the multirotors, fixed-wing UAV, and the net suspended under the multirotors. We also model the impact forces during collision.

A. Tension from the Suspended Net

By having the net attached to the multirotors, we have in effect a system of constrained motion where each wire connecting the net removes one degree of freedom. We consider the net being a rigid body. To model this behavior, one can reduce the state-space and use only *generalized coordinates* that cover the *configuration space* [19]. This, however, will hide the forces acting on the wires during impact. Instead, we chose to model the interconnected system with *constrained* coordinates. The Udwadia-Kalaba equation is presented in [20], which is a way to explicitly calculate the forces of constraints acting on each body. This methodology was used in [21] to develop equations of motions for helicopter slung load systems and [22] for multirotors. We follow a similar approach, described in this section. For textual brevity, we have omitted the torques resulting by the constraints due to attaching wires away from the centre of gravity of each body. They are, however, included in the numerical simulations conducted in Section V.

For generality, we consider the case with N multirotors connected to a common suspended load (in our case, a net). Let wire i, connecting body (multirotor) i to the load. A vector along the wire is given by

$$\mathbf{L}_i^n = \mathbf{p}_i - \mathbf{p}_n \tag{13}$$

where $\mathbf{p}_n$ is the position of the suspended load in $\{n\}$. A constraint g_i acting on body i and the load is given by

$$g_i = ||\mathbf{L}_i||^2 - d_i^2 = 0 \tag{14}$$

where d_i is the nominal length of wire i. (14) can be differentiated twice to obtain

$$\dot{g}_i = 2\dot{\mathbf{L}}_i^\top \mathbf{L}_i \tag{15}$$

$$\ddot{g}_i = 2\ddot{\mathbf{L}}_i^\top \mathbf{L}_i + 2\dot{\mathbf{L}}_i^\top \dot{\mathbf{L}}_i = 0 \tag{16}$$

By defining the concatenated position- and velocity vector $\mathbf{p} := [\mathbf{p}_1^\top, \ldots, \mathbf{p}_N^\top]^\top$, $\mathbf{v} := [\mathbf{v}_1^\top, \ldots, \mathbf{v}_N^\top]^\top$, the constraint can now be put on *standard form* [19]:

$$\mathbf{A}_i(\mathbf{p}, \mathbf{v})\dot{\mathbf{v}} = \mathbf{b}_i(\mathbf{p}, \mathbf{v}) \tag{17}$$

where

$$\mathbf{A}_i = 2\mathbf{L}_i^\top \begin{bmatrix} \mathbf{0}_{3\times 3(i-1)} & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3(N-i)} & -\mathbf{I}_{3\times 3} \end{bmatrix} \tag{18}$$

and

$$\mathbf{b}_i = -\dot{\mathbf{L}}_i^\top \dot{\mathbf{L}}_i \tag{19}$$

According to [23], the constraint forces $\boldsymbol{\tau}_L$ acting on all the bodies are now given by:

$$\boldsymbol{\tau}_L = \mathbf{M}^{1/2}(\mathbf{A}\mathbf{M}^{-1/2})^+(\mathbf{b} - \mathbf{A}\dot{\mathbf{v}}) \tag{20}$$

where $\mathbf{A}$ and $\mathbf{b}$ are concatenations of $\mathbf{A}_i$ and $\mathbf{b}_i$, respectively, $(\cdot)^+$ denotes the MoOre-Penrose pseudo inverse, and $\mathbf{M}$ is a diagonal matrix with the masses of the involved bodies. To include the effects on the attitude dynamics by torques from attachment points, one can follow a similar procedure as listed above. These are more involved expressions, as they include states in the body-fixed coordinate systems. Readers are referred to [23] and [21] for details on derivation. Note that the results from the numerical simulations presented in the next section, includes the full attitude dynamics as well.

