Computer Methods in Applied Mechanics and Engineering 442 (2025) 117999

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

A machine-learning enabled digital-twin framework for tactical drone-swarm design

T.I. Zohdi

Department of Mechanical Engineering, University of California, Berkeley, CA, 94720-1740, USA

ARTICLE INFO

Keywords:
Drone-swarms
Machine-learning
Multibody dynamics
Structural-dynamics

ABSTRACT

The goal of this work is to develop a machine-learning enabled digital-twin to rapidly ascertain optimal programming to achieve desired tactical multi-drone swarmlike behavior. There are two main components of this work. The *first main component* is a framework comprised of a multibody dynamics model for multiple interacting agents, augmented with a machine-learning paradigm that is based on the capability of agents to identify (a) desired targets, (b) obstacles and (c) fellow agents, as well as the resulting collective actions of the drone-swarm of agents. The objective is to construct a system with entirely autonomous behavior by optimizing the actuation parameter values that are embedded within the coupled multibody differential equations for drone-swarm dynamics. This is achieved by minimizing a cost-error function that represents the difference between the simulated overall group behavior and in-field behavior from observed *ground truth* synthetic data in the form of temporal snapshots corresponding to multiple camera frames. The *second main component* of the analysis is to deeply assess the structural performance of drone-swarm members, by studying chassis design, deployment and dynamic-structural performance. As an example, we investigate a tactical quadcopter drone under attack, specifically by subjecting it to series of launched explosions. A Discrete Element Method (DEM) is developed to rapidly design a quadcopter of any complex shape, attach motors and then to subject it to a hostile environment, in order to ascertain its performance. The method also allows one to describe structural damage to the quadcopter drone, its loss of functionality (thrust), etc. Furthermore, the use of DEM can also capture fragmentation of the quadcopter and can ascertain the resulting debris field. Numerical examples are provided to illustrate the two components of the overall model, the computational algorithm and its ease of implementation.

1. Introduction

Research on drones started in the early 1900s, and was initially oriented towards military applications. This research accelerated during World War II, in order to train antiaircraft gunners and to fly attack missions. However, with the exception of the V-2 (Vergeltungswaffe/vengeance weapon) rocket system program in Germany, they were primarily miniature airplanes. It was not until the 1960s, with a variety of military conflicts and concern about losing pilots over hostile territory, that drone research started to grow rapidly. Over the last 20 years, interest in drones has grown dramatically and they have become an integral part of many societal, industrial and defense portfolios. Generally speaking, most drones that are deployed for long distance operations are fixed wing aircraft, while rotorcraft, such as quadcopters, are used for precise “stop and go” operations. We note that cruise missiles are not considered to be drones, since they are the munition payload, although recently many “kamikaze drones” have been deployed

E-mail address: zohdi@berkeley.edu.

https://doi.org/10.1016/j.cma.2025.117999
Received 23 February 2025; Received in revised form 20 March 2025; Accepted 3 April 2025
Available online 2 May 2025
0045-7825/© 2025 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

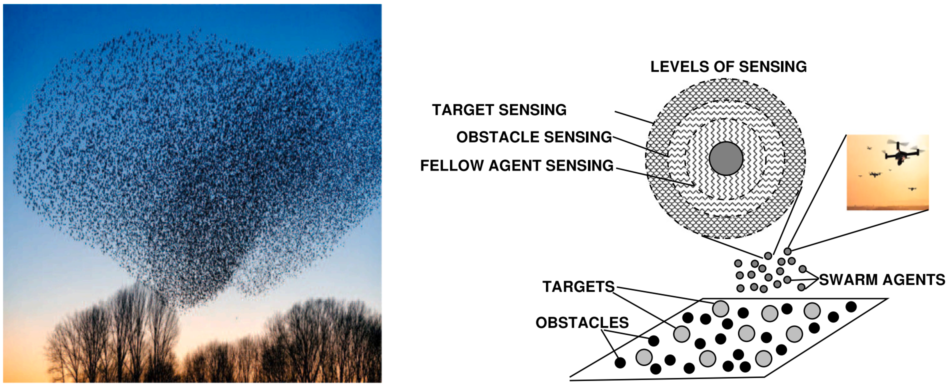

Fig. 1. Natural (left) swarm behavior (starlings, Sturnus Vulgaris, free stock photo from Pixabay) and (right) synthetic drone-swarms. The objective is to construct desired collective behavior by manipulating individual characteristics.

in armed conflicts and the difference between the two is now debatable. Drones have varying degrees of autonomy, ranging from (a) complete remote control by humans to (b) autopilot assistance to (c) completely independent autonomy, which include features such as (1) GPS waypoint navigation (2) preprogrammed routes (3) preprogrammed delivery (4) automated take-off and landing (5) failsafe landing and (6) return to home. Furthermore, depending on their intended use, flight duration, altitude, range, etc., their power sources could be any of the following (1) fuel-power sources, which are suitable for high-altitudes and long range operations, for example for military operations, (2) battery-power sources, which are for quiet applications in complex environments, with limited flight times, which make them ideal for agricultural facilities, industrial facilities and cities, (3) hybrid electric-fuel power sources, which are suitable for dual use operations where fuel and battery powered operations are needed and (4) solar-power sources, for extremely long duration and high-altitude operations that involve monitoring large areas. Under development are nuclear-powered and hydrogen fuel-cell powered systems, which have their own challenges, due to concerns over crash-safety.

Commercially, due to the steady increase in inexpensive drones and camera technology, there are a wide variety of non-military applications, such as world-wide anti-poaching and anti-whaling efforts. For example in oil and gas exploration, drones have been used for geophysical mapping, in particular geomagnetic surveys, where measurements of the Earth's varying magnetic field strength are used to calculate the nature of the underlying magnetic rock structure, in order to locate mineral deposits. Because of the huge areas associated with oil and gas pipelines, monitoring activity can be enhanced and accelerated by deployment of drones. In the field of archaeology, drones are used to accelerate surveying to protect sites from looters. Another direct application is cargo transport, which has been promoted by Amazon, DHL, Google, etc. The use of drones in agriculture is also advantageous for crop dusting, crop health monitoring and precision agriculture. In summary, the variety of uses is constantly growing and is immense. Current drone research is wide-ranging, spanning:

- **fundamentals**: aerodynamics, structural analysis, propulsion, acoustics, thermal analysis, etc.
- **simulation**: machine-learning, artificial intelligence, digital-twins, cameras, LiDAR imaging, onboard computing, etc.
- **controls**: sensing, data acquisition, communications, autonomy, navigation, power-supply, etc.
- **manufacturing**: fabrication methods, 3D-Printing, cost-analysis, scale-up, etc.
- **societal and industrial applications:** agricultural mapping, fire-fighting, defense, manufacturing, etc.
- **education**: pedagogy, training paradigms, literacy, access, outreach, etc.

We refer the reader to a wide cross-section of popular and technical literature on the subject [1–26].

1.1. Drone-swarms

Recently, due to the rise of AI and machine-learning, emphasis has now focussed on collaborative drone-swarm technologies. Accordingly, one main goal of this work is to develop a machine-learning enabled digital-twin to rapidly ascertain optimal programming for desired tactical multi-drone swarmlike behavior (Fig. 1). Swarm modeling has origins in the description of biological groups (flocks of birds, schools of fish, crowds of human beings, etc.), as. well as predators-prey relationships (Breder [27], 1952). In this work, we focus on decentralized paradigms where there is no leader, making the overall system less vulnerable. Early approaches that rely on decentralized organization can be found in Beni [28], Brooks [29], Dudek et al. [30], Cao et al. [31], Liu and Passino [32] and Turpin et al. [33]. Usual models incorporate a tradeoff between long-range interaction and short-range repulsion between individuals, dependent on the relative distance between individuals. The most basic model is to treat each individual as a point mass (Zohdi [34]), which we adopt here, and to allow the overall multi-body system to dynamically move in response to its environment, based on Newtonian mechanics (Gazi and Passino [35], Bender and Fenton [36], Kennedy and Eberhart [37] and

Zohdi [34], [38], [39], [40], [41]).[1] For some creatures, the "visual field" of individuals may play a significant role, while if the agents are robots or drones, the communication can be electronic.[2]

1.2. Camera technologies

Many drones now carry a variety of multispectral and LiDAR (Light Detection And Ranging) type cameras, acoustical sensors and associated signal processing tools. In particular, LiDAR has become quite popular and typically uses light in the high-frequency ultraviolet, visible and near infrared spectrum (Ring [48], Cracknell and Hayes [49], Goyer and Watson [50], Medina et al. [51] and Trickey et al. [52]). It is classified as a "time-of-flight" type technology, utilizing a pulse of light and the time of travel to determine the relative distance of an object. Over the last 20 years, these devices have steadily improved and have become quite lightweight [53–59]. There are a variety of time-of-flight technologies that have been developed, primarily for military reasons, of which Radar, Sonar and LiDAR are prime examples. The various types range from (1) conventional radar, (2) laser/radar altimeters, (3) ultrasound/sonar/seismograms, (4) radiometers and photometers-which measure emitted radiation, (5) hyperspectral cameras, where each pixel has a full spectrum and (6) geodetic-gravity detection, etc. For example, from satellites, the spatial resolution is on the order of pixel-sizes of 1-1000 meters using infrared wavelengths of 700-2100 nanometers. Hyperion-type cameras have even a broader range, 400–2500 nanometers with 200 bands (channels) and 100 nanometers per band. For example, thermographic/infrared cameras, form a heat-zone image (700nm−14000 nm), however, the focusing lens cannot be glass, and are typically made of germanium or sapphire. These devices are fragile and require coatings, making them expensive. There are two main thermographic camera types: (a) cameras using cooled infrared detectors, which need specialized semiconductors, and have a relatively high resolution and (b) cameras using uncooled detectors, sensors and thermo-electronic resistance, which have relatively lower resolution. Furthermore, the initial image is monochrome, and must be color-mapped. Additionally, there are a variety of "corrective" measures (post-processors), such as (1) radiometric enhancements, which improve the illumination for material properties, (2) topographic enhancements, which improve the reflectivity due to shade, sunniness, etc. and (3) atmospheric enhancements, which correct for atmospheric haze.[3] LiDAR has advantages because (1) the systems are simple, since they do not have moving parts associated with a scanner, and can thus be made very compact and can be used in real time and (2) the systems do not require sophisticated post-processing units and are therefore inexpensive. We refer the reader to Zohdi [60] for more details. Considering the above, an implicit, advantageous, task of a drone swarm, regardless of the specific type of camera technology, is to get close enough to multiple desired locations in order to take high-quality pictures.

1.3. Objectives

A much needed component in this field of research is an easy to use tool to design systems of interacting agents that collectively collaborate to achieve a desired task. This motivates the goal of the present work, which is to develop a machine-learning enabled digital-twin to rapidly ascertain optimal programming to achieve desired tactical multi-drone swarmlike behavior. There are two main components of this work:

- **Component 1** is a framework comprised of a multibody dynamics model for multiple interacting agents, augmented with a machine-learning paradigm that is based on the capability of agents to identify (a) desired targets, (b) obstacles and (c) fellow agents, as well as the resulting collective actions of the drone-swarm of agents. The objective is to construct a system with entirely autonomous behavior by optimizing the actuation parameter values that are embedded within the coupled multibody differential equations for drone-swarm dynamics. This is achieved by minimizing a cost-error function that represents the difference between the simulated overall group behavior and in-field behavior from observed *ground truth* synthetic data in the form of temporal snapshots corresponding to multiple camera frames.
- **Component 2** consists of an analysis to deeply assess the structural performance of drone-swarm members, by studying the chassis design, deployment and dynamic-structural performance. As an example, we investigate tactical quadcopter drones under attack, specifically by being subjected to series of launched explosions. A Discrete Element Method (DEM) is developed to rapidly design a quadcopter of any complex shape, attach motors and then to subject it to a hostile environment, in order to ascertain its performance. The method also allows one to describe structural damage to the quadcopter drone, its loss of functionality (thrust), etc. Furthermore, the use of DEM can also capture fragmentation of the quadcopter and can ascertain the resulting debris field.

[1] There are other modeling paradigms, for example mimicing ant colonies (Bonabeau et al. [42]), which exhibit foraging-type behavior and trail-laying-trail-following mechanisms for finding food sources (see Kennedy and Eberhart [37] and Bonabeau et al. [42], Dorigo et al. [43], Bonabeau et al. [42], Bonabeau and Meyer [44] and Fiorelli et al. [45]).

[2] However, in some systems, agents interact with a specific set of other agents, *regardless* of whether they are far away (Feder [46]). This appears to be the case for Starlings (Sturnus vulgaris). In Ballerini et al. [47], the authors concluded, that such birds communicate with a certain number of birds surrounding it and that the interactions are governed by topological distances and not metric distances.

[3] There are also a wide range of satellites that utilize these technologies, such as Landsat, Nimbus (Weather), Radarsat, UARS (Civil, Research and Military), etc.

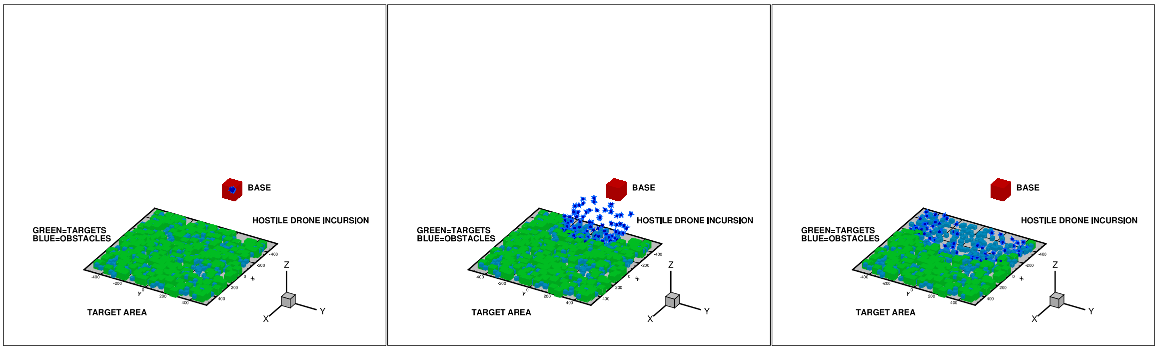

Fig. 2. Model problem (for example, a *hostile drone incursion*) consisting of targets (green) and obstacles (light blue) distributed randomly in a $(\pm 500, \pm 500, \pm 10)$ meter domain and drone-swarm members (bright blue, distributed initially in a $(\pm 10, \pm 10, \pm 10)$ meter domain centered at $(-800, 0, 200)$ meters). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2. A general and flexible drone-swarm model

In order to generally characterize the interactions between drone-swarm agent members, we adopt and modify a flexible framework found in Zohdi [41]. Throughout the first part of the analysis, the objects are assumed to be small enough to be considered (idealized) as point-masses and that the effects of their rotation with respect to their mass center is considered unimportant to their overall motion. Boldface symbols imply vectors or tensors. *A fixed Cartesian coordinate system will be used throughout this work.* The unit vectors for such a system are given by the mutually orthogonal triad of unit vectors $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$. We denote the position of a point (drone-swarm member agent) in space by the vector $\boldsymbol{r}$. In fixed Cartesian coordinates we have

$$\boldsymbol{r} = r_1\boldsymbol{e}_1 + r_2\boldsymbol{e}_2 + r_3\boldsymbol{e}_3, \tag{2.1}$$

and for the velocity we have

$$\boldsymbol{v} = \dot{\boldsymbol{r}} = \dot{r}_1\boldsymbol{e}_1 + \dot{r}_2\boldsymbol{e}_2 + \dot{r}_3\boldsymbol{e}_3, \tag{2.2}$$

and acceleration we have

$$\boldsymbol{a} = \ddot{\boldsymbol{r}} = \ddot{r}_1\boldsymbol{e}_1 + \ddot{r}_2\boldsymbol{e}_2 + \ddot{r}_3\boldsymbol{e}_3. \tag{2.3}$$

In the analysis to follow, we treat the drone-swarm members as point masses, i.e. we ignore their dimensions (Fig. 2). For each drone-swarm member (N_s in total) the equations of motion are

$$m_i\dot{\boldsymbol{v}}_i = m_i\ddot{\boldsymbol{r}}_i = \boldsymbol{\Psi}_i^{tot} = \mathcal{F}(\boldsymbol{\Psi}_i^{mt}, \boldsymbol{\Psi}_i^{mo}, \boldsymbol{\Psi}_i^{mm}) \tag{2.4}$$

where $\boldsymbol{\Psi}_i^{tot}$ represents the total forces acting on a drone-swarm member i, $\boldsymbol{\Psi}_i^{mt}$ represents the interaction between drone-swarm member i and targets to be reached, $\boldsymbol{\Psi}_i^{mo}$ represents the interaction between drone-swarm member i and obstacles and $\boldsymbol{\Psi}_i^{mm}$ represents the interaction between drone-swarm member i and other members. In order to illustrate the overall computational framework, we focus on a model problem having a sufficiently large parameter set which allows for complex dynamics. Later in the analysis, the parameters optimized to drive the system towards desired behavior via a machine-learning algorithm.

2.1. Member-target interaction

Consider member-target interaction (Fig. 3)

$$\|\boldsymbol{r}_i - \boldsymbol{T}_j\| = \left((r_{i1} - T_{j1})^2 + (r_{i2} - T_{j2})^2 + (r_{i3} - T_{j3})^2\right)^{1/2} \stackrel{\text{def}}{=} d_{ij}^{mt}, \tag{2.5}$$

where $\boldsymbol{T}_j$ is the position vector to target j and the direction to each target is

$$\boldsymbol{n}_{i\to j}^{mt} = \frac{\boldsymbol{T}_j - \boldsymbol{r}_i}{\|\boldsymbol{r}_i - \boldsymbol{T}_j\|}. \tag{2.6}$$

For each drone-swarm member (i), we compute a weighted direction to each target

$$\hat{\boldsymbol{n}}_{i\to j}^{mt} = (w_{t1}e^{-a_1 d_{ij}^{mt}} - w_{t2}e^{-a_2 d_{ij}^{mt}})\boldsymbol{n}_{i\to j}^{mt}, \tag{2.7}$$

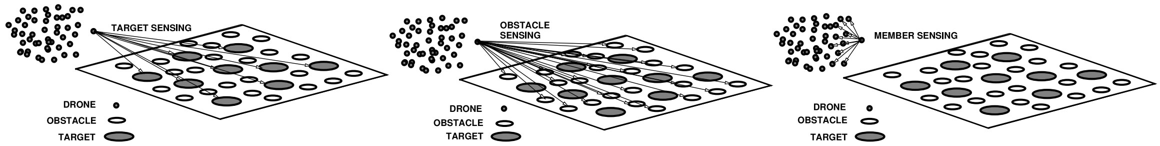

Fig. 3. Components for computing the thrust direction: (1) member-target sensing, (2) member-obstacle sensing and (c) member-member sensing.

where the w_{ti}s are weights reflecting the importance of the target, a_i are decay parameters, which is summed (and normalized later in the analysis) to give an overall direction to move towards

$$\boldsymbol{N}_i^{mt} = \sum_{j=1}^{N_t} \hat{\boldsymbol{n}}_{i\to j}^{mt}. \tag{2.8}$$

Remark. If the distance between a drone-swarm member and a target is greater that a cutoff radius

$$\|\boldsymbol{r}_i - \boldsymbol{T}_j\| > S^{rT}, \tag{2.9}$$

then this target ($\boldsymbol{T}_j$) does not contribute to the direction calculation.

2.2. Member-obstacle interaction

Now consider member-obstacle interaction (Fig. 3)

$$\|\boldsymbol{r}_i - \boldsymbol{O}_j\| = \left((r_{i1} - O_{j1})^2 + (r_{i2} - O_{j2})^2 + (r_{i2} - O_{j2})^2\right)^{1/2} \stackrel{\text{def}}{=} d_{ij}^{mo}, \tag{2.10}$$

where $\boldsymbol{O}_j$ is the position vector to obstacle j and the direction to each obstacle is

$$\boldsymbol{n}_{i\to j}^{mo} = \frac{\boldsymbol{O}_j - \boldsymbol{r}_i}{\|\boldsymbol{r}_i - \boldsymbol{O}_j\|}. \tag{2.11}$$

For each drone-swarm member (i), we compute a weighted direction to each obstacle

$$\hat{\boldsymbol{n}}_{i\to j}^{mo} = (w_{o1} e^{-b_1 d_{ij}^{mo}} - w_{o2} e^{-b_2 d_{ij}^{mo}})\boldsymbol{n}_{i\to j}^{mo}, \tag{2.12}$$

where the w_{oi}s are weights reflecting the importance of the obstacle, b_i are decay parameters, which is summed (and normalized later in the analysis) to give an overall direction to move towards

$$\boldsymbol{N}_i^{mo} = \sum_{j=1}^{N_o} \hat{\boldsymbol{n}}_{i\to j}^{mo}. \tag{2.13}$$

Remark. If the distance between a drone-swarm member and an obstacle is greater that a cutoff radius

$$\|\boldsymbol{r}_i - \boldsymbol{O}_j\| > S^{rO}, \tag{2.14}$$

then this obstacle ($\boldsymbol{O}_j$) does not contribute to the direction calculation.