B. Modeling the Fixed-Wing UAV Impact

This section studies the dynamics during the impact, when the fixed-wing UAV gets arrested by the suspended net. The collision is assumed to be perfectly inelastic such that the bodies will stick together after the collision. In order to calculate the forces and moments on the suspended net, conservation of momentum is applied. The impact is assumed to affect the system in the timespan $t \in [t_-, t_+]$, where we define the duration $\Delta t = t_+ - t_-$.
Further it assumed that linear and angular momentum, denoted as $\mathbf{P} = m\mathbf{v}$ and $\mathbf{L} = \mathbf{I}\boldsymbol{\omega}$ respectively, is conserved. Thus, the sum of momentum directly after $()_+$ and before $()_-$ is equal, with:

$$\mathbf{P}_+ = \mathbf{P}_- = m_f\mathbf{v}_{f,-} + m_n\mathbf{v}_{n,-} \tag{21}$$

$$\mathbf{L}_+ = \mathbf{L}_- = \mathbf{I}_f\boldsymbol{\omega}_{f,-} + \mathbf{I}_n\boldsymbol{\omega}_{n,-} \tag{22}$$

where n denotes the suspended load. Next, the common velocity $\mathbf{v}_+$ and angular velocity $\boldsymbol{\omega}_+$ after the collision can be found by

$$\mathbf{v}_+ = \frac{\mathbf{P}_-}{m_f + m_n} \tag{23}$$

$$\boldsymbol{\omega}_+ = (\mathbf{I}_f + \mathbf{I}_n)^{-1}\mathbf{L}_- \tag{24}$$

Finally, the average forces and moments applied to the suspended load can be found by utilizing the linear and angular impulse law. The linear and angular impulse $\mathbf{J} = \Delta\mathbf{P} = \mathbf{P}_+ - \mathbf{P}_-$ and $\mathbf{H} = \Delta\mathbf{L} = \mathbf{L}_+ - \mathbf{L}_-$ gives the change in momentum. The laws are coupled with forces and moments $\boldsymbol{\tau} = \begin{bmatrix}\mathbf{f}^\top & \mathbf{m}^\top\end{bmatrix}^\top$ over the timespan t_- to t_+ as follows

$$\begin{bmatrix}\mathbf{J} \\ \mathbf{H}\end{bmatrix} = \int_{t_-}^{t_+} \boldsymbol{\tau}\, dt = \bar{\boldsymbol{\tau}}\Delta t \tag{25}$$

Then, the average forces and moments on the suspended load $\bar{\boldsymbol{\tau}}_n$ is given as:

$$\bar{\boldsymbol{\tau}}_n = \frac{1}{\Delta t}\begin{bmatrix}(m_f + m_n)\mathbf{v}_+ - m_n\mathbf{v}_{n,-} \\ (\mathbf{I}_f + \mathbf{I}_n)\boldsymbol{\omega}_+ - \mathbf{I}_n\boldsymbol{\omega}_{n,-}\end{bmatrix} \tag{26}$$

V. Simulation and Results

In this section, we present the results from a numerical simulation using the controllers and models presented in the previous sections. In this case, we consider two multirotors, with a mass of $m_{1,2} = 6$ kg, recovering an incoming fixed-wing UAV at $m_f = 3$ kg. The fixed-wing UAV is approaching at a constant speed of 15 m/s, and the multirotors are set to reach an approach-speed of 7 m/s. Further, the multirotors are equipped with a basic autopilot that handles attitude setpoints, as discussed in Section III-E which is implemented as a PD control structure. This is a similar structure as in the autopilot to be used for experiments, discussed later in Section VI. Next, we consider a net with a width and height of 5 and 3 m, respectively. The numerical simulation is conducted in MATLAB, using *Runge Kutta 4* as integrator at 50 Hz. The total thrust of each multirotor is configured such that it uses half of the available power at hover. Due to the construction of the multirotor, the available torque is likewise limited so that each motor does not exceed its maximum. The multirotor has a

motor-to-motor diameter of 1 m. Further, discrete time sampling is implemented with a *zero-order-hold*. The net is attached 10 cm below the Centre of Gravity of each multirotor.

In Figure 9, snapshots of the dynamics during the catch is shown. We can see that the multirotors successfully intercept the incoming fixed-wing UAV, and are able to handle the load during impact. The tension force on the left multirotor can be seen in Figure 10, where the oscillations of the load can be clearly seen. The steady value in the z axis conform to half the weight of combined net and fixed-wing UAV. Due to a slight twist in the net when it swings, a slight transient can be seen on the y-component of the tension force. Further, in Figure 11, the along-track velocity of the multirotors are shown. We can see that right after impact, some residual oscillations remains due to the swinging payload.

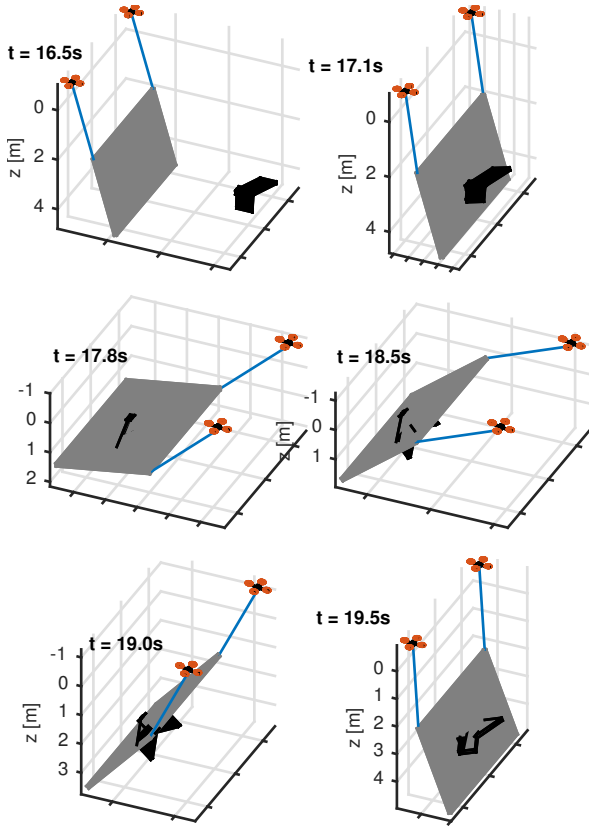

Fig. 9. Snapshots of a simulation run at right before and after the fixed-wing UAV is intercepted by the suspended net.

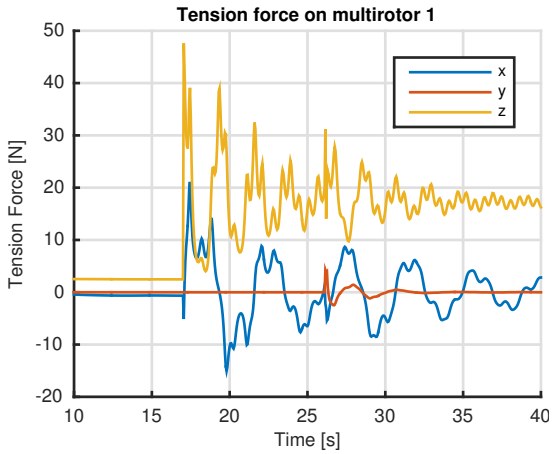

Fig. 10. The tension force on the wire connected to multirotor 1, in all three dimensions. The stationary value of z corresponds to half the total weight of the net and the arrested fixed-wing UAV.

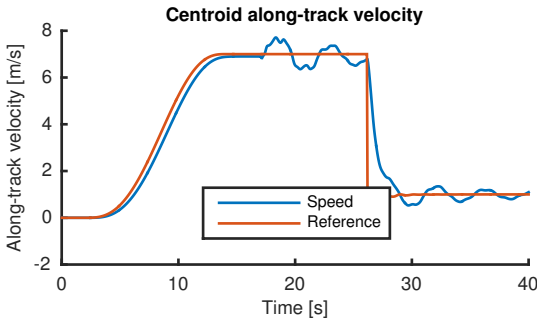

Fig. 11. Resulting velocity along the virtual runway.

TABLE I
SKYWALKER X8 DATA

Airframe weight	3 kg
Maximum takeoff weight	4.2 kg
Cruise speed	18 m/s
Flight time	45 - 60 minutes
Control Surfaces	Elevons (combined aileron and elevator)
Autpilot	Pixhawk w/ ArduPlane

TABLE II
DJI S1000+ DATA

Airframe weight	4 kg
Maximum takeoff weight	11 kg
Flight time	15-25 minutes
Autopilot	Pixhawk w/ ArduCopter

VI. SYSTEMS ARCHITECTURE

This section gives an overview of the architecture of the experimental system, including the different vehicles used, autopilots and communication links. A detailed description of the base system architecture can be seen in [24].

A. Airframes

In this research, we utilize the *Skywalker X8* as our fixed-wing UAV. This is a light-weight low-cost flying-wing type UAV made out of styrofoam, making it an ideal platform for research purposes. The main characteristics are summarized in Table I, and seen in Figure 12. For a multirotor platform, a versatile vehicle with plenty of lift capacity is needed. We are using the *S1000+*, made by DJI. This is a octocopter type design with eight arms and motors, as can be seen in Figure 13. The main data can be seen in Table II.