2.3. Member-member interaction

Now consider member(i)-member(j) interaction (Fig. 3)

$$\|\boldsymbol{r}_i - \boldsymbol{r}_j\| = \left((r_{i1} - r_{j1})^2 + (r_{i2} - r_{j2})^2 + (r_{i3} - r_{j3})^2\right)^{1/2} \stackrel{\text{def}}{=} d_{ij}^{mm}, \tag{2.15}$$

and the direction to each drone-swarm member

$$\boldsymbol{n}_{i\to j}^{mm} = \frac{\boldsymbol{r}_j - \boldsymbol{r}_i}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|}. \tag{2.16}$$

For each drone-swarm member (i), we compute a weighted direction to each drone-swarm member

$$\hat{\boldsymbol{n}}_{i\to j}^{mm} = (w_{m1} e^{-c_1 d_{ij}^{mm}} - w_{m2} e^{-c_2 d_{ij}^{mm}})\boldsymbol{n}_{i\to j}^{mm}, \tag{2.17}$$

where the w_{mi}s are weights reflecting the importance of the members, c_i are decay parameters, which is summed (and normalized later in the analysis) to give an overall direction to move towards

$$\boldsymbol{N}_i^{mm} = \sum_{j=1}^{N_m} \hat{\boldsymbol{n}}_{i\to j}^{mm}. \tag{2.18}$$

Remark. If the distance between a drone-swarm member and another drone-swarm member is greater that a cutoff radius

$$\|\boldsymbol{r}_i - \boldsymbol{r}_j\| > S^{rr}, \tag{2.19}$$

then this drone-swarm member ($\boldsymbol{r}_j$) does not contribute to the direction calculation.

2.4. Weighted observations, summation of interactions and normalization

We now aggregate the contributions by weighting their overall importance with weights for drone-swarm member/target interaction, W_{mt}, drone-swarm member/obstacle interaction, W_{mo} and drone-swarm member/drone-swarm member interaction, W_{mm}[4]:

$$\boldsymbol{N}_i^{tot} = W_{mt}\boldsymbol{N}_i^{mt} + W_{mo}\boldsymbol{N}_i^{mo} + W_{mm}\boldsymbol{N}_i^{mm}, \tag{2.20}$$

normalize the final result

$$\boldsymbol{n}_i^* = \frac{\boldsymbol{N}_i^{tot}}{\|\boldsymbol{N}_i^{tot}\|}, \tag{2.21}$$

which provide the normal direction. The thrust forces are then constructed by multiplying the thrust force available by the drone propulsion system, F_i, by the overall normal direction

$$\boldsymbol{\Psi}_i^{thrust} = F_i\boldsymbol{n}_i^* - m_i\boldsymbol{g}, \tag{2.22}$$

where we have highlighted that an extra gravity compensation thrust component must be added, yielding

$$\boldsymbol{\Psi}_i^{tot} = \mathcal{F}(\boldsymbol{\Psi}_i^{mt}, \boldsymbol{\Psi}_i^{mo}, \boldsymbol{\Psi}_i^{mm}) = \boldsymbol{\Psi}_i^{thrust} - m_i\boldsymbol{g} + m_i\boldsymbol{g} = F_i\boldsymbol{n}_i^*, \tag{2.23}$$

which cancels gravitational forces out. In summary, the algorithm is as follows:

ALGORITHM

- STEP 1: **MEMBER-TARGET INTERACTION:**
- (a) COMPUTE THE NORM:$||\boldsymbol{r}_i - \boldsymbol{T}_j||$
- (b) COMPUTE THE WEIGHTED NORMAL: $\boldsymbol{n}_{i\to j}^{mt}$
- (c) SUM FOR MEMBER-TARGET INTERACTION: $\boldsymbol{N}_i^{mt}$
- STEP 2: **MEMBER-OBSTACLE INTERACTION:**
- (a) COMPUTE THE NORM: $||\boldsymbol{r}_i - \boldsymbol{O}_j||$
- (b) COMPUTE THE WEIGHTED NORMAL: $\boldsymbol{n}_{i\to j}^{mo}$
- (c) SUM FOR MEMBER-OBSTACLE INTERACTION: $\boldsymbol{N}_i^{mo}$
- STEP 3: **MEMBER-MEMBER INTERACTION:**
- (a) COMPUTE THE NORM: $||\boldsymbol{r}_i - \boldsymbol{r}_j||$
- (b) COMPUTE THE WEIGHTED NORMAL: $\boldsymbol{n}_{i\to j}^{mm}$
- (c) SUM FOR MEMBER-MEMBER INTERACTION: $\boldsymbol{N}_i^{mm}$
- STEP 4: **COMPUTE SUMMATION OF INTERACTIONS:** $\boldsymbol{N}_i^{tot}$
- STEP 5: **COMPUTE COMPOSITE (WEIGHTED) DIRECTION:** $\boldsymbol{n}_i^*$
- STEP 6: **COMPUTE THRUST:**$\boldsymbol{\Psi}_i^{thrust} = F_i\boldsymbol{n}_i^* - m_i\boldsymbol{g}$ (extra gravity compensation)

3. Drone-swarm actuation

To actuate the drone-swarm movement, we numerically integrate the equations of motion:

$$m_i\dot{\boldsymbol{v}}_i = \boldsymbol{\Psi}_i^{tot} \tag{3.1}$$

yielding

$$\boldsymbol{v}_i(t + \Delta t) = \boldsymbol{v}_i(t) + \frac{\Delta t}{m_i}\boldsymbol{\Psi}_i^{tot}(t) \tag{3.2}$$

[4] The parameters in the model will be optimized shortly.

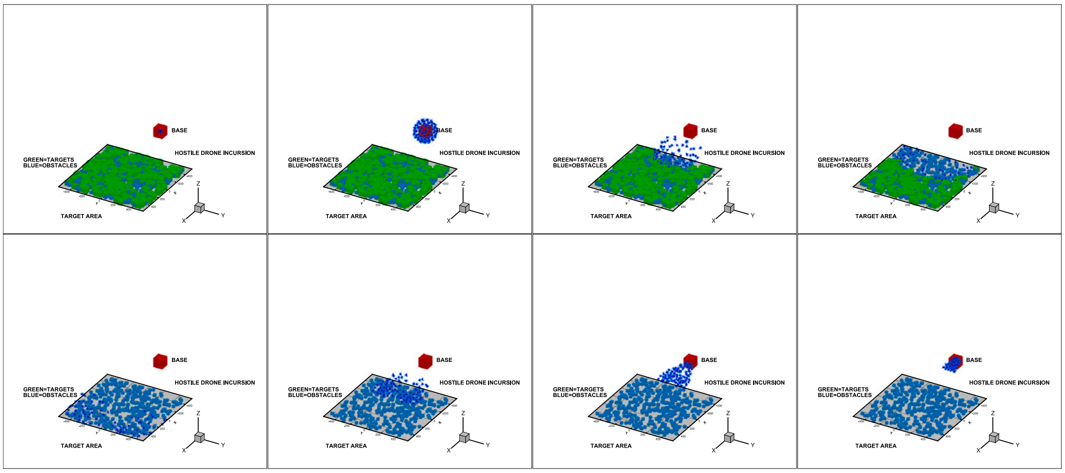

Fig. 4. The propagation of a drone-swarm (for example, a *hostile drone incursion*). In this scenario, they sweep through the targets. The eight frames (left to right) show the motion (equally spaced over the simulation time $T = 35s$).

and

$$\boldsymbol{r}_i(t + \Delta t) = \boldsymbol{r}_i(t) + \Delta t \boldsymbol{v}_i(t). \tag{3.3}$$

Note that if the maximum velocity is exceeded

$$\|\boldsymbol{v}_i(t + \Delta t)\| > v_{i,max}, \tag{3.4}$$

then we define the predicted velocity $\boldsymbol{v}_i^{pred}(t + \Delta t) = \boldsymbol{v}_i(t + \Delta t)$ and the rescaled/corrected velocity

$$\boldsymbol{v}_i^{corr}(t + \Delta t) = v_{i,max} \frac{\boldsymbol{v}_i^{pred}(t + \Delta t)}{\|\boldsymbol{v}_i^{pred}(t + \Delta t)\|}, \tag{3.5}$$

with $\boldsymbol{v}_i(t + \Delta t) = \boldsymbol{v}_i^{pred}(t + \Delta t)$. We then determine if any targets have been reached by checking the distance between drone-swarm-members and targets and comparing it to a sensing tolerance

$$\|\boldsymbol{r}_i - \boldsymbol{T}_j\| \leq Tol^{rT} \tag{3.6}$$

For any $\boldsymbol{T}_j$, if any drone-swarm member has satisfied the criteria, the algorithm takes $\boldsymbol{T}_j$ out of the system for the next time step so that no drone-swarm member wastes resources by attempting to reach $\boldsymbol{T}_j$. *Similarly, if the drones come too close to the obstacles,* $\|\boldsymbol{r}_i - \boldsymbol{O}_j\| \leq Tol^{rO}$*, then* $\boldsymbol{r}_i$ *is immobilized. This stops the* ith *drone-swarm member from contributing further to the process. Furthermore, if drones come too close to one another,* $\|\boldsymbol{r}_i - \boldsymbol{r}_j\| \leq Tol^{rr}$*, then* $\boldsymbol{r}_i$ *and* $\boldsymbol{r}_j$ *are immobilized.* The entire process is then repeated for the next time step. Fig. 4 illustrates the propagation of a drone-swarm, where they sweep through the targets. Eight frames (left to right) illustrate the motion (equally spaced over the simulation time $T = 35s$). This will be discussed in detail shortly.

4. Designing drone-swarm behavior

4.1. Multi-timeframe objective function and machine-learning

There are many parameters in the system, warranting the use of a Machine-Learning Algorithm. Here we follow Zohdi [41], [61–71],[72], [73], [74] in order to optimize behavior by minimizing a cost function. For example, let us consider minimizing the following cost function over the event time period of interest, summing over all of the target states (mapped (inactive) or unmapped (active)) in each time frame and summing up the entire difference (with normalization)

$$\Pi^{total}(\boldsymbol{\Lambda}) = \frac{\sum_{f=1}^{N_f} \{\sum_{i=1}^{N_t} |T_i^a(t_f) - T_i^{a*}(t_f)|\}}{\sum_{f=1}^{N_f} \{\sum_{i=1}^{N_t} T_i^{a,*}(t_f)\}} \tag{4.1}$$

where

- For the targets are:
- $T_i^a(t = t_f) = 1$ if unmapped at time (active) at time $t = t_f$
- $T_i^a(t = t_f) = 0$ if mapped (inactive) at time $t = t_f$

- N_f is the number of time frames,
- N_t is the number of targets,
- Ω_f is spatio-temporal frame capturing states of the targets (such as in Figs. 2 and 4),
- $\boldsymbol{\Lambda} = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_{N_d}\}$ is a design vector of key system (N_d) parameters.

The objective is to drive the system to the parameters generating the best case scenario, via cost function minimization. The design vector of system parameters is:

$$\begin{aligned}\boldsymbol{\Lambda} &= \{\Lambda_1, \Lambda_2, \ldots, \Lambda_N\} \\ &= \{W_{mt}, W_{mo}, W_{mm}, w_{t1}, w_{t2}, w_{o1}, w_{o2}, w_{m1}, w_{m2}, a_1, a_2, b_1, b_2, c_1, c_2, S^{rT}, S^{rO}, S^{rr}\}.\end{aligned} \tag{4.2}$$

Cost functions associated with optimization of complex behavior are oftentimes nonconvex in design parameter space and often nonsmooth, as is the case for the system of interest. Their minimization is usually difficult with direct application of gradient methods. This motivates nonderivative search methods, for example those found in Machine Learning Algorithms (MLA's). One of the most basic subset of MLA's are so-called Genetic Algorithms (GA's). Typically, one will use a GA first in order to isolate multiple local minima, and then use a gradient based algorithm in these locally convex regions or reset the GA to concentrate its search over these more constrained regions. GA's are typically the simplest scheme to start the analysis, and one can, of course, use more sophisticated methods if warranted. For a review of GAs, see the pioneering work of John Holland ([75], [76]), as well as Goldberg [77], Davis [78], Onwubiko [79] and Goldberg and Deb [80]. Here we follow Zohdi [41], [61–71],[72], [73], [74] in order to minimize Eq. (4.1), which we will refer to as the "cost/error function".

4.2. Algorithm

The machine-learning GA approach is extremely well-suited for nonconvex, nonsmooth, multicomponent, multistage systems (see Fig. 5) and, broadly speaking, involves the following essential concepts:

1. **POPULATION GENERATION:** Generate a parameter population of genetic strings: $\boldsymbol{\Lambda}^i$.
2. **PERFORMANCE EVALUATION:** Compute performance of each genetic string: $\Pi(\boldsymbol{\Lambda}^i)$.
3. **RANK STRINGS:** Rank them $\boldsymbol{\Lambda}^i$, $i = 1, \ldots, S$.
4. **MATING PROCESS:** Mate pairs/produce offspring.
5. **GENE ELIMINATION:** Eliminate poorly performing genetic strings.
6. **POPULATION REGENERATION:** Repeat process with updated gene pool and new *random* genetic strings.
7. **SOLUTION POST-PROCESSING:** Employ gradient-based methods afterwards in local "valleys", *if smooth enough*.

4.2.1. Algorithmic specifics

Following Zohdi [41], [61–71],[72], [73], [74] the algorithm is as follows:

- **STEP 1:** Randomly generate a population of S starting genetic strings, $\boldsymbol{\Lambda}^i$, (i=1,2, 3, …, S):

$$\boldsymbol{\Lambda}^i \stackrel{\text{def}}{=} \left\{\begin{array}{c}\Lambda_1^i \\ \Lambda_2^i \\ \Lambda_3^i \\ \ldots \\ \Lambda_N^i\end{array}\right\} \tag{4.3}$$

- **STEP 2:** Compute fitness of each string $\Pi(\boldsymbol{\Lambda}^i)$, (i=1, …, S)
- **STEP 3:** Rank genetic strings: $\boldsymbol{\Lambda}^i$, (i=1, …, S)
- **STEP 4:** Mate nearest pairs and produce two offspring, (i=1, …, S):

$$\boldsymbol{\lambda}^i \stackrel{\text{def}}{=} \boldsymbol{\Phi}\circ\boldsymbol{\Lambda}^i + (\mathbf{1}-\boldsymbol{\Phi})\circ\boldsymbol{\Lambda}^{i+1} \stackrel{\text{def}}{=} \left\{\begin{array}{c}\phi_1\Lambda_1^i \\ \phi_2\Lambda_2^i \\ \phi_3\Lambda_3^i \\ \ldots \\ \phi_N \Lambda i_N\end{array}\right\} + \left\{\begin{array}{c}(1-\phi_1)\Lambda_1^{i+1} \\ (1-\phi_2)\Lambda_2^{i+1} \\ (1-\phi_3)\Lambda_3^{i+1} \\ \ldots \\ (1-\phi_N)\Lambda_N^{i+1}\end{array}\right\} \tag{4.4}$$

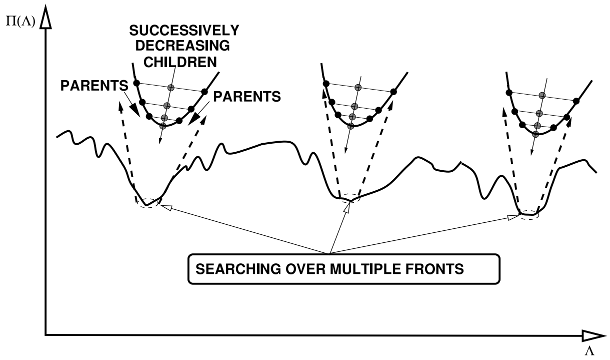

Fig. 5. The basic action of a genetic algorithm: multifrontal search over a nonconvex and nonsmooth design space (Zohdi [41], [61–71],[72], [73], [74]).

and

$$\lambda^{i+1} \stackrel{\text{def}}{=} \boldsymbol{\Gamma}\circ\boldsymbol{\Lambda}^{i} + (\mathbf{1}-\boldsymbol{\Gamma})\circ\boldsymbol{\Lambda}^{i+1} \stackrel{\text{def}}{=} \left\{\begin{array}{c} \gamma_1 \Lambda_1^i \\ \gamma_2 \Lambda_2^i \\ \gamma_3 \Lambda_3^i \\ \dots \\ \gamma_N \Lambda_N \end{array}\right\} + \left\{\begin{array}{c} (1-\gamma_1)\Lambda_1^{i+1} \\ (1-\gamma_2)\Lambda_2^{i+1} \\ (1-\gamma_3)\Lambda_3^{i+1} \\ \dots \\ (1-\gamma_N)\Lambda_N^{i+1} \end{array}\right\} \tag{4.5}$$

where for this operation, the ϕ_i and γ_i are random numbers, such that $0 \le \phi_i \le 1$, $0 \le \gamma_i \le 1$, which are different for each component of each genetic string.

- **STEP 5:** Keep only the top K parents and their K offspring.
- **STEP 6:** Repeat STEPS 1–6 with top gene pool (K offspring and K parents), plus M new, randomly generated, strings.
- **IMPORTANT OPTION:** Rescale and restart the search around best performing parameter set every few generations.

5. Numerical experiments

As a model problem, consider the following algorithm:

1. Initialize the locations of the targets: $\boldsymbol{T}_i = (T_x, T_y, T_z)_i$, i=1, 2, … N_T=targets.
2. Initialize the locations of the obstacles: $\boldsymbol{O}_i = (O_x, O_y, O_z)_i$, i=1, 2, … N_O=obstacles.
3. Initialize the locations of the drone-swarm-members: $\boldsymbol{r}_i = (r_x, r_y, r_z)_i$, i=1, 2, … N_s=drone-swarm members.
4. For each drone-swarm member (i), determine the distance and directed normal to each target, obstacle and other drone-swarm members.
5. For each drone-swarm member (i), determine interaction functions $\boldsymbol{N}_i^{mt}$, $\boldsymbol{N}_i^{mo}$, $\boldsymbol{N}_i^{mt}$ and $\boldsymbol{n}_i^*$.
6. For each drone-swarm member (i), determine force acting upon it, $\boldsymbol{\Psi}_i^{tot} = F_i \boldsymbol{n}_i^*$.
7. For each drone-swarm member (i), integrate the equations of motion (checking constraints) to produce $\boldsymbol{v}_i(t+\Delta t)$ and $\boldsymbol{r}_i(t+\Delta t)$.
8. Determine if any targets have been reached by checking the distance between drone-swarm members and targets
$$\|\boldsymbol{r}_i - \boldsymbol{T}_j\| \le Tol^{rT}. \tag{5.1}$$
For any $\boldsymbol{T}_j$, if any drone-swarm member has satisfied the this criteria, then take $\boldsymbol{T}_j$ out of the system for the next time step so that no drone-swarm member wastes resources by attempting to reach $\boldsymbol{T}_j$.
9. If $\|\boldsymbol{r}_i - \boldsymbol{O}_j\| \le Tol^{rO}$, then $\boldsymbol{r}_i$ is *immobilized*. Furthermore, if $\|\boldsymbol{r}_i - \boldsymbol{r}_j\| \le Tol^{rr}$, then $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ are *immobilized*.
10. The entire process is then repeated for the next time step.

5.1. Ground truth

To generate a *ground truth response*, a set of system design parameters ($\boldsymbol{\Lambda}$) were chosen at random within the design interval space. A simulation was run for the system with that design parameter setting and the response of the system was recorded. This was the ground truth response used in the cost function. As a preliminary example, we considered the following parameters:

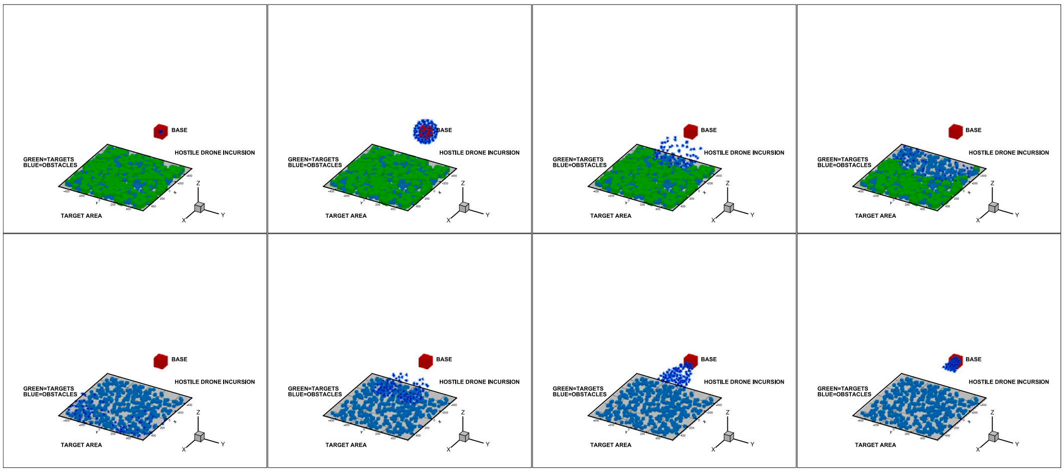

Fig. 6. The sequences of the model problem (for example, a *hostile drone incursion*) show initially green (unreached) targets, which are eliminated when they are mapped. The light blue blocks indicate the obstacles and the bright blue objects are the drones. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- Mass = 10 kg,
- 100 drone-swarm members,
- 200 targets,
- 200 obstacles,
- $T = 35$ seconds,
- $\Delta t = 0.001$ seconds,
- Initial drone-swarm velocity, $\boldsymbol{v}_i(t=0) = \mathbf{0}$ m/s,
- Domain to be reached consists of targets and obstacles distributed randomly in a $(\pm 500, \pm 500, \pm 10)$ meter domain, with drone-swarm members distributed initially in a $(\pm 10, \pm 10, \pm 10)$ meter domain centered at $(-800, 0, 200)$ meters.
- Thrust force available by the drone propulsion system, $F_i = 10^6$ Nt,
- Maximum velocity drone-swarm member $v_{max} = 100$ m/s.