Fig. 12. The Skywalker X8 in flight.

B. Autopilot and Payload Computer

Both the fixed-wing and the two multirotors are controlled by the same basic hardware. We are using the Pixhawk [25] autopilot hardware, configured with Ardupilot [26] software as the low-level autopilot. This setup can be configured for a multitude of different vehicle types, and handles basic navigation and control, such as sensor fusion from accelerometers

Fig. 13. DJI S1000+ frame. *Image courtesy of dji.com*

and gyros, and attitude control. In addition, the vehicles are fitted with an onboard Linux Computer, running customized software. This sends commands and reference values to the autopilot. On the computer, we utilize the LSTS toolchain [27] as a framework for controller implementation. The toolchain is developed by the *Underwater Systems and Technology Laboratory* at University of Porto, Portugal, and is available open-source at Github [28]. The main component is DUNE, which is a modular software framework for autonomous control. The toolchain also contains a ground control segment (Neptus), messaging protocol (IMC), and a Linux distribution (Glued). The components communicate as illustrated in Figure 14.

C. Navigation

The autopilot internally fuses data from a MEMS-based IMU with a magnetometer and GNSS to provide a full state attitude and position reference solution using an Extended Kalman Filter. As described earlier, the position acquired from traditional GNSS is without the required accuracy to do precision landing and formation flight [13]. However, by using real-time kinematic (RTK) techniques with a differential correction, centimeter-level real-time positioning can be achieved due to the short signal wavelength (19 cm for GPS L1) of GNSS signals. This requires that integer carrier phase ambiguities are successfully resolved [13]. RTKLIB [29] is an open-source library for computing these ambiguities, and providing real-time position updates using raw data from a GNSS receiver, in our case the u-blox M8T [30]. The same receiver is used at both the base station and in the vehicles.

D. Network, Net and Custom Sensors

The vehicles communicate over a wireless 5.8 GHz network, using a radio from Ubiquity Networks. This is an ethernet/IP based radio, with *Time Division Multiple Access* medium control. This ensures constant transfer-delay in the network.
To keep the weight of the net low, but still have it strong enough to sustain multiple recovery missions, we chose a net made from *Polyethylene*. With a mask size of 12 cm, it weights only 45 g/m^2. To have a reasonable margin for error, we chose a net size of 3 times 5 m. We will equip hooks along the wings and in the nose of the fixed-wing UAV. For safety, we have made a device to release the suspended net from the multirotors.
To measure the load of the net to the multirotors, we have constructed a device to measure the tension and angle of the attached wire. The angles are measured using two digital magnetic encoders (MTS 360 from *PIHER* [31]) in a gimbal-like structure, while the tension is measured using a light-weight load cell (LSB200 from *Futek* [32]).

VII. Software in the Loop Simulations

As a preparation for experiments, the control software is tested in a *Software In the Loop* (SIL) simulation procedure. Here, the software is the same as when running on the target vehicle, but where the dynamics and response are done by a simulator. Our control software is implemented in C++, and by using DUNE makes it straight forward to run the software in a SIL environment. Further, one of the benefits of using the open-source autopilot *ardupilot*, is that is has an option to run the autopilot code with an accompanied simulator in SIL as well. This makes all the software interfaces and communication channels identical, which makes it a good test of the implementation.
In this test, we investigate the performance of the control software with the focus on the coordination task. While the simulation presented in Section V included the complete dynamics of the suspended net, including the impact forces, there is no net in this test. We will however be able to detect that the fixed-wing UAV would have been arrested by the net given its position relative to the multirotors.
The results of the simulation can be seen in Figures 15–17. Figure 15 shows the trajectory of the fixed-wing UAV as it approaches the virtual runway. As it comes closer to the multirotors, the coordinated multirotors start the along-track velocity