The trial "design" vector of system parameter inputs

$$\begin{aligned} \boldsymbol{\Lambda}^{trial} &\stackrel{\text{def}}{=} \{\Lambda_1 \ldots \Lambda_N\} \\ &= \{W_{mt}, W_{mo}, W_{mm}, w_{t1}, w_{t2}, w_{o1}, w_{o2}, w_{m1}, w_{m2}, a_1, a_2, b_1, b_2, c_1, c_2, S^{rT}, S^{rO}, S^{rr}\}. \end{aligned} \tag{5.2}$$

was given by a randomly generated vector

$$\begin{aligned} \boldsymbol{\Lambda}^{trial} \stackrel{\text{def}}{=} \{&0.336211, 0.032175, 0.239331, 0.580259, 0.441191, 0.063997, 0.389643, 1.644445, 0.553937, \\ &0.007149, 0.565078, 0.592352, 0.634431, 0.930645, 0.258904, 8.207462, 6.525159, 8.556350\}, \end{aligned}$$

in the following intervals:

- Overall weights: $0 \le W_{mt}, W_{mo}, W_{mm} \le 10$,
- Target weights: $0 \le w_{t1}, w_{t2} \le 1$,
- Obstacle weights: $0 \le w_{o1}, w_{o2} \le 1$,
- Member weights: $0 \le w_{m1}, w_{m2} \le 1$,
- Decay coefficients: $0 \le a_1, a_2 \le 1$, $0 \le b_1, b_2 \le 1$, $0 \le c_1, c_2 \le 1$ and
- The sensing distances: $0 \le S^{rT} \le 10, 0 \le S^{rO} \le 10, 0 \le S^{rr} \le 10$.

We allowed a long enough time to reach the whole domain (35 s). The results are shown in Figs. 6 and 7.

The sequences of the model problem show initially green (unreached) targets, which are eliminated when they are mapped. The blue blocks indicate the obstacles. The algorithm is quite adept in picking up missed targets by successive sweeps. We note that as the targets get reached, they are dropped from the system, and the drone-swarm members naturally aggregate to the targets that are remaining. We also note that we did not put an upper or lower bound on the altitude that the drones could attain in this model problem, although that is relatively easy to enforce.

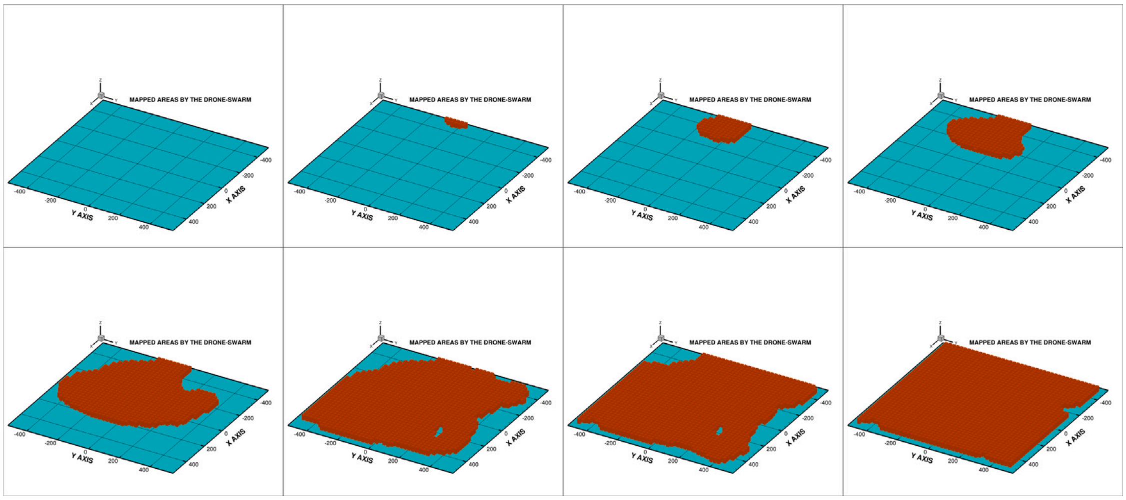

Fig. 7. The sequences of the model problem showing the mapped areas as they are registered.

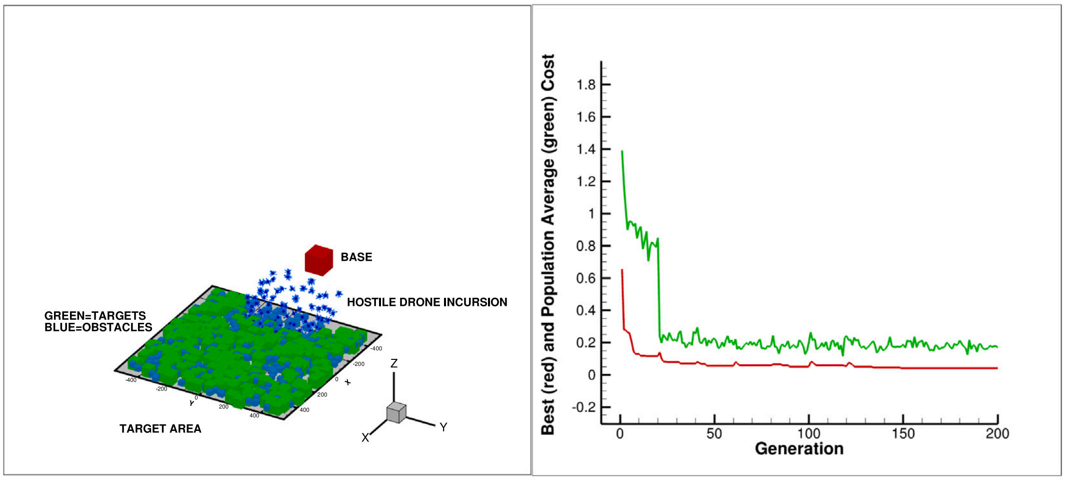

Fig. 8. Machine learning output (for example, a *hostile drone incursion*), generation after generation-the reduction of the cost function (Π) for the 18 parameter set is shown. On the right, the best performing gene (**red**) is shown as a function of successive generations, in addition to the average performance of the population of genes (**green**). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.2. Numerical examples

We applied the Machine Learning Algorithm and with $T = 35$ seconds and 100 sampling frames. Fig. 8 illustrates the best performing gene (design parameter set) in red, as a function of successive generations, as well as the average performance of the population of genes in green. The design parameters were optimized over the following intervals previously mentioned. We used the following MLA settings:

- Population size per generation: 24,
- Number of parents to keep in each generation: 4,
- Number of children created in each generation: 4,
- Number of completely new genes created in each generation: 12,
- Number of generations for readaptation around a new search interval:20,
- Number of generations: 200.

The algorithm was automatically reset every *20 generations*. The entire 200 generation simulation, with 24 genes per evaluation (4800 total designs) took on the order of 4 min on a laptop, *making it ideal as a design tool.* Fig. 8 (average of all genes performance and top gene performance) and the resulting design vector:

$$\boldsymbol{\Lambda}^{i=200} \stackrel{\text{def}}{=} \{0.336352, 0.032175, 0.239331, 0.580263, 0.441188, 0.063997, 0.389643, 1.644446, 0.553946, 0.007149, 0.565087, 0.592353, 0.634552, 0.930888, 0.258904, 8.207454, 6.525155, 8.556332\}$$

and

$$\Pi^{i=200}(\boldsymbol{\Lambda}) = 0.04221377. \tag{5.3}$$

We note that the parameters that the method found vary only after the 5th decimal place with those of the optimal (trial) vector. We note that, for a given set of parameters, a complete simulation takes on the order of 0.1 s, thus over 36,000 parameter sets can be evaluated in an hour, *without even exploiting the inherent parallelism of the MLA.*

5.3. Cost-function enhancement

5.3.1. Energy penalties

Energy consumption can be an important consideration in drone-swarm system design. Accordingly, one can include *energy penalties* for excessive power usage, by integrating the power for drone to determine the total energy usage:

$$E_i^{tot} = \int_0^T |\boldsymbol{\Psi}_i^{thrust} \cdot \boldsymbol{v}_i|\, dt \tag{5.4}$$

and comparing it to the available battery energy, E_i^{bat}. If the energy expenditure is greater (*overage*) than the available battery power, we penalize the system performance metric in the following way:

- Penalty active:

$$E_i^{tot} > E_i^{bat} \Rightarrow \Pi_i^{e,pen} = \frac{(E_i^{tot} - E_i^{bat})}{E_i^{bat}}. \tag{5.5}$$

- Penalty inactive:

$$E_i^{tot} \le E_i^{bat} \Rightarrow \Pi_i^{e,pen} = 0. \tag{5.6}$$

- Summation over agents:

$$\Pi^{e,pen,tot} = \sum_{i=1}^{N} \Pi_i^{e,pen}. \tag{5.7}$$

5.3.2. Acoustical penalties

We remark that additional detectable multiphysics quantities associated with the drone-swarm's behavior is relatively straightforward, such as the acoustical emission, which could be important for stealth applications. This can be incorporated by computing the acoustical emission that is detected at the obstacles (receivers):

$$A_i = \sum_{j=1}^{N_O} \frac{I_j}{4\pi \|\boldsymbol{O}_i - \boldsymbol{r}_j\|^2}, \tag{5.8}$$

where I_j is the source from each drone-swarm member and A_i is the acoustical signal picked up by each obstacle station, where sound intensity, denoted I is measured in $\mathrm{W/m^2}$ (see Fig. 9). We note that because of the $(1/r)$ behavior associated with acoustical power, such data is much more sensitive than positional data, but could be added to the cost. Specifically, one could include an *acoustical penalty* such as by first calculating $A_i(t)$ at each location and then calculating the *overage* penalty

- Penalty active:

$$A_i^{tot} > A_i^{max} \Rightarrow \boldsymbol{\Phi}_i(t) = \frac{(A_i - A_i^{max})}{A_i^{max}} \tag{5.9}$$

- Penalty inactive:

$$A_i^{tot} \le A_i^{max} \Rightarrow \boldsymbol{\Phi}_i(t) = 0 \tag{5.10}$$

  and

$$\Pi^{a,pen} = \int_0^T \boldsymbol{\Phi}_i(t)\, dt \tag{5.11}$$

- Summation over agents:

$$\Pi^{a,pen,tot} = \sum_{i=1}^{N} \Pi_i^{a,pen}. \tag{5.12}$$

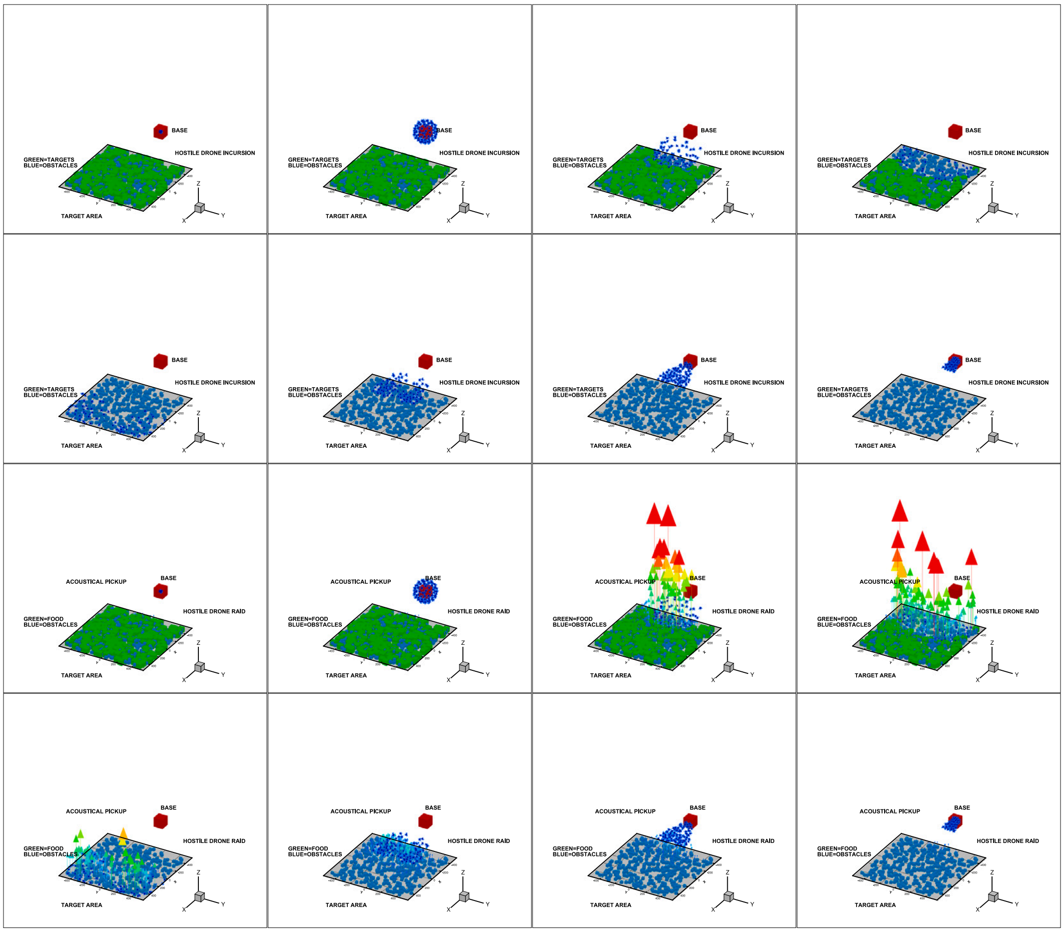

Fig. 9. The propagation of the drone-swarm (for example, a *hostile drone incursion*) with acoustical signals at the obstacles utilizing Eq. (5.8). In this scenario, they sweep through the targets and the acoustics is picked up by the obstacles. The first eight frames (left to right) show the motion without acoustics. Second eight frames (left to right) show the motion with acoustics.

5.4. Penalized cost function

The total penalized cost function is therefore (building on Eq. (4.1))

$$\Pi^{total}(\boldsymbol{\Lambda}) = w_t \frac{\sum_{f=1}^{N_f}\{\sum_{i=1}^{N_t}|T_i^a(t_f) - T_i^{a*}(t_f)|\}}{\sum_{f=1}^{N_f}\{\sum_{i=1}^{N_t} T_i^{a,*}(t_f)\}} + w_e \Pi^{e,pen,tot} + w_a \Pi^{a,pen,tot} \tag{5.13}$$

where w_t is the weight for the target metric, w_e is the weight for the energy penalty and w_a is the weight for the acoustic penalty. If one wishes to have more detailed descriptions beyond a point mass model (for example, for a quadcopter), one must augment the balance of linear momentum ($\boldsymbol{G}_{cm,i}$)

$$\dot{\boldsymbol{G}}_{cm,i} = m_i \ddot{\boldsymbol{r}}_{cm,i} = \boldsymbol{\Psi}_i^{tot}, \tag{5.14}$$

with a balance of angular momentum ($\boldsymbol{H}_{cm,i}$), which governs the rotation of the structure given by

$$\dot{\boldsymbol{H}}_{cm,i} = \frac{d(\overline{\boldsymbol{\mathcal{I}}}_i \cdot \boldsymbol{\omega}_i)}{dt} = \boldsymbol{M}_{cm,i}^{tot}, \tag{5.15}$$

where $\boldsymbol{M}_{cm,i}^{tot}$ is the total external moment about the center of mass, $\overline{\boldsymbol{\mathcal{I}}}_i$ is the mass moment of inertia and $\boldsymbol{\omega}_i$ is the angular velocity. For detailed modeling of the dynamics and control of drones we refer the reader to Mueller and D'Andrea [81], [82], Mueller et al. [83], Hehn et al. [84], Houska et al. [85], Taglibue etc al [86] and Holda et al. [87] and Zohdi [88]. In many applications, the computed positions, velocities and accelerations of the members of a drone-swarm for example people or vehicles, must be translated into realizable movement. Furthermore, the communication latency and information exchange poses a significant technological hurdle. In practice, further sophistication, i.e. constraints on movement and communication, must be embedded into the computational

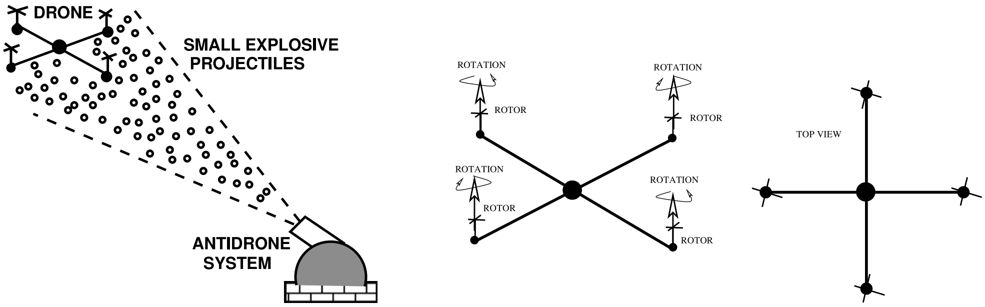

Fig. 10. Left: A drone under attack. Right: A (skeletal) drone schematic with propellor thrusts. The entire body is governed by rigid body kinematics with two primary variables: the angular velocity of the body $\boldsymbol{\omega}$ and the velocity of the mass center $\boldsymbol{v}_c$. The velocities and positions of all other points on the body can be determined by rigid body kinematics. Generally, two rotors spin one way, while two of them spin the other way so that the main body of the quadcopter does not rapidly spin as it flies (conservation of angular momentum.).

model for the application at hand. However, the fundamental computational philosophy and modeling strategy should remain relatively unchanged. One could reformulate the cost function to minimize energy usage, incorporating the range, performance and structure of a drone, and its dynamic performance in a complex environment, and potentially hostile environment. In particular, recent interest in drones has grown dramatically, driven primarily by military applications. Now, nearly every country in the world has tactical drones under development. Generally speaking, most military drones that are intended for long distance operations are fixed wing aircraft, while for precise "stop and go" operations in complex, potentially hostile, environments they are performed with quadcopters. Accordingly, as an example, in the upcoming analysis, we focus on the performance of quadcopter drones, which comprises the second component of the overall analysis.

6. Structural design and performance of drones

To assess the performance of quadcopter drones, we develop a method for rapid structural design, deployment and dynamic performance evaluation of tactical quadcopter drones under attack, specifically by being subjected to series of launched explosions. A Discrete Element Method (DEM) is developed to rapidly design a quadcopter of any complex shape, attach motors and then to subject it to a hostile environment, in order to ascertain its deployed performance in the field. The method also allows one to potentially describe damage to the quadcopter drone, its loss of functionality (thrust), etc. Furthermore, the use of DEM can also allow for fragmentation of the quadcopter and can also ascertain the resulting debris field (Zohdi [88]) (see Fig. 10).

6.1. Generation of a drone chasis

To generate the quadcopter body, we insert particles within an envelope/grid intersection (Fig. 11). For example, a convenient, easy to parametrize envelope is given by sweeping through a rectangular parallelepiped of $(\pm R_1, \pm R_2, \pm R_3)$ and checking the intersection of the hull envelope equation, for example given by a generalized ellipsoidal equation (Zohdi [88])

$$\frac{|x_1 - x_{1o}|^{p_1}}{R_1} + \frac{|x_2 - x_{2o}|^{p_1}}{R_2} + \frac{|x_3 - x_{3o}|^{p_3}}{R_3} \leq 1, \tag{6.1}$$

where (x_1, x_2, x_3) are the coordinates of the DEM particles, (x_1, x_2, x_3) are the coordinates of center of the chassis, (R_1, R_2, R_3) are the generalized radii and (p_1, p_2, p_3) are exponents of the generalized ellipsoid, with a box of discrete element "subbox" positions (Fig. 11). Where there is an intersection, a particle is placed in the subbox. The particles are initially rigidly bonded together, but may become dislodged due to external forces (discussed later in the presentation). For exponent values of (p_1, p_2, p_3) equal to two, we generate a familiar ellipsoid, for values less than one we generate involute (nonconvex shapes, Fig. 11), and for exponent values of (p_1, p_2, p_3) greater than two, we generate a box-like shapes (see Fig. 12).

6.2. Group dynamics of a rigidly bound collection of drone particles/elements

In order to make the analysis general, we consider rigid clusters of DEM particles. Later we will tailor the cluster to specific drone designs. We consider the DEM cluster to be already formed, with particles rigidly bonded together. Later, we will allow particles to become dislodged from the cluster. Consider a collection of rigidly-bonded particles, $i = 1, 2, \ldots, N_c$, in a cluster. The individual particle dynamics are described by (which leads to a coupled system)

$$m_i \ddot{\boldsymbol{r}}_i = m_i \dot{\boldsymbol{v}}_i = \underbrace{\boldsymbol{\Psi}_i^{tot}}_{\text{total forces}} = \underbrace{\boldsymbol{\Psi}_i^{int}}_{\text{internal}} + \underbrace{\boldsymbol{\Psi}_i^{ext}}_{\text{external}}, \tag{6.2}$$

where m_i is the mass of the ith particle, $\boldsymbol{r}_i$ is the position vector, $\boldsymbol{v}_i$ is the particle velocity, $\boldsymbol{\Psi}_i^{ext}$ is an external force field and $\boldsymbol{\Psi}_i^{int}$ is the sum of the internal (equal in magnitude and opposite in direction) forces acting on the ith particle, due to other particles

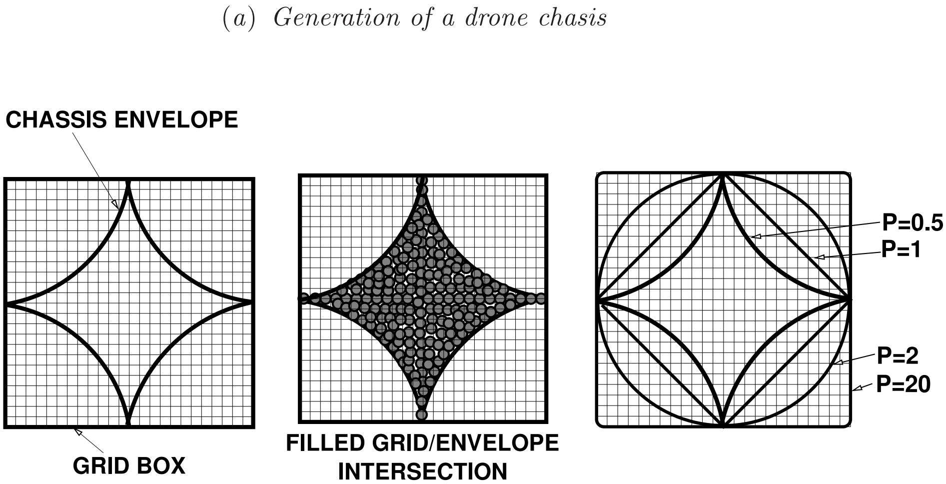

Fig. 11. Generating a drone chasis with discrete elements (based on Zohdi [88]). This achieved by sweeping through a rectangular parallelepiped of $(\pm R_1, \pm R_2, \pm R_3)$ and checking the intersection of the hull envelope equation above with a box of discrete element position subboxes (left). Where there is an intersection, and particle is placed in the subbox (middle). A generalized ellipsoidal equation (Eq. (6.1)) is used where for exponent values of (p_1, p_2, p_3) equal to two, we generate a familiar ellipsoid, for values less than one we generate involute (nonconvex shapes), and for exponent values of (p_1, p_2, p_3) greater than two, we generate a box-like shapes (right).

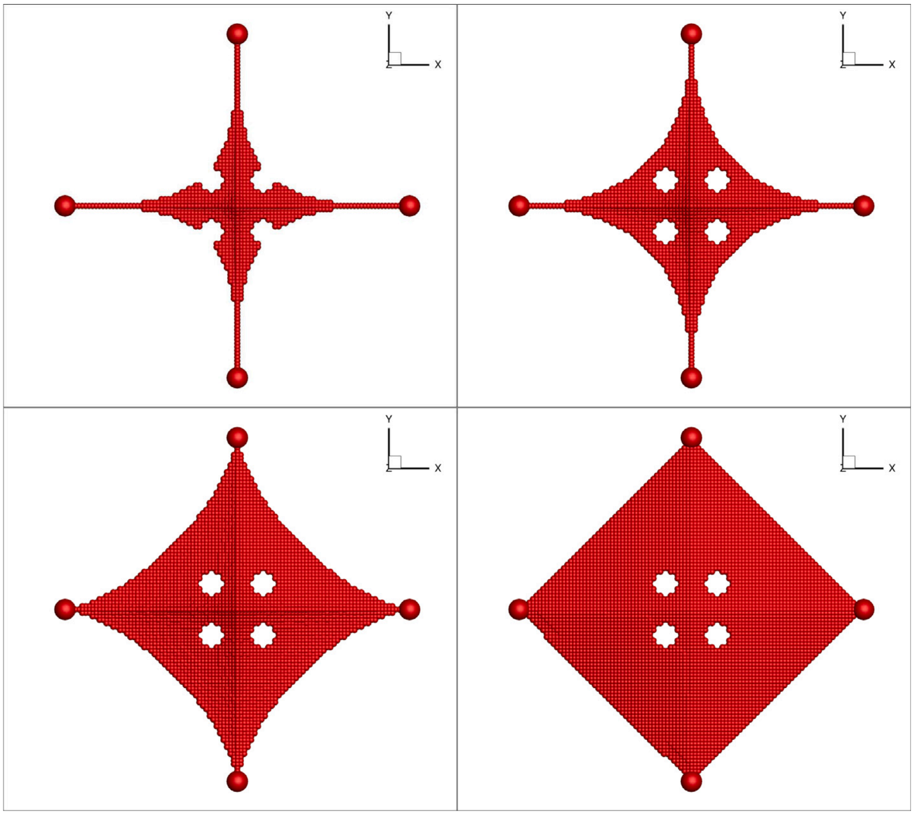

Fig. 12. Various chasis envelopes: (a) $(p_1, p_2, p_3) = (0.4, 0.4, 0.4)$, (b) $(p_1, p_2, p_3) = (0.5, 0.5, 0.5)$, (c) $(p_1, p_2, p_3) = (0.7, 0.7, 0.7)$, (d) $(p_1, p_2, p_3) = (1.0, 1.0, 1.0)$ with "lightweighting" holes punched out for weight reduction. Additional lightweighting of the structure is extremely easy to analyze by simply deleting discrete elements within the Discrete Element Method framework.

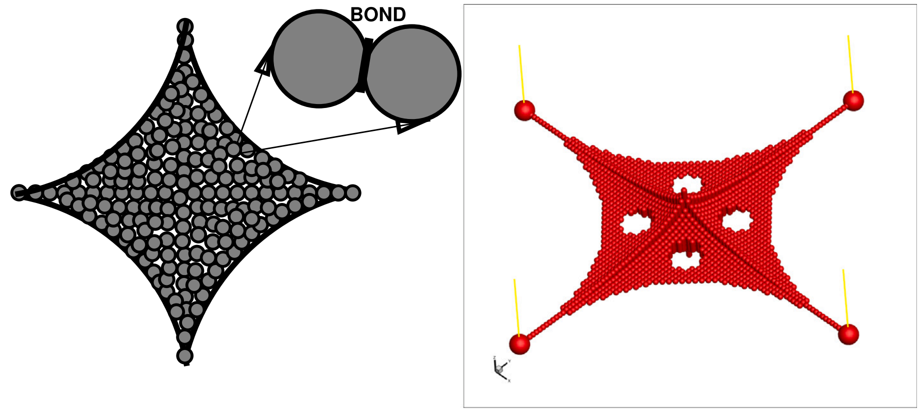

Fig. 13. For $(p_1, p_2, p_3) = (0.5, 0.5, 0.5)$. A DEM generated drone frame. Left: A zoom on the *locations* of the DEM particles. The particles are bound by the mathematical dynamics-constraints to move collectively as a rigid body (group translation and rotation), unless a particle is dislodged by excessive force. If dislodged, the fragment moves according to its own dynamics.

in the system ("internal" particle-to-particle bonding forces, contact forces etc. When we consider a collection of particles that are bound together as a rigid body, because internal forces between particles within in the system are opposite in direction and equal in magnitude, the specific character of the internal particle-to-particle bonding forces is not relevant to the overall system dynamics,

$$\sum_{i=1}^{N_c} \left(\boldsymbol{\Psi}_i^{ext} + \boldsymbol{\Psi}_i^{int}\right) = \sum_{i=1}^{N_c} \boldsymbol{\Psi}_i^{ext} + \underbrace{\sum_{i=1}^{N_c} \boldsymbol{\Psi}_i^{int}}_{=\mathbf{0}} = \sum_{i=1}^{N_c} \boldsymbol{\Psi}_i^{ext} \stackrel{\text{def}}{=} \boldsymbol{\Psi}^{EXT}, \tag{6.3}$$

where $\boldsymbol{\Psi}^{EXT}$ is the overall external force acting on the cluster and N_c are the number of particles in the DEM cluster. The position vector of the center of mass of the system is given by

$$\boldsymbol{r}_{cm} \stackrel{\text{def}}{=} \frac{\sum_{i=1}^{N_c} m_i \boldsymbol{r}_i}{\sum_{i=1}^{N_c} m_i} = \frac{1}{\mathcal{M}} \sum_{i=1}^{N_c} m_i \boldsymbol{r}_i, \tag{6.4}$$

where $\mathcal{M}$ is the total system mass. A decomposition of the position vector for particle i, of the form $\boldsymbol{r}_i = \boldsymbol{r}_{cm} + \boldsymbol{r}_{cm\to i}$, allows the linear momentum of the system of particles ($\boldsymbol{G}$) to be written as

$$\sum_{i=1}^{N_c} \underbrace{m_i \dot{\boldsymbol{r}}_i}_{\boldsymbol{G}_i} = \sum_{i=1}^{N_c} m_i (\dot{\boldsymbol{r}}_{cm} + \dot{\boldsymbol{r}}_{cm\to i}) = \sum_{i=1}^{N_c} m_i \dot{\boldsymbol{r}}_{cm} = \dot{\boldsymbol{r}}_{cm} \sum_{i=1}^{N_c} m_i = \mathcal{M} \dot{\boldsymbol{r}}_{cm} \stackrel{\text{def}}{=} \boldsymbol{G}_{cm}, \tag{6.5}$$

since $\sum_{i=1}^{N_c} m_i \dot{\boldsymbol{r}}_{cm\to i} = \mathbf{0}$. Furthermore, $\dot{\boldsymbol{G}}_{cm} = \mathcal{M} \ddot{\boldsymbol{r}}_{cm}$, thus

$$\dot{\boldsymbol{G}}_{cm} = \mathcal{M} \ddot{\boldsymbol{r}}_{cm} = \sum_{i=1}^{N_c} \boldsymbol{\psi}_i^{ext} \stackrel{\text{def}}{=} \boldsymbol{\Psi}^{EXT}. \tag{6.6}$$

The angular momentum relative to the center of mass can be written as (utilizing $\dot{\boldsymbol{r}}_i = \boldsymbol{v}_i = \boldsymbol{v}_{cm} + \boldsymbol{v}_{cm\to i}$)

$$\sum_{i=1}^{N_c} \boldsymbol{H}_{cm\to i} = \sum_{i=1}^{N_c} (\boldsymbol{r}_{cm\to i} \times m_i \boldsymbol{v}_{cm\to i}) = \sum_{i=1}^{N_c} (\boldsymbol{r}_{cm\to i} \times m_i (\boldsymbol{v}_i - \boldsymbol{v}_{cm})) \tag{6.7}$$

$$= \sum_{i=1}^{N_c} (m_i \boldsymbol{r}_{cm\to i} \times \boldsymbol{v}_i) - \left(\underbrace{\sum_{i=1}^{N_c} m_i \boldsymbol{r}_{cm\to i}}_{=0} \right) \times \boldsymbol{v}_{cm} = \boldsymbol{H}_{cm}, \tag{6.8}$$

for a rigid body. Since $\boldsymbol{v}_{cm\to i} = \boldsymbol{\omega} \times \boldsymbol{r}_{cm\to i}$

$$\boldsymbol{H}_{cm} = \sum_{i=1}^{N_c} \boldsymbol{H}_{cm\to i} = \sum_{i=1}^{N_c} m_i (\boldsymbol{r}_{cm\to i} \times \boldsymbol{v}_{cm\to i}) = \sum_{i=1}^{N_c} m_i (\boldsymbol{r}_{cm\to i} \times (\boldsymbol{\omega} \times \boldsymbol{r}_{cm\to i})). \tag{6.9}$$

Decomposing the relative position vector into its components

$$\boldsymbol{r}_{cm\to i} = \boldsymbol{r}_i - \boldsymbol{r}_{cm} = \hat{x}_{i1} \boldsymbol{e}_1 + \hat{x}_{i2} \boldsymbol{e}_2 + \hat{x}_{i3} \boldsymbol{e}_3, \tag{6.10}$$

where $\hat{x}_{i1}$, $\hat{x}_{i2}$ and $\hat{x}_{i3}$ are the coordinates of the mass points measured *relative to the center of mass*, and expanding the angular momentum expression, yields

$$H_1 = \omega_1 \sum_{i=1}^{N_c} (\hat{x}_{i2}^2 + \hat{x}_{i3}^2)\, m_i - \omega_2 \sum_{i=1}^{N_c} \hat{x}_{i1}\hat{x}_{i2}\, m_i - \omega_3 \sum_{i=1}^{N_c} \hat{x}_{i1}\hat{x}_{i3}\, m_i \tag{6.11}$$

and

$$H_2 = -\omega_1 \sum_{i=1}^{N_c} \hat{x}_{i1}\hat{x}_{i2}\, m_i + \omega_2 \sum_{i=1}^{N_c} (\hat{x}_{i1}^2 + \hat{x}_{i3}^2)\, m_i - \omega_3 \sum_{i=1}^{N_c} \hat{x}_{i2}\hat{x}_{i3}\, m_i \tag{6.12}$$

and

$$H_3 = -\omega_1 \sum_{i=1}^{N_c} \hat{x}_{i1}\hat{x}_{i3}\, m_i - \omega_2 \sum_{i=1}^{N_c} \hat{x}_{i2}\hat{x}_{i3}\, m_i + \omega_3 \sum_{i=1}^{N_c} (\hat{x}_{i1}^2 + \hat{x}_{i2}^2)\, m_i, \tag{6.13}$$

which can be concisely written as

$$\boldsymbol{H}_{cm} = \overline{\boldsymbol{\mathcal{I}}} \cdot \boldsymbol{\omega}, \tag{6.14}$$

where we define the moments of inertia with respect to the center of mass

$$\overline{\mathcal{I}}_{11} = \sum_{i=1}^{N_c} (\hat{x}_{i2}^2 + \hat{x}_{i3}^2)\, m_i, \qquad \overline{\mathcal{I}}_{22} = \sum_{i=1}^{N_c} (\hat{x}_{i1}^2 + \hat{x}_{i3}^2)\, m_i, \qquad \overline{\mathcal{I}}_{33} = \sum_{i=1}^{N_c} (\hat{x}_{i1}^2 + \hat{x}_{i2}^2)\, m_i, \tag{6.15}$$

$$\overline{\mathcal{I}}_{12} = \overline{\mathcal{I}}_{21} = -\sum_{i=1}^{N_c} \hat{x}_{i1}\hat{x}_{i2}\, m_i, \qquad \overline{\mathcal{I}}_{23} = \overline{\mathcal{I}}_{32} = -\sum_{i=1}^{N_c} \hat{x}_{i2}\hat{x}_{i3}\, m_i, \qquad \overline{\mathcal{I}}_{13} = \overline{\mathcal{I}}_{31} = -\sum_{i=1}^{N_c} \hat{x}_{i1}\hat{x}_{i3}\, m_i, \tag{6.16}$$

or explicitly

$$\overline{\boldsymbol{\mathcal{I}}} = \begin{bmatrix} \overline{\mathcal{I}}_{11} & \overline{\mathcal{I}}_{12} & \overline{\mathcal{I}}_{13} \\ \overline{\mathcal{I}}_{21} & \overline{\mathcal{I}}_{22} & \overline{\mathcal{I}}_{23} \\ \overline{\mathcal{I}}_{31} & \overline{\mathcal{I}}_{32} & \overline{\mathcal{I}}_{33} \end{bmatrix}. \tag{6.17}$$

The particles' own inertia contribution about their respective mass-centers to the overall moment of inertia of the agglomerated body can be described by the Huygens-Steiner (generalized "parallel axis" theorem) formula ($p, s = 1, 2, 3$)

$$\bar{\mathcal{I}}_{ps} = \sum_{i=1}^{N_c} \left(\bar{\mathcal{I}}_{ps}^{i} + m_i(\|\boldsymbol{r}_i - \boldsymbol{r}_{cm}\|^2 \delta_{ps} - \hat{x}_{ip}\hat{x}_{is}) \right). \tag{6.18}$$

For a spherical particle, $\bar{\mathcal{I}}_{pp}^{i} = \frac{2}{5} m_i R_i^2$, and for $p \neq s$, $\bar{\mathcal{I}}_{ps}^{i} = 0$ (no products of inertia), R_i being the particle radius.[5] Finally, for the derivative of the angular momentum, utilizing $\ddot{\boldsymbol{r}}_i = \boldsymbol{a}_i = \boldsymbol{a}_{cm} + \boldsymbol{a}_{cm\to i}$, we obtain

$$\dot{\boldsymbol{H}}_{cm}^{rel} = \sum_{i=1}^{N_c} (\boldsymbol{r}_{cm\to i} \times m_i \boldsymbol{a}_{cm\to i}) = \sum_{i=1}^{N_c} (\boldsymbol{r}_{cm\to i} \times m_i(\boldsymbol{a}_i - \boldsymbol{a}_{cm})) \tag{6.19}$$

$$= \sum_{i=1}^{N_c} (m_i \boldsymbol{r}_{cm\to i} \times \boldsymbol{a}_i) - \underbrace{(\sum_{i=1}^{N_c} m_i \boldsymbol{r}_{cm\to i})}_{=0} \times \boldsymbol{a}_{cm} = \dot{\boldsymbol{H}}_{cm}, \tag{6.20}$$

and consequently

$$\dot{\boldsymbol{H}}_{cm} = \frac{d(\overline{\boldsymbol{\mathcal{I}}} \cdot \boldsymbol{\omega})}{dt} = \sum_{i=1}^{N_c} \boldsymbol{r}_{cm\to i} \times \boldsymbol{\psi}_i^{ext} \stackrel{\text{def}}{=} \boldsymbol{M}_{cm}^{EXT}, \tag{6.21}$$

where $\boldsymbol{M}_{cm}^{EXT}$ is the total external moment about the center of mass.

6.3. Numerical methods for the dynamics of a DEM cluster

We now treat the dynamics of a cluster numerically. We first focus on the translational motion of the center of mass, and then turn to the rotational contribution.

[5] If the particles are sufficiently small, each particle's own moment inertia (about its own center) is insignificant, leading to $\bar{\mathcal{I}}_{ps} = \sum_{i=1}^{N_c} m_i(\|\boldsymbol{r}_i - \boldsymbol{r}_{cm}\|^2 \delta_{ps} - \hat{x}_{ip}\hat{x}_{is})$.

6.3.1. DEM particle cluster translational contribution

The translational component of the center of mass can be written as

$$\mathcal{M}\ddot{\boldsymbol{r}}_{cm} = \mathcal{M}\dot{\boldsymbol{v}}_{cm} = \boldsymbol{\Psi}^{EXT}. \tag{6.22}$$

A trapezoidal time-stepping rule is used, whereby at some intermediate moment in time $t \le t + \phi\Delta t \le t + \Delta t$ $(0 \le \phi \le 1)$

$$\dot{\boldsymbol{v}}_{cm}(t+\phi\Delta t) \approx \frac{\boldsymbol{v}_{cm}(t+\Delta t) - \boldsymbol{v}_{cm}(t)}{\Delta t} \tag{6.23}$$

$$= \frac{1}{\mathcal{M}(t+\phi\Delta t)}\boldsymbol{\Psi}^{EXT}(t+\phi\Delta t) \tag{6.24}$$

$$\approx \frac{1}{\mathcal{M}(t+\phi\Delta t)}\left(\phi\boldsymbol{\Psi}^{EXT}(t+\Delta t) + (1-\phi)\boldsymbol{\Psi}^{EXT}(t)\right), \tag{6.25}$$

where $\mathcal{M}(t+\phi\Delta t) \approx \phi\mathcal{M}(t+\Delta t) + (1-\phi)\mathcal{M}(t)$, leading to

$$\boldsymbol{v}_{cm}(t+\Delta t) = \boldsymbol{v}_{cm}(t) + \frac{\Delta t}{\mathcal{M}(t+\phi\Delta t)}\left(\phi\boldsymbol{\Psi}^{EXT}(t+\Delta t) + (1-\phi)\boldsymbol{\Psi}^{EXT}(t)\right). \tag{6.26}$$

For the position, we have

$$\dot{\boldsymbol{r}}_{cm}(t+\phi\Delta t) \approx \frac{\boldsymbol{r}_{cm}(t+\Delta t) - \boldsymbol{r}_{cm}(t)}{\Delta t} \approx \boldsymbol{v}_{cm}(t+\phi\Delta t) \approx \left(\phi\boldsymbol{v}_{cm}(t+\Delta t) + (1-\phi)\boldsymbol{v}_{cm}(t)\right), \tag{6.27}$$

leading to

$$\boldsymbol{r}_{cm}(t+\Delta t) = \boldsymbol{r}_{cm}(t) + \Delta t\left(\phi\boldsymbol{v}_{cm}(t+\Delta t) + (1-\phi)\boldsymbol{v}_{cm}(t)\right). \tag{6.28}$$

6.3.2. DEM particle cluster rotational contribution

The quadcopter's angular velocity and rotation are determined in a similar manner by integrating the equations for an angular momentum balance

$$\dot{\boldsymbol{H}}_{cm} = \frac{d(\overline{\boldsymbol{\mathcal{I}}}\cdot\boldsymbol{\omega})}{dt} = \boldsymbol{M}_{cm}^{EXT}, \tag{6.29}$$

where $\overline{\boldsymbol{\mathcal{I}}}$ is the mass moment of the quadcopter, $\boldsymbol{\omega}$ is the angular velocity and $\boldsymbol{M}_{cm}^{EXT}$ is the sum of all moment contributions external to the quadcopter, around its center of mass. We remark that there are essentially two possible approaches to compute the rotational dynamics; either (1) an inertially-fixed frame or (2) a body-fixed frame. For the discrete element approach, it is advantageous to use a inertially-fixed frame.[6] The procedure is, within a time step, to decompose an increment of motion into a rigid body translation and rotation about the center of mass. The rotation is determined by solving for the angular velocity and the subsequent incremental rotation of the body around the axis of rotation, which is aligned with the angular velocity vector direction. This leads to a coupled set of nonlinear equations which are solved iteratively.