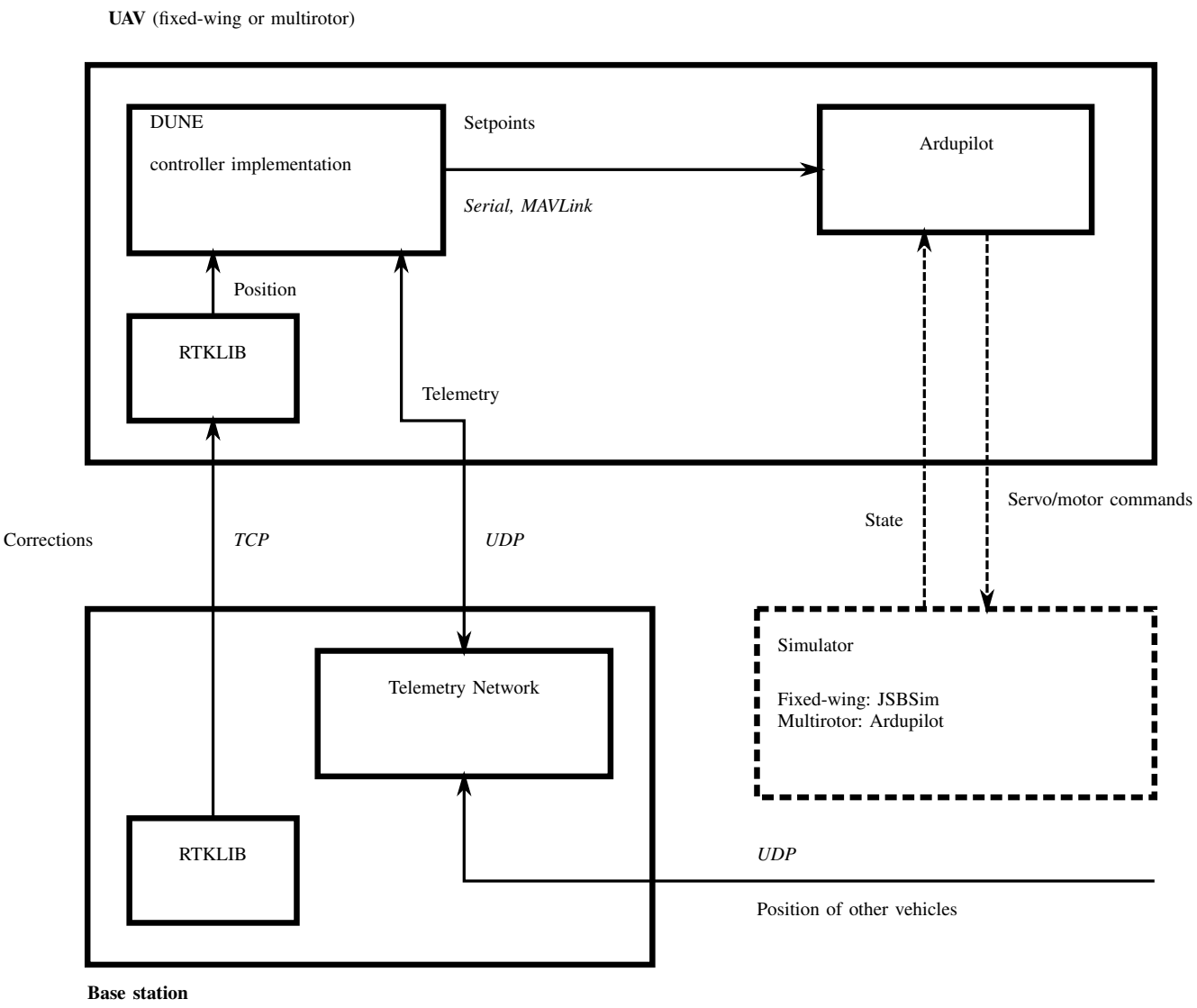

Fig. 14. Overview of the different elements involved in the UAV control system. A base station runs a server hosting corrections for RTK GNSS, and which is transmitted to each vehicle. The telemetry network is based on a 5.8 GHz ethernet/IP radio link, with *time division multiple access* as medium control. DUNE, running our control software in the vehicle on a *Beaglebone Black*, transmits setpoints to the autopilot. In the figure, *italic text* denotes a transport protocol.

profile to intercept the fixed-wing UAV at the prescribed speed. Figure 16 gives a closer look at this phase. As noted above, this simulation does not contain any net dynamics. Thus, in Figure 17, we see the relative distance between the vehicles at the time the fixed-wing UAV would have hit the suspended net.

VIII. Conclusions

In this paper, we have presented a concept for recovery of a fixed-wing UAV in a net suspended by two multirotors. We have suggested a control design to comply with the concept. Further, numerical simulations which includes the full non-linear dynamics of the multirotors and the suspended net were conducted, which showed the feasibility of the controller. Further, we have suggested an implementation strategy, and the control system was implemented and tested in a SIL-setup which included the interface to the autopilots.

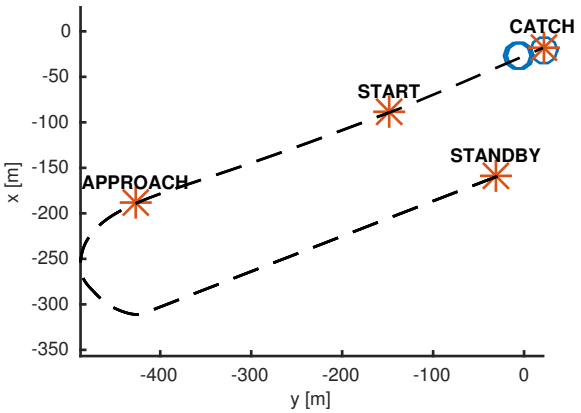

Fig. 15. An overview of the *software in the loop* simulation. The red cross marks the location of the fixed-wing UAV, while the blue circle represents the centroid location of the multirotors. The state of the supervisor at is marked at the corresponding position of the fixed-wing UAV.

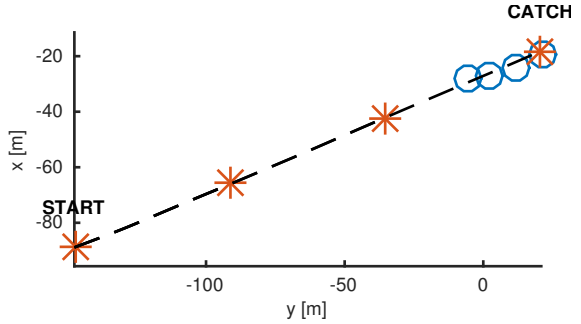

Fig. 16. A closer look at the simulation in the catch-phase. START marks the start of the multirotor along-track velocity, as to intercept the incoming fixed-wing UAV at the prescribed speed.

REFERENCES

[1] S. Pullen, P. Enge, and J. Lee, "Local-Area Differential GNSS Architectures Optimized to Support Unmanned Aerial Vehicles (UAVs)," in *Proceedings of the International Technical Meeting of The Institute of Navigation,*, San Diego, CA, 2013.

[2] S. Huh and D. H. Shim, "A Vision-Based Automatic Landing Method for Fixed-Wing UAVs," *Journal of Intelligent and Robotic Systems*, vol. 57, pp. 217–231, 2010.

[3] Insitu.com, "Insitu - ScanEagle," 2016. [Online]. Available: http://www.insitu.com/information-delivery/unmanned-systems/scaneagle

[4] Spectrum.ieee.org, "Watch This Massive Drone Launch and Recover Another Drone in Flight," 2015. [Online]. Available: http://spectrum.ieee.org/automaton/robotics/drones/insitu-flares-drone-launch-and-recovery

[5] N. Sarigul-klijn and M. M. Sarigul-klijn, "A Novel Sea Launch and Recovery Concept for fixed wing UAVs," in *54th AIAA Aerospace Sciences Meeting*, no. 4-8 January, San Diego, California, USA, 2016, pp. 1–11.

[6] A. G. Sim, "Flight Characteristics of a Manned, Low-Speed, Controlled Deep Stall Vehicle," *NASA Technical Memorandum*, pp. 1–10, 1984.

[7] H. Taniguchi, "Analysis of deepstall landing for uav," *26Th International Congress of the Aeronautical Sciences*, pp. 1–6, 2008.

[8] S. H. Mathisen, K. Gryte, T. Johansen, and T. I. Fossen, "Non-linear Model Predictive Control for Longitudinal and Lateral Guidance of a Small Fixed-Wing UAV in Precision Deep Stall Landing," in *AIAA Guidance, Navigation, and Control Conference*, San Diego, 2016.