In a fixed frame of reference the angular momentum can be written as

$$\dot{\boldsymbol{H}}_{cm} = \frac{d(\overline{\boldsymbol{\mathcal{I}}}\cdot\boldsymbol{\omega})}{dt} = \boldsymbol{M}_{cm}^{EXT}. \tag{6.30}$$

$\overline{\boldsymbol{\mathcal{I}}}$ is implicitly dependent on $\boldsymbol{\omega}(t)$, leading to a coupled system of nonlinear ODE's. These will be solved iteratively. Eq. (6.30) is discretized by a trapezoidal scheme

$$\frac{d(\overline{\boldsymbol{\mathcal{I}}}\cdot\boldsymbol{\omega})}{dt}\Big|_{t+\phi\Delta t} = \frac{(\overline{\boldsymbol{\mathcal{I}}}\cdot\boldsymbol{\omega})|_{t+\Delta t} - (\overline{\boldsymbol{\mathcal{I}}}\cdot\boldsymbol{\omega})|_{t}}{\Delta t}. \tag{6.31}$$

thus leading to

$$(\overline{\boldsymbol{\mathcal{I}}}\cdot\boldsymbol{\omega})|_{t+\Delta t} = (\overline{\boldsymbol{\mathcal{I}}}\cdot\boldsymbol{\omega})|_{t} + \Delta t\boldsymbol{M}_{cm}^{EXT}(t+\phi\Delta t). \tag{6.32}$$

Solving for $\boldsymbol{\omega}(t+\Delta t)$ yields

$$\boldsymbol{\omega}(t+\Delta t) = \left(\overline{\boldsymbol{\mathcal{I}}}(t+\Delta t)\right)^{-1}\cdot\left((\overline{\boldsymbol{\mathcal{I}}}\cdot\boldsymbol{\omega})|_{t} + \Delta t\boldsymbol{M}_{cm}^{EXT}(t+\phi\Delta t)\right), \tag{6.33}$$

where

$$\boldsymbol{M}_{cm}^{EXT}(t+\phi\Delta t) \approx \phi\boldsymbol{M}_{cm}^{EXT}(t+\Delta t) + (1-\phi)\boldsymbol{M}_{cm}^{EXT}(t) \tag{6.34}$$

which yields an implicit nonlinear equation, of the form $\boldsymbol{\omega}(t+\Delta t) = \mathcal{F}(\boldsymbol{\omega}(t+\Delta t))$, since $\overline{\boldsymbol{\mathcal{I}}}(t+\Delta t)$, due to the body's rotation. An iterative, implicit, solution scheme may be written as follows for $K = 1, 2 \ldots$

$$\boldsymbol{\omega}^{K+1}(t+\Delta t) = \left(\overline{\boldsymbol{\mathcal{I}}}^{K}(t+\Delta t)\right)^{-1}\cdot\left((\overline{\boldsymbol{\mathcal{I}}}\cdot\boldsymbol{\omega})|_{t} + \Delta t\boldsymbol{M}_{cm}^{EXT,K}(t+\phi\Delta t)\right), \tag{6.35}$$

[6] For a body-fixed formulation, see Powell and Zohdi [89].

where $\overline{\mathcal{I}}^K(t+\Delta t)$ can be re-computed from the previous formulas.[7] After the update for $\boldsymbol{\omega}^{K+1}(t+\Delta t)$ has been computed (utilizing the $\overline{\mathcal{I}}^K(t+\Delta t)$ from the previous iteration), the rotation of the body about the center of mass can be determined.

Remark-Propellor thrust: A propeller's thrust is directly proportional to its speed of rotation. This relationship is nonlinear, with airspeed, propellor design etc. controlling the overall thrust produced. In the analysis at hand, we simply assign a thrust to each motor. In the case of a hovering action, each carrying 1/4th the gravitation load. Two of the motors spin with rotation vectors pointing upwards and two pointing downwards, although the thrust is upwards for all 4 are by having the propellors flipped on two downward spinning propellors. For more details see [90].

6.3.3. Iterative superposition scheme-including loss of dislodged particles

The total velocity of any particle can be decomposed into the velocity of the center of mass of the entire object and the rotation of the particle relative to the center of mass:

$$\boldsymbol{v}_i = \boldsymbol{v}_{cm} + (\boldsymbol{v}_i - \boldsymbol{v}_{cm}) = \boldsymbol{v}_{cm} + \boldsymbol{v}_{cm\to i} = \boldsymbol{v}_{cm} + \boldsymbol{\omega}\times(\boldsymbol{r}_{cm} - \boldsymbol{r}_i) = \boldsymbol{v}_{cm} + \boldsymbol{\omega}\times\boldsymbol{r}_{cm\to i} \tag{6.36}$$

Explicitly, the overall motion for the bonded particles is computed by $\boldsymbol{r}_i = \boldsymbol{r}_{cm} + \boldsymbol{\omega}\times(\boldsymbol{r}_i - \boldsymbol{r}_{cm})$, sequentially by computing:

- **Velocity:** $\boldsymbol{C}_1 = \phi\boldsymbol{v}_{cm}(t+\Delta t) + (1-\phi)\boldsymbol{v}_{cm}(t)$,
- **Angular velocity:** $\boldsymbol{C}_2 = \phi\boldsymbol{\omega}(t+\Delta t) + (1-\phi)\boldsymbol{\omega}(t)$,
- **Center of mass position:** $\boldsymbol{C}_3 = \phi\boldsymbol{r}_{cm}(t+\Delta t) + (1-\phi)\boldsymbol{r}_{cm}(t)$,
- **Particle positions:** $\boldsymbol{r}_i(t+\Delta t) = \boldsymbol{r}_i + \Delta t(\boldsymbol{C}_1 + \boldsymbol{C}_5)$,
- $\boldsymbol{C}_4 = \phi\boldsymbol{r}_i(t+\Delta t) + (1-\phi)\boldsymbol{r}_i(t) - \boldsymbol{C}_3$ and $\boldsymbol{C}_5 = \boldsymbol{C}_2\times\boldsymbol{C}_4$.

Remark-Option for Dislodged Particles: Although it is outside of the scope of the present work, to incorporate the possibility for particles to break off, a unilateral fragmentation threshold must be met for a particle to be deemed "dislodged", which subsequently moves according to its own dynamics. We refer the reader to Zohdi [91], [92], [93], [94], [95], [96] for details. This is discussed further in the summary and conclusions.

6.4. Algorithmic procedure

The overall procedure is as follows, at time t:

1. Generate the quadcopter body by inserting particles within the envelope/grid interaction (Fig. 11):
$$\frac{|x_1 - x_{1o}|^{p_1}}{R_1} + \frac{|x_2 - x_{2o}|^{p_1}}{R_2} + \frac{|x_3 - x_{3o}|^{p_3}}{R_3} \le 1. \tag{6.37}$$
Also place extra masses in the locations for the motors.
2. Set initial conditions, if $t = 0$.
3. Compute the thrust of the motors (orthogonal to the quadcopter body).
4. Compute the new position of the center of mass.
5. Compute (iteratively) the positions of the particles in the body $\boldsymbol{r}_i^K(t+\Delta t)$, $K = 1, 2, \ldots$:
$$\|\boldsymbol{r}_i^{K+1}(t+\Delta t) - \boldsymbol{r}_i^K(t+\Delta t)\| \le TOL\|\boldsymbol{r}_i^{K+1}(t+\Delta t)\|. \tag{6.38}$$
This requires computation of the position of the center of mass, the rotation of the body, and the calculation of the positions of the particles within the iterations:
[a] Compute/update: $\boldsymbol{v}_{cm}^{K+1}(t+\Delta t) = \boldsymbol{v}_{cm}(t) + \frac{\Delta t}{\mathcal{M}(t+\phi\Delta t)}\left(\phi\boldsymbol{\Psi}^{K+1,EXT}(t+\Delta t) + (1-\phi)\boldsymbol{\Psi}^{EXT}(t)\right)$.
(b) Compute/update: $\boldsymbol{r}_{cm}^{K+1}(t+\Delta t) = \boldsymbol{r}_{cm}(t) + \Delta t\left(\phi\boldsymbol{v}_{cm}(t+\Delta t) + (1-\phi)\boldsymbol{v}_{cm}(t)\right)$.
(c) Compute/update: $\boldsymbol{M}_{cm}^{EXT}(t+\phi\Delta t) \approx \phi\boldsymbol{M}_{cm}^{EXT}(t+\Delta t) + (1-\phi)\boldsymbol{M}_{cm}^{EXT}(t)$,
(d) Compute/update: $\boldsymbol{\omega}^{K+1}(t+\Delta t) = \left(\overline{\mathcal{I}}^K(t+\Delta t)\right)^{-1}\cdot\left((\overline{\mathcal{I}}\cdot\boldsymbol{\omega})|_t + \Delta t\boldsymbol{M}_{cm}^{EXT,K}(t+\phi\Delta t)\right)$,
(e) Compute/update: $\boldsymbol{v}_i = \boldsymbol{v}_{cm} + \boldsymbol{\omega}\times\boldsymbol{r}_{cm\to i}$
(f) Compute/update: $\boldsymbol{r}_i(t+\Delta t) = \boldsymbol{r}_i + \Delta t(\boldsymbol{C}_1 + \boldsymbol{C}_5)$,
(g) Repeat steps (a)–(f) until Eq. (6.38) is satisfied.
6. Increment time forward and repeat the procedure.

6.5. Numerical examples

As an example, consider an antiaircraft system that explodes pressure waves at a hovering drone. We consider an initially hovering drone, with the thrust from the propellors always acting perpendicular to the drone, being hit repeatedly with a set of

[7] One may view the overall process as a fixed-point calculation of the form $\boldsymbol{\omega}^{K+1}(t+\Delta t) = \mathcal{F}(\boldsymbol{\omega}^K(t+\Delta t))$.

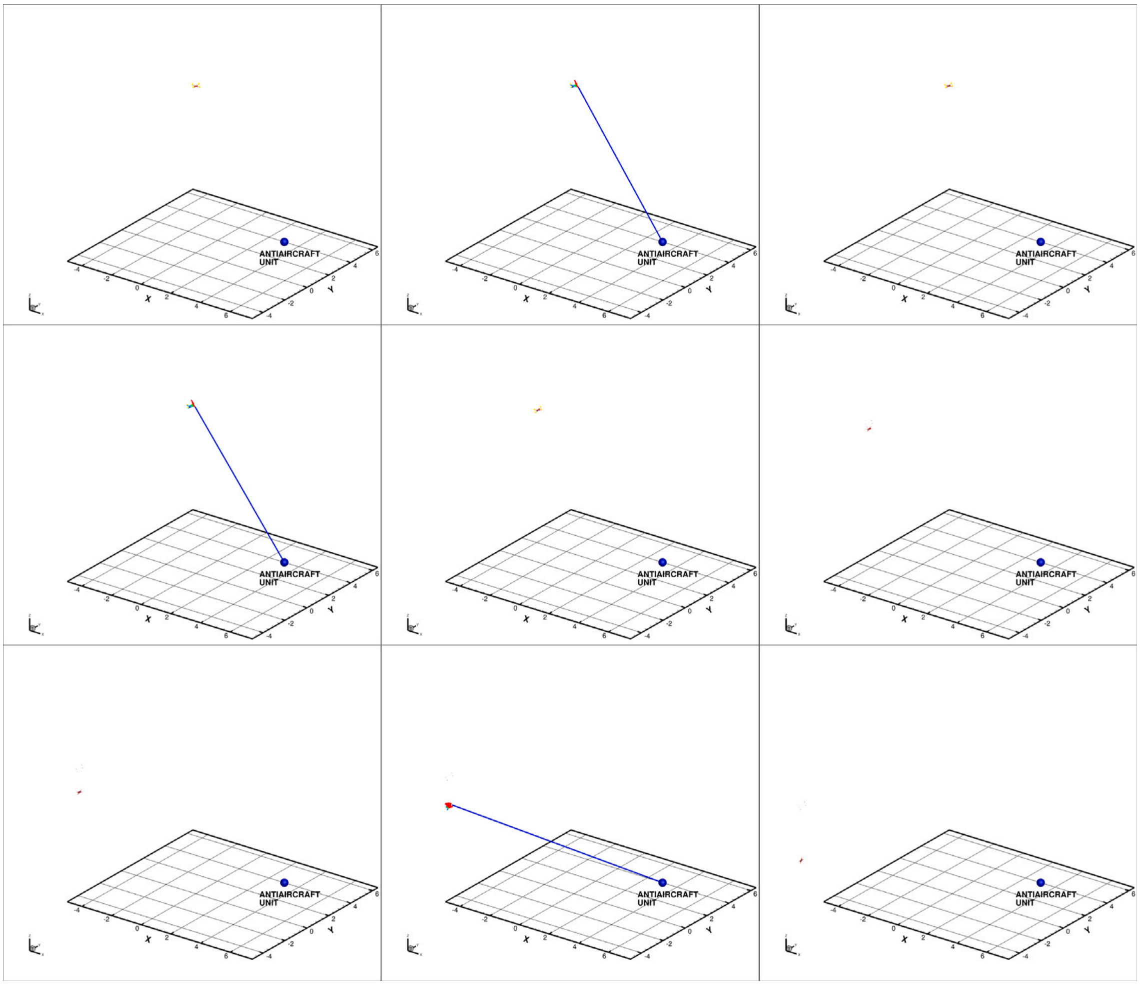

Fig. 14. Destabilization of a quadcopter by repeated explosive blasts. See zoom in Figs. 15 and 16.

pulses, where the maximum intensity slightly off center (see Figs. 14–16) with a decay that is scaled by the distance from the targeting centerline, according to:

$$P(\boldsymbol{x}) = P_o e^{-ad(\boldsymbol{x})} \tag{6.39}$$

where $P(\boldsymbol{x})$ is the pressure at $\boldsymbol{x}$, P_o is the pressure the center, $d(\boldsymbol{x})$ is the distance from the center to $\boldsymbol{x}$ and a is a decay coefficient. The following simulation parameters were chosen:

- Generation grid for DEM: 100 × 100 × 100, yielding 11179 intersecting sites and hence 11179 particles,
- Total time duration: $T = 35$ seconds,
- Time step size: $\Delta t = 0.00005$ seconds,
- Starting position of center of mass: $\boldsymbol{r}_{cm}(t = 0) = \boldsymbol{0}$ (horizontal, Fig. 13),
- Time stepping parameter: $\phi = 0.5$ (midpoint rule),
- Drone shape exponents: $(p_1, p_2, p_3) = (0.5, 0.5, 0.5)$ (Fig. 13),
- Size of drone: $(R_1, R_2, R_3) = (0.25, 0.25, 0.05)$,
- Mass of the drone chassis: $M = 1$ kg,
- Starting angular velocity: $\boldsymbol{\omega}(t = 0) = \boldsymbol{0}$ rad/sec,
- Motor masses: $M_m = 0.25$ kg each,
- Thrust force per motor:4.55 N (this allows for perfect hovering (propellor thrust balancing gravity) if there is no external impulse),
- Density of the chassis material: $\rho = 1000$ kg/m^3,
- Blast frequency: one every second.

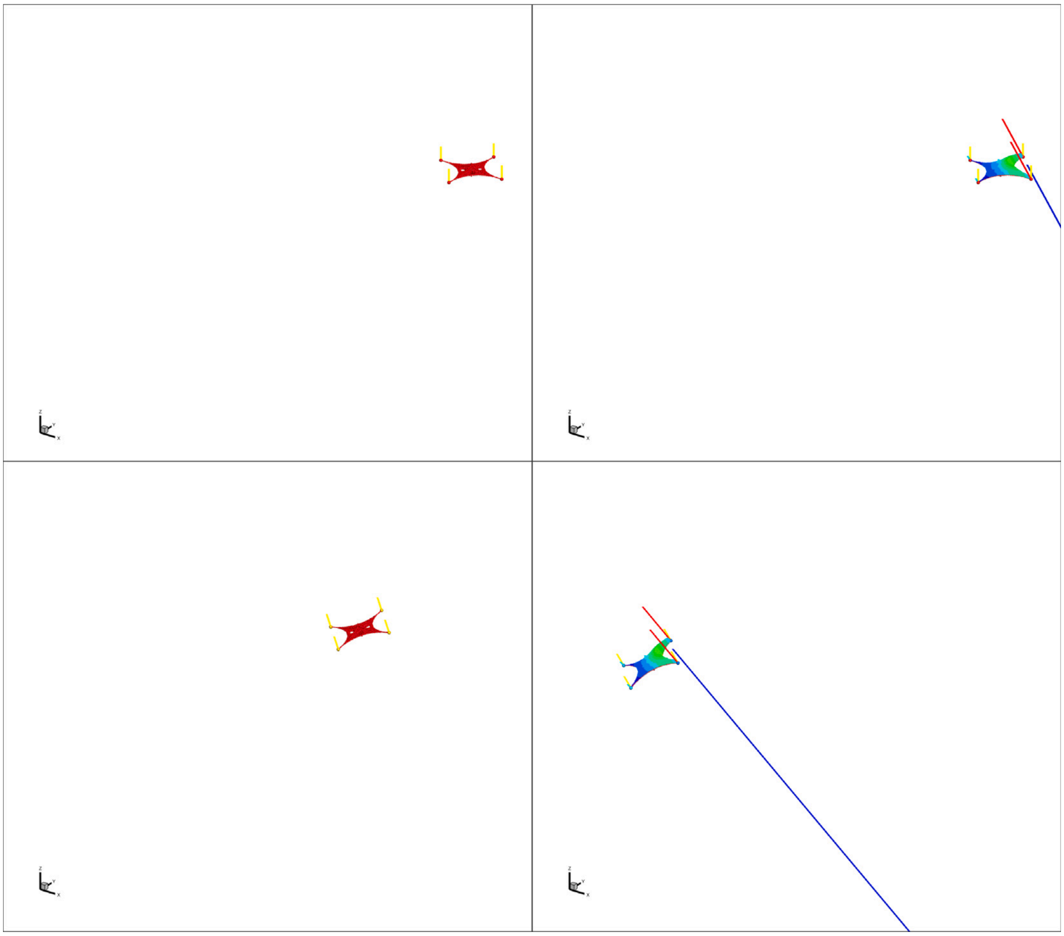

Fig. 15. Destabilization of a quadcopter by repeated explosive pulses, with a zoom closeup (previously shown in Fig. 14).

As seen in Figs. 14–16, the repeated impulses are strong enough to overturn the drone, and to destabilize it, as well as to separate of the motors. Importantly, although we have not discussed the evolution of damage, these approaches are important for ascertaining whether a particle will become dislodged by checking a unilateral *integrity bonding criteria* between particles:

- Remains bonded/intact: $\|\boldsymbol{F}(\boldsymbol{x})\| \le F^* \Rightarrow \dot{\gamma}(\boldsymbol{x}) = 0$.
- Debonding evolution: $\|\boldsymbol{F}(\boldsymbol{x})\| \ge F^* \Rightarrow \dot{\gamma}(\boldsymbol{x}) = \alpha(1 - \frac{\|\boldsymbol{F}\|}{F^*})$,

where α is a rate parameter and γ is a integrity parameter, where initially $\gamma(\boldsymbol{x}, t = 1) = 1$; i.e. no damage/full integrity. When $\gamma \le TOL$ then the particle is allowed to debond and mass is lost. We define γ as the integrity and $1-\gamma$ as damage. If the part of the body separates then it is released with the position and velocity of the body at that point. The contact mechanics of dislodged particles with other dislodged material, the remainder of the drone hull, etc., is outside the scope of the present work. Temporally-adaptive iterative methods maybe needed for more complex particle interaction. We refer the reader to methods found in Zohdi [91], [92], [93], [94], [95], [96] that address general systems of this type. As an example, consider a debonding relation that is dependent on the intensity of the external blast force applied:

- Remains bonded/intact: $\|\boldsymbol{F}^{ext}(\boldsymbol{x})\| \le F^* \Rightarrow \dot{\gamma}(\boldsymbol{x}) = 0$,
- Evolutionary debonding: $\|\boldsymbol{F}^{ext}(\boldsymbol{x})\| \ge F^* \Rightarrow \dot{\gamma}(\boldsymbol{x}) = \alpha(1 - \frac{\|\boldsymbol{F}^{ext}\|}{F^*})$

The results of this assumption are shown in Figs. 17 and 18. When $\gamma(\boldsymbol{x}) \le TOL$, then the fragment is released.

In the previous example, we did not compute the interaction of fragments with the chasis or other fragments. Furthermore, regardless of whether the particles surrounding a particle were dislodged, if a particle does not meet the dislodging criteria, it still moves with the main (remaining) rigid body and does not separate from the main body. In other words, the topology of the surrounding fragmentation, for example separating an entire rotor arm, has not been taken into account. In this simple example, the fragmentation is particle by particle. Clearly, the interaction of clusters of fragments, with a full contact analysis, is a logical extension

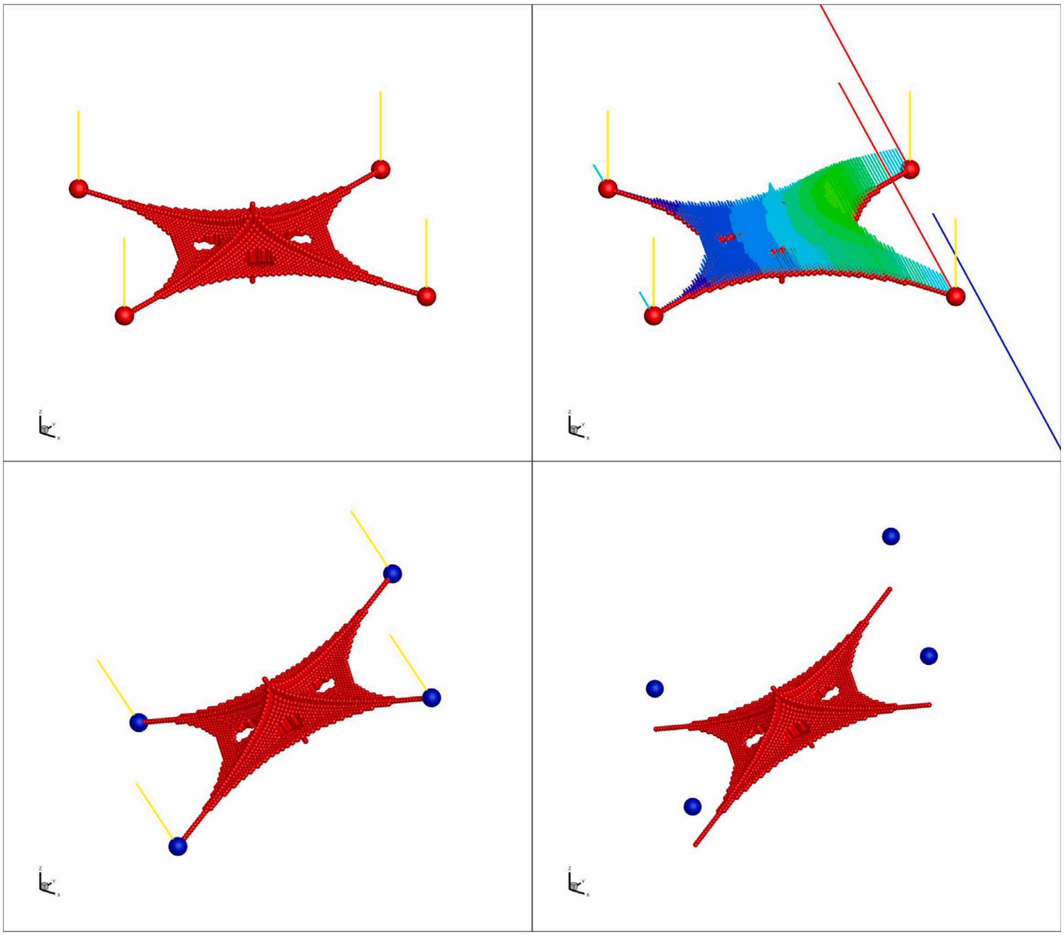

Fig. 16. Destabilization of a quadcopter by repeated explosive pulses, with a larger zoom closeup with loss of motors (previously shown in Figs. 14 and 15). The blue color coding indicated that the motors have shut off and are non-functioning. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the analysis presented. This would require a direct numerical simulation-type formulation for dynamics of a multi-particulate system ($i = 1, 2...N$ bonded and fragmented particles)

$$m_i\ddot{\boldsymbol{r}}_i = \boldsymbol{\Psi}_i^{tot}(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{N_p}, \text{contact forces}, \text{bonding forces}) \tag{6.40}$$

where $\boldsymbol{r}_i$ is the position vector of the ith particle, $\boldsymbol{\Psi}_i^{tot}$ represents all forces acting on particle i, with contact detection and contact force computation. The simulation of related flowing particulate systems has been extensively investigated for the last decade by Zohdi [97]–[96], employing numerical schemes based on high-performance iterative solvers, sorting-binning for fast inter-particle calculations, Verlet lists, domain decomposition, parallel processing and temporally-adaptive methods, described further in the Appendices. These types of formulations can easily describe the interaction of multiple fragments from breakup/disintegration and blasts where the application of continuum approaches would be virtually impossible. The dynamics of fragments of clusters that evolve and interact with the quadcopter and other fragments is complex. Studies on cluster-to-cluster interaction can be found in Zohdi [91], [92], [93], [94], [95], [96]. There is an extremely close area to this field, namely the study of "granular" or "particulate" media, for example see Duran [98], Pöschel and Schwager [99], Onate et al. [100], [101], Rojek et al. [102], Carbonell et al. [103], Labra and Onate [104] and Zohdi [94]. In summary, with such a formulation, one can explore aspects such as (1) trade-offs between mass, structural stabilization, vulnerability-from the attacker's point of view, (2) inverse problems for prescribed target destabilization-from the defender's point of view and (3) subsequent trajectory analysis, to name a few.

Remark-Incorporation of reduced-order models: While the simulator we have presented is quite fast, even faster reactions are needed in some critical situations, utilizing reduced order models for example using Artificial Neural Nets (ANN). In the realm of model reduction, two possible extensions come to mind:

- Reducing the complexity of the model-based method to an ANN by training on new synthetic outcomes for which there is no experimental data

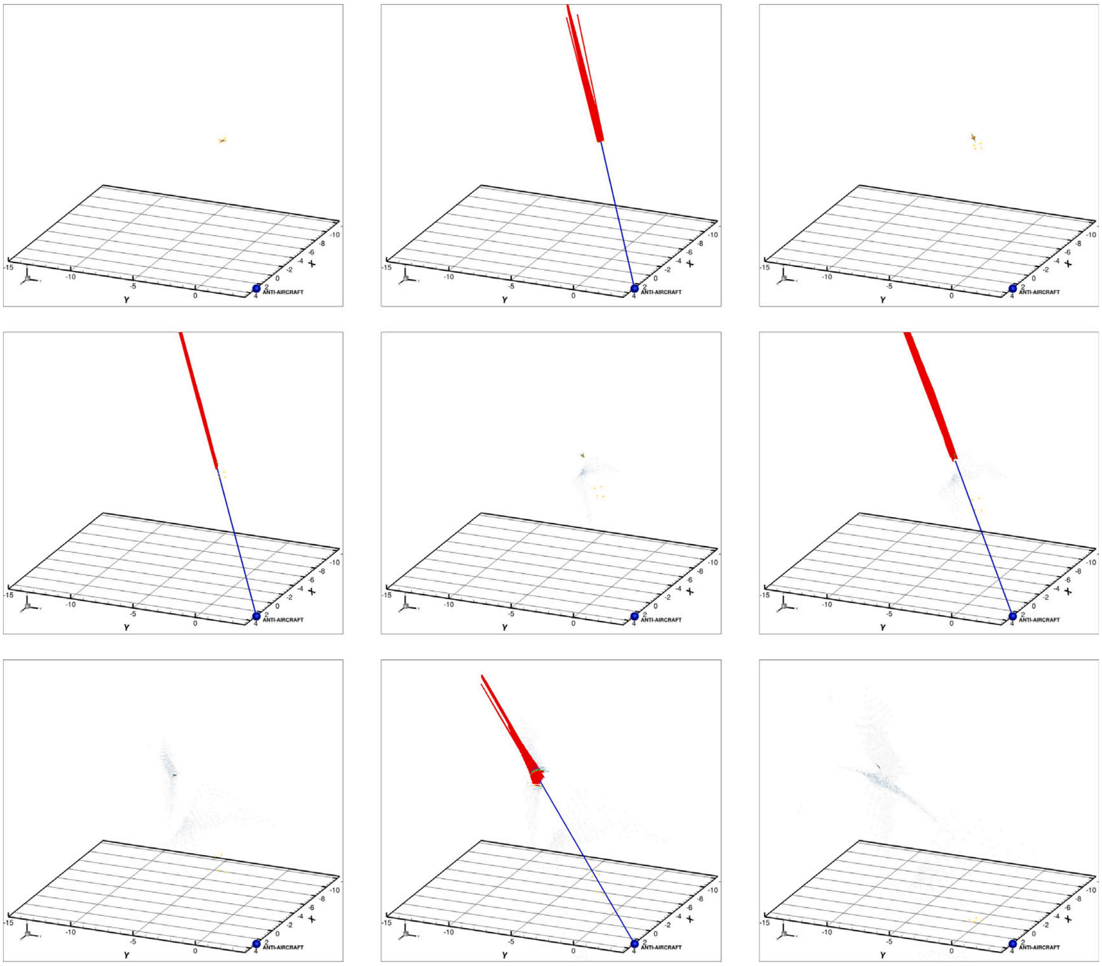

Fig. 17. Destabilization, *fragmentation* and disintegration of a quadcopter by repeated explosive pulses, with resulting debris formation. See zoom in Fig. 18.

- Directly producing model-free ANN from the incremental data, thus completely circumventing any model from the very beginning.

In theory, an ANN reduced order model could accelerate simulations even further. However, huge amounts of data for training sets would be needed to compensate for the fact that no physics is built into the Neural Nets. The hybrid use of digital-twins, genetic-based machine-learning and Artificial Neural Networks is currently under investigation by the author. We refer the reader to Appendix 3 for more details.

7. Summary and the role of evolving technology

In summary, the goal of this work was to develop a rapid machine-learning enabled digital-twin to ascertain optimal programming for desired tactical multi-drone swarmlike behavior. There were main two components of this analysis. The first component was a framework comprised of a multibody dynamics model for multiple interacting agents, augmented with a machine-learning framework to allow drone-swarm agents to identify (a) desired targets, (b) obstacles and (c) fellow agents, as well to determine the resulting optimal actions the agents should undertake. The objective was to construct a system with entirely autonomous interaction and actuation parameter values that are embedded in the coupled multibody differential equations of motion for the drone-swarm agents. This was achieved by minimizing a cost/error function that represented the difference between the simulated overall group behavior and in-field observed behavior from synthetic data in the form of temporal snapshots that would correspond to multiple camera frames. In the second main component of the analysis, in order more deeply assess the design of a drone-swarm member, we developed a method for rapid chassis design, deployment and dynamic performance. As an example, we studied the response of tactical quadcopter drones under attack, specifically by being subjected to series of launched explosions. A Discrete Element Method (DEM) was developed to rapidly design a quadcopter of virtually any complex shape, attach motors and then to subject it

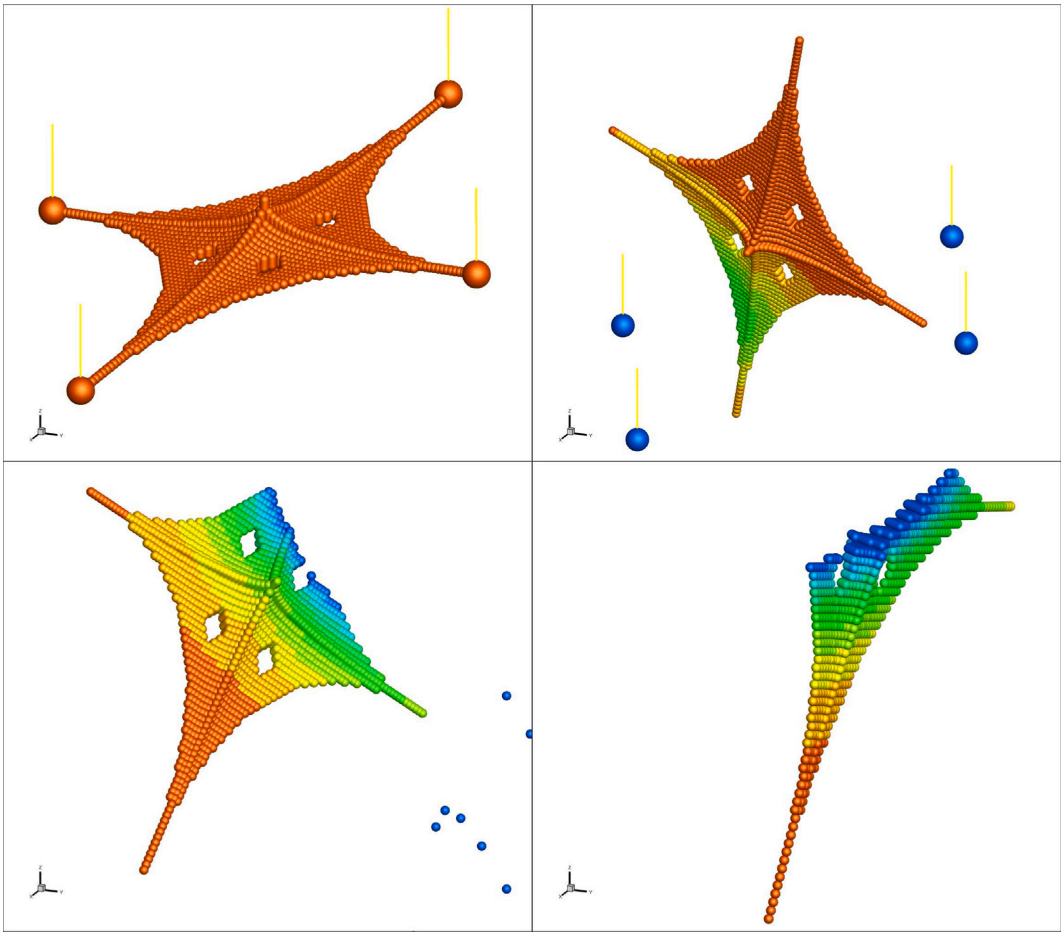

Fig. 18. Zoom on destabilization, *fragmentation* and disintegration (previously shown in Fig. 17) of a quadcopter by repeated explosive pulses. Orange indicates full structural integrity, while blue indicates high structural damage. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to a hostile environment, in order to ascertain its performance in the field. The method also allowed for the characterization of loss of functionality (thrust), progressive damage and fragmentation of the quadcopter drone.

Typically, such drones have a variety of sensors on board, ranging from 6 DOF 3-axis gyroscopes, a variety of accelerometers, Inertial Measurement Units (IMU's), barometers and GPS receivers. They also have a number of actuators, such as digital electronic speed controllers that are connected to engine, motors, servomotors and propellors. There are numerous emerging technologies that have been enabled by drones, driven primarily by defense systems. As a result, there are consequential geopolitical issues associated with *quasi-commericial* drones, in particular related to defense and security issues. While small-scale commercial drones were initially considered of little concern, recent wars, such as the Ukraine conflict have starkly altered that view. In particular, First-Person Viewer (FPV) drones, which stream a video to a ground-based controller, have become centrally important. Furthermore, when retrofitted to carry explosives, they become a lethal weapon. There are roughly ten key components common to most tactical drones: (1) Lightweight airframes, (2) Brushless DC motors, (3) Power transmission, (4) Flight controls, (5) Batteries and power supply, (6) Cameras, (7) Sensors, (8) Navigation, (9) Onboard computing, processing and memory and (10) Wireless communication. Additionally, there are highly specialized materials involved, such as sintered permanent neodymium (NdFeB) for brushless DC motors and lithium polymers (LiPo) for lightweight batteries are critical components. In particular, over the last 30 years, battery technology utilize gel polymers and solid-state electrolytes to deliver extremely high energy per unit of weight that have led to approximately a 300% increase in energy density and nearly a 100% reduction in cost of lithium-ion batteries. As of 2025, companies based in China dominate the commercial drone market, with overall estimates of nearly 85% market share (see popular press articles [105–125]), producing drones that are easy to use, simple to maintain and relatively inexpensive, backed by a skilled workforce who is able to supply and innovate products at unmatched rapid speeds. However, over that last two years, there has been huge worldwide investments, in particular by the US, Taiwan, Europe in drone technologies, in parallel with the rise of AI and machine-learning. The research emphasis is now focussing on collaborative multi-drone swarm technologies. In this regard,

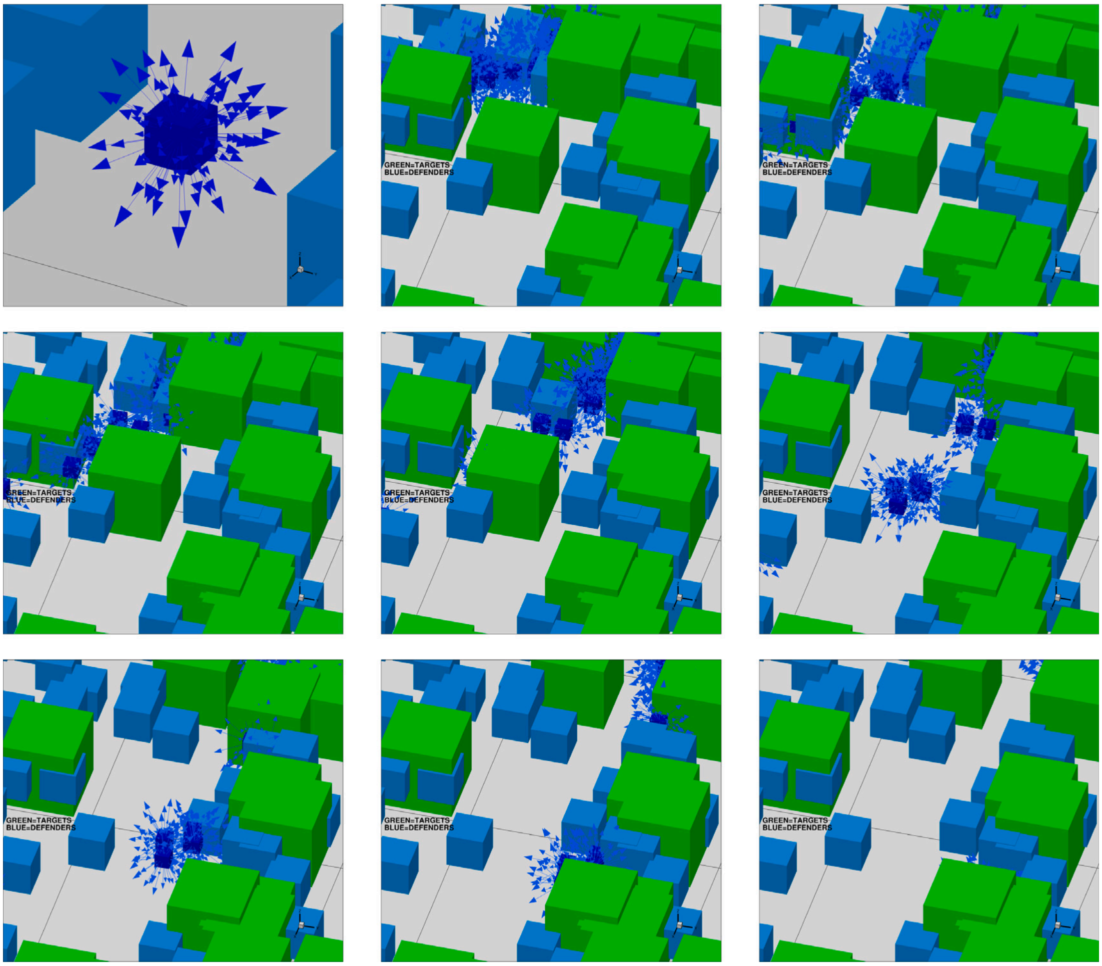

Fig. 19. A zoom on the LiDAR scanning of drone-swarm members as they move through a complex environment.

as mentioned earlier in the present work, advances in camera technologies have become critical for drone-swarm technologies propagate.

The role of evolving camera technology for complex drone manipulation cannot be overstated. In particular, cameras that incorporate (a) multispectral imaging that extend the classical visible wavelength paradigms (380–720 nm), to thermographic/infrared regimes (1000–14000 nm) to create an image and (b) 3D (time-of-flight) cameras, using LiDAR, radio-based imaging and tomography are critical. It is now possible to rapidly extract, frame-by-frame, 3D voxel fields of dynamic thermo-fluidic events utilizing real-time tomographically-based imaging. For example, in Fig. 19 LiDAR rays are shown being emitted from each drone, scanning each object in a multidirectional manner. We refer the interested reader to Elsinga et al. [126–151]–Tariq [152] for details on the wide-range of topics discussed above. This is particularly critical due to the increase of heat-seeking weaponry, unidentified flying objects, drones, high-altitude remote sensing, satellite constellations, etc., in an increasingly crowded airspace. Furthermore, methods such as tomographic-PIV has evolved over the last 20 years to extend PIV to measuring 3D vector fields, using MART (Multiplicative Algebraic Reconstruction Technique), which was introduced by Herman and Lent [127] that creates a digital voxel representation of the volume, where the intensity values correlate to the values represented by the particles at those locations. We refer the interested reader to Elsinga et al. [126–151]–Tariq [152] and Aguirre-Pablo, et al. [130] who demonstrated the viability of using four low-cost smartphone cameras to perform Tomographic PIV using colored shadows to imprint two or three different time-steps on the same image. The use of voxels (Foley et al. [153]), is widespread in the visualization and analysis of medical and scientific data (Chmielewski et al. [154]) and in the video-gaming industry. Accordingly, in Zohdi [67], a machine-learning framework was developed that rapidly simulates and adapts object geometries in order to match the thermo-flow field signature generated by an unknown object, across a time series of voxel-frames. In order to achieve this, a thermo-fluid model was developed, based on rapid-solution of coupled multiphysical flows involving the Navier–Stokes equations and the first law of thermodynamics, using a voxel rendering of the domain environment. This voxel-framework was then combined with a genomic-based machine-learning algorithm to develop a digital-twin (digital-replica) of the system that can run in real-time or faster than the actual physical system. The adaptation of these methods for drone-based data acquisition and, conversely, drone identification, is currently under investigation by the author.

Declaration of competing interest

I declare that I have no conflicting interests.

Acknowledgments

This work was partially supported by the UC Berkeley College of Engineering.