[9] Delftdynamics.nl, "DroneCatcher catches drone," 2015. [Online]. Available: http://www.delftdynamics.nl/index.php/en/news-en/117-dronecatcher-catches-drone

[10] Mtu.edu, "Michigan Tech Robotic Falconry," 2016. [Online]. Available: http://me.sites.mtu.edu/rastgaar/home/news/

[11] Theverge.com, "Tokyo police unveil net-wielding interceptor drone," 2015. [Online]. Available: http://www.theverge.com/2015/12/11/9891128/tokyo-interceptor-net-drone

[12] Nextgenerationvision.fr, "Drone interception," 2014. [Online]. Available: https://youtu.be/APWG3VEGbJw

[13] R. Skulstad, C. Syversen, M. Merz, N. Sokolova, T. Fossen, and T. Johansen, "Autonomous Net Recovery of Fixed-Wing UAV with Single-Frequency Carrier-Phase Differential GNSS," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 30, no. 5, pp. 18 – 27, 2015.

[14] T. I. Fossen, *Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.

[15] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, 3rd ed. Wiley New York, 2006.

[16] K. Klausen, T. I. Fossen, and T. A. Johansen, "Nonlinear Control of a Multirotor UAV with Suspended Load," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. Denver, CO: IEEE, 2015, pp. 176 – 184.

[17] R. Mahony, V. Kumar, and P. Corke, "Modeling, Estimation, and Control of Quadrotor," *IEEE Robotics and Automation magazine*, vol. 19, no. 3, pp. 20–32, sep 2012.

[18] H. Bai, M. Arcak, and J. T. Wen, *Cooperative control design: A systematic, passivity-based approach*. Springer-Verlag New York, 2011.

[19] J. Ginsberg, *Engineering Dynamics*. Cambridge University Press, 2008.

[20] F. Udwadia and R. Kalaba, "A new perspective on constrained motion," *Proceedings: Mathematical and Physical Sciences*, vol. 439, no. 2, pp. 407–410, 1992.

[21] M. Bisgaard, J. D. Bendtsen, and A. L. Cour-Harbo, "Modeling of Generic Slung Load System," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 2, pp. 573–585, mar 2009.

[22] K. Klausen, T. I. Fossen, T. A. Johansen, and A. P. Aguiar, "Cooperative path-following for multirotor UAVs with a suspended payload," *2015 IEEE Conference on Control Applications (CCA)*, pp. 1354–1360, 2015.

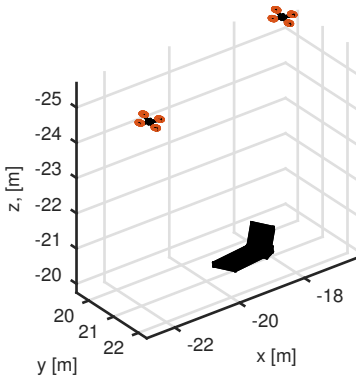

Fig. 17. A snapshot of the multirotors and fixed-wing UAV at the moment when the fixed-wing UAV would have hit the net.

[23] F. E. Udwadia and P. Phohomsiri, "Explicit Poincaré equations of motion for general constrained systems. Part I. Analytical results," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 463, no. 2082, pp. 1421–1434, jun 2007.

[24] A. Zolich, T. A. Johansen, K. Cisek, and K. Klausen, "Unmanned Aerial System Architecture for Maritime Missions . Design & Hardware Description," in *IEEE RED-UAS Conference*, 2015.

[25] Pixhawk.ethz.ch, "PX4 Pixhawk," 2016. [Online]. Available: https://pixhawk.ethz.ch/

[26] Ardupilot.com, "Ardupilot - Open source autopilot," 2016. [Online]. Available: http://ardupilot.com

[27] J. Pinto, P. S. Dias, R. Martins, J. Fortuna, E. Marques, and J. Sousa, "The LSTS toolchain for networked vehicle systems," *OCEANS 2013 MTS/IEEE Bergen: The Challenges of the Northern Dimension*, 2013.

[28] Github.com, "LSTS: Underwater Systems and Technology Laboratory," 2016. [Online]. Available: https://github.com/LSTS/

[29] T. Takasu and A. Yasuda, "Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB," in *Proceedings of the International Symposium on GPS/GNSS*, Jeju, Korea, 2009.

[30] U-blox.com, "NEO/LEA-M8T — u-blox," 2016.

[31] Piher.net, "MTS-360," 2016. [Online]. Available: http://www.piher.net/

[32] Futek.com, "LSB200 S-Beam," 2016. [Online]. Available: http://www.futek.com/lsb200/overview.aspx