Appendix A. Temporal discretization

Consider a particle's equation of motion given by

$$m\dot{\boldsymbol{v}} = \boldsymbol{\Psi}, \tag{A.1}$$

where $\boldsymbol{\Psi}$ is the force provided from interactions with other particles the external environment. Expanding the velocity in a Taylor series about $t + \phi\Delta t$ we obtain ($0 \le \phi \le 1$)

$$\boldsymbol{v}(t + \Delta t) = \boldsymbol{v}(t + \phi\Delta t) + \frac{d\boldsymbol{v}}{dt}|_{t+\phi\Delta t}(1 - \phi)\Delta t + \frac{1}{2}\frac{d^2\boldsymbol{v}}{dt^2}|_{t+\phi\Delta t}(1 - \phi)^2(\Delta t)^2 + \mathcal{O}(\Delta t)^3 \tag{A.2}$$

and

$$\boldsymbol{v}(t) = \boldsymbol{v}(t + \phi\Delta t) - \frac{d\boldsymbol{v}}{dt}|_{t+\phi\Delta t}\phi\Delta t + \frac{1}{2}\frac{d^2\boldsymbol{v}}{dt^2}|_{t+\phi\Delta t}\phi^2(\Delta t)^2 + \mathcal{O}(\Delta t)^3. \tag{A.3}$$

Subtracting the two expressions yields

$$\frac{d\boldsymbol{v}}{dt}|_{t+\phi\Delta t} = \frac{\boldsymbol{v}(t + \Delta t) - \boldsymbol{v}(t)}{\Delta t} + \hat{\mathcal{O}}(\Delta t), \tag{A.4}$$

where $\hat{\mathcal{O}}(\Delta t) = \mathcal{O}(\Delta t)^2$ when $\phi = \frac{1}{2}$. Inserting this into the equation of motion yields

$$\boldsymbol{v}(t + \Delta t) = \boldsymbol{v}(t) + \frac{\Delta t}{m}\boldsymbol{\Psi}(t + \phi\Delta t) + \hat{\mathcal{O}}(\Delta t)^2. \tag{A.5}$$

Note that adding a weighted sum of Eqs. (A.2) and (A.3) yields

$$\boldsymbol{v}(t + \phi\Delta t) = \phi\boldsymbol{v}(t + \Delta t) + (1 - \phi)\boldsymbol{v}(t) + \mathcal{O}(\Delta t)^2, \tag{A.6}$$

which will be useful shortly. Now expanding the position of the center of mass in a Taylor series about $t + \phi\Delta t$ we obtain

$$\boldsymbol{r}(t + \Delta t) = \boldsymbol{r}(t + \phi\Delta t) + \frac{d\boldsymbol{r}}{dt}|_{t+\phi\Delta t}(1 - \phi)\Delta t + \frac{1}{2}\frac{d^2\boldsymbol{r}}{dt^2}|_{t+\phi\Delta t}(1 - \phi)^2(\Delta t)^2 + \mathcal{O}(\Delta t)^3 \tag{A.7}$$

and

$$\boldsymbol{r}(t) = \boldsymbol{r}(t + \phi\Delta t) - \frac{d\boldsymbol{r}}{dt}|_{t+\phi\Delta t}\phi\Delta t + \frac{1}{2}\frac{d^2\boldsymbol{r}}{dt^2}|_{t+\phi\Delta t}\phi^2(\Delta t)^2 + \mathcal{O}(\Delta t)^3. \tag{A.8}$$

Subtracting the two expressions yields

$$\frac{\boldsymbol{r}(t + \Delta t) - \boldsymbol{r}(t)}{\Delta t} = \boldsymbol{v}(t + \phi\Delta t) + \hat{\mathcal{O}}(\Delta t). \tag{A.9}$$

Inserting Eq. (A.6) yields

$$\boldsymbol{r}(t + \Delta t) = \boldsymbol{r}(t) + (\phi\boldsymbol{v}(t + \Delta t) + (1 - \phi)\boldsymbol{v}(t))\,\Delta t + \hat{\mathcal{O}}(\Delta t)^2 \tag{A.10}$$

and thus using Eq. (A.5) yields

$$\boldsymbol{r}(t + \Delta t) = \boldsymbol{r}(t) + \boldsymbol{v}(t)\Delta t + \frac{\phi(\Delta t)^2}{m}\boldsymbol{\Psi}(t + \phi\Delta t) + \hat{\mathcal{O}}(\Delta t)^2. \tag{A.11}$$

The term $\boldsymbol{\Psi}(t + \phi\Delta t)$ can be approximated by

$$\boldsymbol{\Psi}(t + \phi\Delta t) \approx \phi\boldsymbol{\Psi}(\boldsymbol{r}(t + \Delta t)) + (1 - \phi)\boldsymbol{\Psi}(\boldsymbol{r}(t)), \tag{A.12}$$

yielding

$$\boldsymbol{r}(t + \Delta t) = \boldsymbol{r}(t) + \boldsymbol{v}(t)\Delta t + \frac{\phi(\Delta t)^2}{m}\left(\phi\boldsymbol{\Psi}(\boldsymbol{r}(t + \Delta t)) + (1 - \phi)\boldsymbol{\Psi}(\boldsymbol{r}(t))\right) + \hat{\mathcal{O}}(\Delta t)^2. \tag{A.13}$$

We note that

- When $\phi = 1$, then this is the (implicit) Backward Euler scheme, which is very stable (very dissipative) and $\mathcal{O}(\Delta t)^2$ locally in time,
- When $\phi = 0$, then this is the (explicit) Forward Euler scheme, which is conditionally stable and $\mathcal{O}(\Delta t)^2$ locally in time,
- When $\phi = 0.5$, then this is the (implicit) "Midpoint" scheme, which is stable and $\hat{\mathcal{O}}(\Delta t)^2 = \mathcal{O}(\Delta t)^3$ locally in time.

Appendix B. Temporally adaptive iterative schemes

For illustration purposes, after time discretization of the acceleration term in the equations of motion $m\ddot{\boldsymbol{r}} = \boldsymbol{\Psi}$ using a ϕ-method

$$\boldsymbol{r}^{L+1} = \boldsymbol{r}^L + \boldsymbol{v}^L \Delta t + \frac{\phi(\Delta t)^2}{m}\left(\phi\boldsymbol{\Psi}(\boldsymbol{r}^{L+1}) + (1-\phi)\boldsymbol{\Psi}(\boldsymbol{r}^L)\right), \tag{B.1}$$

one arrives at the following abstract form, for the entire system of particles,

$$\mathcal{A}(\boldsymbol{r}^{L+1}) = \mathcal{F}. \tag{B.2}$$

It is convenient to write

$$\mathcal{A}(\boldsymbol{r}^{L+1}) - \mathcal{F} = \mathcal{G}(\boldsymbol{r}^{L+1}) - \boldsymbol{r}^{L+1} + \mathcal{R} = \boldsymbol{0}, \tag{B.3}$$

where $\mathcal{R}$ is a remainder term that does not depend on the solution, i.e. $\mathcal{R} \neq \mathcal{R}(\boldsymbol{r}^{L+1})$. A straightforward iterative scheme can be written as

$$\boldsymbol{r}^{L+1,K} = \mathcal{G}(\boldsymbol{r}^{L+1,K-1}) + \mathcal{R}, \tag{B.4}$$

where $K = 1, 2, 3, \ldots$ is the index of iteration within time step $L+1$. The convergence of such a scheme is dependent on the behavior of $\mathcal{G}$. Namely, a sufficient condition for convergence is that $\mathcal{G}$ is a contraction mapping for all $\boldsymbol{r}^{L+1,K}$, $K = 1, 2, 3\ldots$ In order to investigate this further, we define the iteration error as $\epsilon^{L+1,K} \stackrel{\text{def}}{=} \boldsymbol{r}^{L+1,K} - \boldsymbol{r}^{L+1}$. A necessary restriction for convergence is iterative self consistency, i.e. the "exact" (discretized) solution must be represented by the scheme

$$\mathcal{G}(\boldsymbol{r}^{L+1}) + \mathcal{R} = \boldsymbol{r}^{L+1}. \tag{B.5}$$

Enforcing this restriction, a sufficient condition for convergence is the existence of a contraction mapping

$$\epsilon^{L+1,K} = \|\boldsymbol{r}^{L+1,K} - \boldsymbol{r}^{L+1}\| = \|\mathcal{G}(\boldsymbol{r}^{L+1,K-1}) - \mathcal{G}(\boldsymbol{r}^{L+1})\| \tag{B.6}$$

$$\leq \eta^{L+1,K}\|\boldsymbol{r}^{L+1,K-1} - \boldsymbol{r}^{L+1}\|, \tag{B.7}$$

where, if $0 \leq \eta^{L+1,K} < 1$ for each iteration K, then $\epsilon^{L+1,K} \to \boldsymbol{0}$ for any arbitrary starting value $\boldsymbol{r}^{L+1,K=0}$, as $K \to \infty$. This type of contraction condition is sufficient, but not necessary, for convergence. Inserting this into $m\ddot{\boldsymbol{r}} = \boldsymbol{\Psi}(\boldsymbol{r})$ leads to

$$\boldsymbol{r}^{L+1,K} = \underbrace{\boldsymbol{r}^L + \boldsymbol{v}^L \Delta t + \frac{\phi(\Delta t)^2}{m}\left((1-\phi)\boldsymbol{\Psi}(\boldsymbol{r}^L)\right)}_{\mathcal{R}} + \underbrace{\frac{\phi(\Delta t)^2}{m}\left(\phi\boldsymbol{\Psi}(\boldsymbol{r}^{L+1,K-1})\right)}_{\mathcal{G}(\boldsymbol{r}^{L+1,K-1})}, \tag{B.8}$$

whose convergence is restricted by $\eta \propto \frac{(\phi\Delta t)^2}{m}$. Therefore, we see that the contraction constant of $\mathcal{G}$ is (1) directly dependent on the strength of the interaction forces, (2) inversely proportional to m and (3) directly proportional to $\phi\Delta t$. Therefore, if convergence is slow within a time step, the time step size, which is adjustable, can be reduced by an appropriate amount to increase the rate of convergence. Thus, decreasing the time step size improves the convergence, however, we want to simultaneously maximize the time-step sizes to decrease overall computing time, while still meeting an error tolerance on the numerical solution's accuracy. In order to achieve this goal, we follow an approach found in Zohdi [97]–[96] originally developed for continuum thermo-chemical multifield problems in which (1) one approximates

$$\eta^{L+1,K} \approx S(\Delta t)^p \tag{B.9}$$

(S is a constant) and (2) one assumes that the error within an iteration to behave according to

$$(S(\Delta t)^p)^K \epsilon^{L+1,0} = \epsilon^{L+1,K}, \tag{B.10}$$

$K = 1, 2, \ldots$, where $\epsilon^{L+1,0}$ is the initial norm of the iterative error and S is intrinsic to the system.[8] Our goal is to meet an error tolerance in exactly a preset number of iterations. To this end, one writes

$$(S(\Delta t_{\text{tol}})^p)^{K_d} \epsilon^{L+1,0} = TOL, \tag{B.11}$$

where TOL is a tolerance and where K_d is the number of desired iterations.[9] If the error tolerance is not met in the desired number of iterations, the contraction constant $\eta^{L+1,K}$ is too large. Accordingly, one can solve for a new smaller step size, under the assumption that S is constant,

$$\Delta t_{\text{tol}} = \Delta t \left(\frac{\left(\frac{TOL}{\epsilon^{L+1,0}}\right)^{\frac{1}{pK_d}}}{\left(\frac{\epsilon^{L+1,K}}{\epsilon^{L+1,0}}\right)^{\frac{1}{pK}}} \right). \tag{B.12}$$

[8] For the class of problems under consideration, due to the quadratic dependency on Δt, $p \approx 2$.

[9] Typically, K_d is chosen to be between five to ten iterations.

The assumption that S is constant is not critical, since the time steps are to be recursively refined and unrefined throughout the simulation. Clearly, the expression in Eq. (B.12) can also be used for time step enlargement, if convergence is met in less than K_d iterations.[10] An implementation of the procedure is as follows:

$$
\begin{array}{l}
(1)\,\text{GLOBAL FIXED – POINT ITERATION : (SET i = 1 AND K = 0) :}\\
(2)\,\text{IF } i > N_p \text{ THEN GO TO (4)}\\
(3)\,\text{IF } i \leq N_p \text{ THEN :}\\
\quad (a)\ \text{COMPUTE POSITION :}\boldsymbol{r}_i^{L+1,K}\\
\quad (b)\ \text{GO TO (2) FOR NEXT PARTICLE } (i = i+1)\\
(4)\,\text{ERROR MEASURE :}\\
\quad (a)\epsilon_K \stackrel{\text{def}}{=} \dfrac{\sum_{i=1}^{N_p} \|\boldsymbol{r}_i^{L+1,K} - \boldsymbol{r}_i^{L+1,K-1}\|}{\sum_{i=1}^{N_p} \|\boldsymbol{r}_i^{L+1,K} - \boldsymbol{r}_i^{L}\|} \qquad \text{(normalized)}\\
\quad (b)Z_K \stackrel{\text{def}}{=} \dfrac{\epsilon_K}{TOL_r}\\
\quad (c)\boldsymbol{\Phi}_K \stackrel{\text{def}}{=} \left(\dfrac{(\frac{TOL}{\epsilon_0})^{\frac{1}{pK_d}}}{(\frac{\epsilon_K}{\epsilon_0})^{\frac{1}{pK}}}\right)\\
(5)\,\underline{\text{IF TOLERANCE MET } (Z_K \leq 1) \text{ AND } K < K_d \text{ THEN :}}\\
\quad (a)\,\text{INCREMENT TIME : } t = t + \Delta t\\
\quad (b)\,\text{CONSTRUCT NEW TIME STEP : } \Delta t = \boldsymbol{\Phi}_K \Delta t,\\
\quad (c)\,\text{SELECT MINIMUM : } \Delta t = MIN(\Delta t^{lim}, \Delta t) \text{ AND GO TO (1)}\\
(6)\,\underline{\text{IF TOLERANCE NOT MET } (Z_K > 1) \text{ AND } K = K_d \text{ THEN :}}\\
\quad (a)\,\text{CONSTRUCT NEW TIME STEP :}\Delta t = \boldsymbol{\Phi}_K \Delta t\\
\quad (b)\,\text{RESTART AT TIME = t AND GO TO (1)}
\end{array}
\tag{B.13}
$$

Generally speaking, the iterative error, which is a function of the time step size, is temporally variable and can become stronger, weaker, or possibly oscillatory, is extremely difficult to ascertain a-priori as a function of the time step size. Therefore, to circumvent this problem, the adaptive strategy presented in this section was developed to provide accurate solutions by iteratively adjusting the time steps. Specifically, a sufficient condition for the convergence of the presented fixed-point scheme was that the spectral radius or contraction constant of the coupled operator, which depends on the time step size, must be less than unity. This observation was used to adaptively maximize the time step sizes, while simultaneously controlling the coupled operator's spectral radius, in order to deliver solutions below an error tolerance within a prespecified number of desired iterations. This recursive staggering error control can allow for substantial reduction of computational effort by the adaptive use of large time steps. Furthermore, such a recursive process has a reduced sensitivity, relative to an explicit staggering approach, to the order in which the individual equations are solved, since it is self-correcting.

Remark-Iterative Solutions: With regard to the solution process, a recursive iterative scheme of the Jacobi-type, where the updates are made only after one complete system iteration, was illustrated in the derivations only for algebraic simplicity. The Jacobi method is easier to address theoretically, while the Gauss–Seidel type method, which involves immediately using the most current values, when they become available, is usually used at the implementation level. As is well-known, under relatively general conditions, if the Jacobi method converges, the Gauss–Seidel method converges at a faster rate, while if the Jacobi method diverges, the Gauss–Seidel method diverges at a faster rate (for example, see Ames [155] or Axelsson [156]). It is important to realize that the Jacobi method is perfectly parallelizable. In other words, the calculation for each particle are uncoupled, with the updates only coming afterwards. Gauss–Seidel, since it requires the most current updates, couples the particle calculations immediately. However, these methods can be combined to create hybrid approaches, whereby the entire particulate flow is partitioned into groups and within each group a Gauss–Seidel method is applied. In other words, for a group, the positions of any particles from outside are initially frozen, as far as calculations involving members of the group are concerned. After each isolated group's solution (particle positions) has converged, computed in parallel, then all positions are updated, i.e. the most current positions become available to all members of the swarm, and the isolated group calculations are repeated. Classical solution methods require $\mathcal{O}(N^3)$ operations, whereas iterative schemes, such as the one presented, typically require order N^q, where $1 \leq q \leq 2$. For details see Axelsson [156]. Also, such solvers are highly advantageous since solutions to previous time steps can be used as the first guess to accelerate the solution procedure.

Appendix C. Incorporation of reduced-order models

Artificial Neural Networks (ANN) have received huge attention in the scientific community over the last decade and are based on layered input–output type frameworks that are essentially adaptive nonlinear regressions of the form $\mathcal{O} = \mathcal{B}(\boldsymbol{I}, \boldsymbol{w})$, where $\mathcal{O}$ is a

[10] Time-step size adaptivity is important, since the system's dynamics can dramatically change over the course of time, possibly requiring quite different time step sizes to control the iterative error. However, to maintain the accuracy of the time-stepping scheme, one must respect an upper bound dictated by the discretization error, i.e., $\Delta t \leq \Delta t^{lim}$.

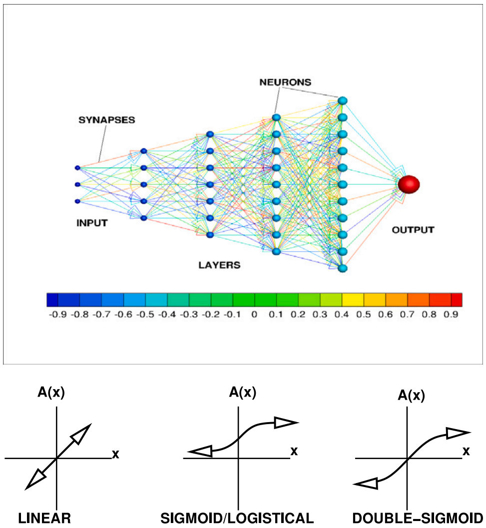

Fig. 20. Top: An ANN comprised of (1) Five layers (one input layer and four hidden layers) (2) 35 activation neurons (3+5+7+9+11) and (3) 223 weighted synapses. The color-coding represents the value of the weights. Bottom: Various neuron activation functions: (1) Linear (2) Sigmoid and (3) Double Sigmoid.

desired output and $\mathcal{B}$ is the ANN. This is discussed next and follows a framework developed in Zohdi [157]. Specifically, ANN are based on layered input–output type frameworks that are essentially adaptive nonlinear regressions of the form

$$\mathcal{O} = \mathcal{B}(I_1, I_2, \ldots, I_M, w_1, w_2, \ldots, w_N), \tag{C.1}$$

where $\mathcal{O}$ is a desired output and $\mathcal{B}$ is the ANN comprised of:

- **Synapses**, which multiply inputs (I_i, $i = 1, 2, \ldots, M$) by weights (w_i, $i = 1, 2, \ldots, N$) that represent the input relevance to the desired output,
- **Neurons**, which aggregate outputs from all incoming synapses and apply activation functions to process the data and
- **Training**, which calibrates the weights to match a desired overall output.

For example, Fig. 20 illustrates a detailed ANN comprised of (1) Five layers (one input layer and four hidden layers) (2) 35 activation neurons (3+5+7+9+11) and (3) 223 weighted synapses. *The primary issue with ANNs is the calibration or "training" of the synapse weights.*

The key components of an ANN can be summarized as follows (which is centered around training):

- STEP 1: Guess a set of trial weights, given by the vector $\boldsymbol{w}^{i=1}$, for the synapses and insert into the ANN (detailed construction shown shortly)

$$\mathcal{B}(\boldsymbol{I}, \boldsymbol{w}^i) = \mathcal{O}^i, \tag{C.2}$$

  which produces an overall trial output.
- STEP 2: Compute the error:

$$\mathcal{E}^i \stackrel{\text{def}}{=} \|\mathcal{O}^{desired} - \mathcal{O}^i\|, \tag{C.3}$$

  where $\mathcal{O}^{desired}$ is the desired output, which could come from experimental/field data or results from a complex computational model of a system, where a reduced complexity ANN may be useful to represent the system.
- STEP 3: The minimization of the error by adjusting the weights:

$$\boldsymbol{w}^{i+1} = \boldsymbol{w}^i + \Delta\boldsymbol{w}^{i+1} \tag{C.4}$$

- STEP 4: Repeat Steps 1–3 until the best set of weights are found to minimize the error.

The determination of the synapse weights can be cast as a nonconvex optimization problem, whereby the cost/error function represents the normed difference between observed data and the output of the ANN for a selected set of weights. The objective is to select a set of weight which minimizes the cost/error. One family of methods that are extremely well-suited for this process are genetic-based machine-learning algorithms. There are a variety of approaches to minimize the error, for example by utilizing the genetic-based machine-learning algorithm (MLA) introduced earlier, which is well-suited for nonconvex optimization. This proceeds by minimizing Π, by varying the design parameters, $\boldsymbol{\Lambda}^i \stackrel{\text{def}}{=} \{\Lambda^i_1, \Lambda^i_2, \Lambda^i_3, \ldots, \Lambda^i_N\}$, where the search is conducted within the constrained ranges of $\Lambda_1^{(-)} \leq \Lambda_1 \leq \Lambda_1^{(+)}$, $\Lambda_2^{(-)} \leq \Lambda_2 \leq \Lambda_2^{(+)}$, $\Lambda_3^{(-)} \leq \Lambda_3 \leq \Lambda_3^{(+)}$, etc. These upper and lower limits are dictated by what is physically feasible.

Data availability

No data was used for the research described in the article.

References

[1] Brian P. Tice, Unmanned aerial vehicles – The force multiplier of the 1990s, Airpower J. (1991) When used, UAVs should generally perform missions characterized by the three Ds: dull, dirty, and dangerous.

[2] Ed. Alvarado, 237 Ways Drone Applications Revolutionize Business, Drone Industry Insights, 2021.

[3] F. Rekabi-Bana, J. Hu, T. Krajník, F. Arvin, Unified robust path planning and optimal trajectory generation for efficient 3D area coverage of quadrotor UAVs, IEEE Trans. Intell. Transp. Syst. (2023).

[4] J. Hu, H. Niu, J. Carrasco, B. Lennox, F. Arvin, Fault-tolerant cooperative navigation of networked uav swarms for forest fire monitoring, in: Aerospace Science and Technology, in: Remote sensing of the environment using unmanned aerial systems (UAS). [S.l.], vol. 2022 (2022) 2023. OCLC 1329422815,

[5] Matthew T. Perks, Dal Sasso, Silvano Fortunato, Alexandre Hauet, Elizabeth Jamieson, Jérôme Le Coz, Sophie Pearce, Salvador Peña Haro, Alonso Pizarro, Dariia Strelnikova, Flavia Tauro, James Bomhof, Salvatore Grimaldi, Alain Goulet, Borbála Hortobágyi, Magali Jodeau, Towards harmonisation of image velocimetry techniques for river surface velocity observations, Earth Syst. Sci. Data (ISSN: 1866-3516) 12 (3) (2020) 1545–1559, http://dx.doi.org/10.5194/essd-12-1545-2020, Bibcode:2020ESSD.12.1545P.

[6] Cengiz Koparan, A. Bulent Koc, Charles V. Privette, Calvin B. Sawyer, Adaptive water sampling device for aerial robots, Drones (ISSN: 2504-446X) 4 (1) (2020) 5, http://dx.doi.org/10.3390/drones4010005.

[7] Cengiz Koparan, Ali Bulent Koc, Charles V. Privette, Calvin B. Sawyer, Julia L. Sharp, Evaluation of a UAV-assisted autonomous water sampling, Water 10 (5) (2018) 655, http://dx.doi.org/10.3390/w10050655.

[8] Cengiz Koparan, Ali Bulent Koc, Charles V. Privette, Calvin B. Sawyer, In situ water quality measurements using an Unmanned Aerial Vehicle (UAV) system, Water 10 (3) (2018) 264, http://dx.doi.org/10.3390/w10030264.

[9] Cengiz Koparan, Ali Bulent Koc, Charles V. Privette, Calvin B. Sawyer, Autonomous in situ measurements of noncontaminant water quality indicators and sample collection with a UAV, Water 11 (3) (2019) 604, http://dx.doi.org/10.3390/w11030604.

[10] Leslie Cary, James Coyne, ICAO unmanned aircraft systems (UAS), in: Circular, vol. 328, 2011–2012 UAS Yearbook – UAS: The Global Perspective (PDF), Blyenburgh & Co, 2016, pp. 112–115.

[11] Hu J., A. Lanzon, An innovative tri-rotor drone and associated distributed aerial drone swarm control, Robot. Auton. Syst. 103 (2018) 162–174, http://dx.doi.org/10.1016/j.robot.2018.02.019.

[12] A. Garrow Laurie, Brian J. German, Caroline E. Leonard, Urban air mobility: A comprehensive review and comparative analysis with autonomous and electric ground transportation for informing future research, Transp. Res. Part C: Emerg. Technol. (ISSN: 0968-090X) 132 (2021) 103377, http://dx.doi.org/10.1016/j.trc.2021.103377, Bibcode:2021TRPC..13203377G.

[13] Caizhi Zhang, Yuqi Qiu, Jiawei Chen, Yuehua Li, Zhitao Liu, Yang Liu, Jiujun Zhang, Chan Siew Hwa, A comprehensive review of electrochemical hybrid power supply systems and intelligent energy managements for unmanned aerial vehicles in public services, Energy AI 9 (2022) 100175, http://dx.doi.org/10.1016/j.egyai.2022.100175, hdl:10356/164036. Bibcode:2022EneAI...900175Z, 2666-5468.

[14] Dario Floreano, Robert J. Wood, Science, technology and the future of small autonomous drones, Nat. 521 (7553) (2015) 460–466, http://dx.doi.org/10.1038/nature14542, Bibcode:2015Natur.521..460F, PMID 26017445. S2CID 4463263.

[15] Giancarmine Fasano, Domenico Accardo, Anna Elena Tirri, Antonio Moccia, Ettore De Lellis, Radar/electro-optical data fusion for non-cooperative UAS sense and avoid, Aerosp. Sci. Technol. 46 (2015) 436–450, http://dx.doi.org/10.1016/j.ast.2015.08.010, Bibcode:2015AeST.46.436F.

[16] Salvatore Manfreda, Matthew McCabe, Pauline Miller, Richard Lucas, Victor Pajuelo Madrigal, Giorgos Mallinis, Eyal Ben Dor, David Helman, Lyndon Estes, Giuseppe Ciraolo, Jana Müllerová, Flavia Tauro, M. de Lima, João de Lima, Antonino Maltese, On the use of unmanned aerial systems for environmental monitoring, Remote. Sens. (ISSN: 2072-4292) 10 (4) (2018) 641, http://dx.doi.org/10.3390/rs10040641, Bibcode:2018RemS...10..641M. hdl:10251/127481.

[17] Daniel F. Carlson, Søren Rysgaard, Adapting open-source drone autopilots for real-time iceberg observations, MethodsX (ISSN: 2215-0161) 5 (2018) 1059–1072, http://dx.doi.org/10.1016/j.mex.2018.09.003, PMC 6139390. PMID 30225206.

[18] J. Lesko, M. Schreiner, D. Megyesi, Levente Kovacs, Pixhawk PX-4 autopilot in control of a small unmanned airplane, in: 2019 Modern Safety Technologies in Transportation, MOSATT. Kosice, Slovakia, IEEE, ISBN: 978-1-7281-5083-3, 2019, pp. 90–93, http://dx.doi.org/10.1109/MOSATT48908.2019.8944101, S2CID 209695691.

[19] Pierre-Jean Bristeau, François Callou, David Vissière, Nicolas Petit, The navigation and control technology inside the AR.Drone micro UAV, in: IFAC World Congress, 2011.

[20] Evgenii Vinogradov, A.V.S. Kumar, Sai Bhargav, Franco Minucci, Sofie Pollin, Enrico Natalizio, Remote ID for separation provision and multi-agent navigation, in: 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference, DASC, ISBN: 979-8-3503-3357-2, 2023, pp. 1–10, http://dx.doi.org/10.1109/DASC58513.2023.10311133, arXiv:2309.00843.

[21] Manohari Balasingam, Drones in medicine-the rise of the machines, Int. J. Clin. Pr. 71 (9) (2017) http://dx.doi.org/10.1111/ijcp.12989, PMID 28851081, e12989.

[22] Anna M. Johnson, Christopher J. Cunningham, Evan Arnold, Wayne D. Rosamond, Jessica K. Zègre-Hemsey, Impact of using drones in emergency medicine: What does the future hold? Open Access Emerg. Med. 13 (2021) 487–498, http://dx.doi.org/10.2147/OAEM.S247020, PMC 8605877. PMID 34815722.

[23] Kelsey D. Atherton, The marines are getting supersized drones for battlefield resupply, in: Popular Science, Keller, John (2023-12-13), 2023, Marine Corps orders 28 unmanned quadcopter aircraft for battlefield resupply in $11 million contract award. Military Aerospace.

[24] R.M. Dileep, V.A. Navaneeth, Savita Ullagaddi, Ajit Danti, A study and analysis on various types of agricultural drones and its applications, in: 2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN, IEEE, ISBN: 978-1-7281-8818-8, 2020, pp. 181–185, http://dx.doi.org/10.1109/ICRCICN50933.2020.9296195.

[25] Dmitry Nazarov, Anton Nazarov, Elena Kulikova, Drones in agriculture: Analysis of different countries, BIO Web Conf. (ISSN: 2117-4458) 67 (02029) (2023).

[26] Jiyang Liu, Yu Ding, Rui Qiu, Zhiyi Meng, Deguo Sun, Xinying Peng, Drone-assisted long-distance delivery of medical supplies with recharging stations in rural communities, Heal. Soc. Care Commun. (ISSN: 0966-0410) 2024 (1) (2024) http://dx.doi.org/10.1155/2024/9143099.

[27] C.M. Breder, Equations descriptive of fish schools and other animal aggregations, Ecology 35 (3) (1954) 361–370.

[28] G. Beni, The concept of cellular robotic system, in: IEEE International Symposium on Intelligent Control, 1988, pp. 57–62.
[29] R.A. Brooks, Intelligence without reason, in: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-91, 1991, pp. 569–595.
[30] G. Dudek, M. Jenkin, E. Milios, D. Wilkes, A taxonomy for multi-agent robotics, Auton. Robots 3 (1996) 375–397.
[31] Y.U. Cao, A.S. Fukunaga, A. Kahng, Cooperative mobile robotics: Antecedents and directions, Auton. Robots 4 (1) (1997) 7–27.
[32] Y. Liu, K.M. Passino, Swarm Intelligence: Literature Overview, Technical Report, Ohio State University, 2000.
[33] M. Turpin, N. Michael, V. Kumar, Capt: Concurrent assignment and planning of trajectories for multiple robots, Int. J. Robot. Res. (2014) 2014, https://journals.sagepub.com/doi/10.1177/0278364913515307.
[34] T.I. Zohdi, Computational design of swarms, Int. J. Numer. Methods Eng. 57 (2003) 2205–2219.
[35] V. Gazi, K.M. Passino, Stability analysis of swarms, in: Proceedings of the American Control Conference. Anchorage, AK May 8-10, 2002.
[36] J. Bender, R. Fenton, On the flow capacity of automated highways, Transp. Sci. 4 (1970) 52–63.
[37] J. Kennedy, R. Eberhart, Swarm Intelligence, Morgan Kaufmann Publishers, 2001.
[38] T.I. Zohdi, Mechanistic modeling of swarms, Comput. Methods Appl. Mech. Engrg. 198 (21–26) (2009) 2039–2051.
[39] T.I. Zohdi, An agent-based computational framework for simulation of competing hostile planet-wide populations, Comput. Methods Appl. Mech. Engrg. (2017) http://dx.doi.org/10.1016/j.cma.2016.04.032.
[40] Zohdi T. I., Multiple UAVs for mapping: a review of basic modeling, simulation and applications, Annu. Rev. Environ. Resour. (2018) http://dx.doi.org/10.1146/annurev-environ-102017-025912.
[41] T.I. Zohdi, The game of drones: rapid agent-based machine-learning models for multi-UAV path planning, Comput. Mech. (2019) http://dx.doi.org/10.1007/s00466-019-01761-9.
[42] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: from Natural to Artificial Systems, Oxford University Press, New York, 1999.
[43] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, Syst. Man Cybern. Part B, IEEE Trans. on 26 (1) (1996) 29–41.
[44] E. Bonabeau, C. Meyer, Swarm intelligence: A whole new way to think about business, Harv. Bus. Rev. 79 (5) (2001) 106–114.
[45] E. Fiorelli, N.E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, D.M. Fratantoni, Multi-auv control and adaptive sampling in monterey bay, in: Autonomous Underwater Vehicles, 2004 IEEE/OES, 2004, pp. 134–147.
[46] T. Feder, Statistical physics is for the birds, Phys. Today (2007) 28–29.
[47] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, V. Zdravkovic, Interaction ruling animal collective behavior depends on topological rather than metric distance: evidence from a field study, PNAS 105 (4) (2008) 1232–1237.
[48] J. Ring, The Laser in Astronomy, New Scientist, 1963, pp. 672–673.
[49] A.P. Cracknell, L. Hayes, Introduction to Remote Sensing, second ed., Taylor and Francis, London, ISBN: 0-8493-9255-1, 2007, OCLC 70765252.
[50] G.G. Goyer, R. Watson, The laser and its application to meteorology, Bull. Am. Meteorol. Soc. 44 (9) (1963) 564–575, [568].
[51] A. Medina, F. Gaya, F. Pozo, Compact laser radar and three-dimensional camera, J. Opt. Soc. Amer. A 23 (2006) 800–805.
[52] P. Trickey, X. Cao, Characterization of the OPAL obscurant penetrating LiDAR in various degraded visual environments, in: Proc. SPIE 8737, Degraded Visual Environments: Enhanced, Synthetic, and External Vision Solutions, vol. 2013, 2013, p. 87370E, http://dx.doi.org/10.1117/12.2015259, 2013.
[53] M. Hansard, S. Lee, O. Choi, R. Horaud, Time-of-flight cameras: Principles, methods and applications, SpringerBriefs Comput. Sci. (2012) http://dx.doi.org/10.1007/978-1-4471-4658-2,
[54] S. Schuon, C. Theobalt, J. Davis, S. Thrun, High-quality scanning using time-of-flight depth superresolution, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008, Institute of Electrical and Electronics Engineers, 2008, pp. 1–7.
[55] S.B. Gokturk, H. Yalcin, C. Bamji, A time-of-flight depth sensor - System description, issues and solutions, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2004, Institute of Electrical and Electronics Engineers, 2005, pp. 35–45, http://dx.doi.org/10.1109/CVPR.2004.291.
[56] ASC's 3D Flash LiDAR camera selected for OSIRIS-REx asteroid mission. NASASpaceFlight.com. 2012-05-13.
[57] Jan Aue, Dirk Langer, Bernhard Muller-Bessler, Burkhard Huhnke, Efficient segmentation of 3D LiDAR point clouds handling partial occlusion, in: 2011 IEEE Intelligent Vehicles Symposium, IV, IEEE, Baden-Baden, Germany, ISBN: 978-1-4577-0890-9, 2011, http://dx.doi.org/10.1109/ivs.2011.5940442.
[58] S. Hsu, S. Acharya, A. Rafii, R. New, Performance of a time-of-flight range camera for intelligent vehicle safety applications, in: Advanced Microsystems for Automotive Applications 2006, Springer, VDI-Buch, ISBN: 978-3-540-33410-1, 2006, pp. 205–219, http://dx.doi.org/10.1007/3-540-33410-6-16, Archived from the original (pdf) on 2006-12-06.
[59] O. Elkhalili, O.M. Schrey, W. Ulfig, W. Brockherde, Hosticka B.J., A 64x8 pixel 3-D CMOS time-of flight image sensor for car safety applications, Eur. Solid State Circuits Conf. 2006 (2006) 568–571, http://dx.doi.org/10.1109/ESSCIR.2006.307488,
[60] T.I. Zohdi, Rapid simulation-based uncertainty quantification of flash-type time-of-flight and LiDAR-based body-scanning processes, Comput. Methods Appl. Mech. Engrg. (2019) http://dx.doi.org/10.1016/j.cma.2019.03.056.
[61] T.I. Zohdi, An agent-based computational framework for simulation of global pandemic and social response on planet X, Comput. Mech. (2020) http://dx.doi.org/10.1007/s00466-020-01886-2.
[62] T.I. Zohdi, A digital twin framework for machine learning optimization of aerial fire fighting and pilot safety, Comput. Methods Appl. Mech. Eng. 373 (2021) 12021, 113446.
[63] T.I. Zohdi, A digital-twin and machine-learning framework for precise heat and energy management of data-centers, Comput. Mech. (2022) (2022) http://dx.doi.org/10.1007/s00466-022-02152-3.
[64] T.I. Zohdi, An adaptive digital framework for energy management of complex multi-device systems, Comput. Mech. (2022) http://dx.doi.org/10.1007/s00466-022-02212-8.
[65] T.I. Zohdi, A machine-learning framework for the simulation of nuclear deflection of Planet-Killer-Asteroids, Comput. Methods Appl. Mech. Eng. (2022) http://dx.doi.org/10.1016/j.cma.2022.115316.
[66] T.I. Zohdi, A machine-learning digital-twin for rapid large-scale solar-thermal energy system design, Comput. Methods Appl. Mech. Engrg. (2023) 115991, http://dx.doi.org/10.1016/j.cma.2023.115991.
[67] T.I. Zohdi, A voxel-based machine-learning framework for thermo-fluidic identification of unknown objects, Comput. Methods Appl. Mech. Engrg. (2023) 116571, http://dx.doi.org/10.1016/j.cma.2023.116571.
[68] T.I. Zohdi, A machine-learning enabled digital-twin framework for the rapid design of satellite constellations for Planet-X, Comput. Mech. (2024) http://dx.doi.org/10.1007/s00466-024-02467-3.
[69] T.I. Zohdi, A digital-twin for rapid simulation of modular direct air capture systems, Int. J. Eng. Sci. 203 (2024) 104120.
[70] T.I. Zohdi, A machine-learning enabled digital-twin framework for next generation precision agriculture and forestry, Comput. Methods Appl. Mech. Engrg. (2024) 117250, http://dx.doi.org/10.1016/j.cma.2024.117250.
[71] T.I. Zohdi, A voxel-based machine-learning digital-oven-twin for precise cooking, Comput. Mech. (2024) http://dx.doi.org/10.1007/s00466-024-02575-0.
[72] T.I. Zohdi, Genetic design of solids possessing a random-particulate microstructure, Philos. Trans. R. Soc.: Math. Phys. Eng. Sci. 361 (1806) (2003) 1021–1043.
[73] T.I. Zohdi, Dynamic thermomechanical modeling and simulation of the design of rapid free-form 3D printing processes with evolutionary machine learning, Comput. Methods Appl. Mech. Engrg. (2017) http://dx.doi.org/10.1016/j.cma.2017.11.030.

[74] T.I. Zohdi, Electrodynamic machine-learning-enhanced fault-tolerance of robotic free-form printing of complex mixtures, Comput. Mech. (2018) http://dx.doi.org/10.1007/s00466-018-1629-y.
[75] J.H. Holland, Adaptation in Natural & Artificial Systems. Ann Arbor, Mich, University of Michigan Press, 1975.
[76] J.H. Holland, J.H. Miller, Artificial adaptive agents in economic theory (PDF), Am. Econ. Rev. 81 (2) (1991) 365–371.
[77] D.E. Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley, 1989.
[78] L. Davis, Handbook of Genetic Algorithms, Thompson Computer Press, 1991.
[79] C. Onwubiko, Introduction to Engineering Design Optimization, Prentice Hall, 2000.
[80] D.E. Goldberg, K. Deb, Special issue on genetic algorithms, Comput. Methods Appl. Mech. Eng. 186 (2–4) (2000) 121–124.
[81] M.W. Mueller, R. D'Andrea, Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers, IEEE Int. Conf. Robot. Autom. (ICRA) 2014 (2014).
[82] M.W. Mueller, R. D'Andrea, Relaxed hover solutions for multicopters: application to algorithmic redundancy and novel vehicles, Int. J. Robot. Res. 35 (8) (2015) 873–889.
[83] M.W. Mueller, M. Hehn, R. D'Andrea, A computationally efficient motion primitive for quadrocopter trajectory generation, IEEE Trans. Robot. 31 (8) (2015) 1294–1310.
[84] M. Hehn, R. Ritz, R.D. Andrea, Performance benchmarking of quadrotor systems using time-optimal control, Auton. Robots 33 (1–2) (2012) 69–88.
[85] B. Houska, H. Ferreau, M. Diehl, ACADO toolkit: An open source framework for automatic control and dynamic optimization, Optim. Control. Appl. Methods 32 (3) (2011) 298–312.
[86] A. Tagliabue, X. Wu, M.W. Mueller, Model-free online motion adaptation for optimal range and endurance of multicopters, in: IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, 2019.
[87] C. Holda, B. Ghalamchi, M.W. Mueller, Tilting multicopter rotors for increased power efficiency and yaw authority, in: International Conference on Unmanned Aerial Systems, ICUAS, IEEE, 2018, pp. 143–148.
[88] T.I. Zohdi, On the dynamics and breakup of quadcopters using a discrete element method framework, Comput. Methods Appl. Mech. Engrg. 327 (2017) 503–521.
[89] D. Powell, T.I. Zohdi, Attachment mode performance of network-modeled ballistic fabric shielding, Compos. Part B: Eng. 40 (6) (2009) 451–460.
[90] Wikipedia https://en.wikipedia.org/wiki/Propeller_theory.
[91] T.I. Zohdi, Charge-induced clustering in multifield particulate flow, Int. J. Numer. Methods Eng. 62 (7) (2005) 870–898.
[92] T.I. Zohdi, Computation of strongly coupled multifield interaction in particle-fluid systems, Comput. Methods Appl. Mech. Engrg. 196 (2007) 3927–3950.
[93] T.I. Zohdi, On the dynamics of charged electromagnetic particulate jets, Arch. Comput. Methods Eng. 17 (2) (2010) 109–135.
[94] T.I. Zohdi, Dynamics of Charged Particulate Systems. Modeling, Theory and Computation, Springer-Verlag, 2012.
[95] T.I. Zohdi, Numerical simulation of charged particulate cluster-droplet impact on electrified surfaces, J. Comput. Phys. 233 (2013) 509–526.
[96] T.I. Zohdi, A direct particle-based computational framework for electrically-enhanced thermo-mechanical sintering of powdered materials, Math. Mech. Solids (2014) 1–21, http://dx.doi.org/10.1007/s11831-013-9092-6.
[97] T.I. Zohdi, An adaptive-recursive staggering strategy for simulating multifield coupled processes in microheterogeneous solids, Int. J. Numer. Methods Eng. 53 (2002) 1511–1532.
[98] J. Duran, Sands, Powders and Grains. An Introduction to the Physics of Granular Matter, Springer Verlag, 1997.
[99] T. Pöschel, T. Schwager, Computational Granular Dynamics, Springer Verlag, 2004.
[100] E. Onate, S.R. Idelsohn, M.A. Celigueta, R. Rossi, Advances in the particle finite element method for the analysis of fluid-multibody interaction and bed erosion in free surface flows, Comput. Methods Appl. Mech. Engrg. 197 (19–20) (2008) 1777–1800.
[101] E. Onate, M.A. Celigueta, S.R. Idelsohn, F. Salazar, B. Suárez, Possibilities of the particle finite element method for fluid-soil–structure interaction problems, Comput. Mech. 48 (2011) 307–318.
[102] J. Rojek, C. Labra, O. Su, E. Onate, Comparative study of different discrete element models and evaluation of equivalent micromechanical parameters, Int. J. Solids Struct. 49 (2012) (2012) 1497–1517, http://dx.doi.org/10.1016/j.ijsolstr.2012.02.032.
[103] J.M. Carbonell, E. Onate, Suarez B., Modeling of ground excavation with the particle finite element method, J. Eng. Mech. ASCE 136 (2010) 455–463.
[104] C. Labra, Onate E., High-density sphere packing for discrete element method simulations, Commun. Numer. Methods Eng. 25 (7) (2009) 837–849.
[105] Zeyi Yang, Why China's Dominance in Commercial Drones Has Become a Global Security Matter, MIT Technology Review, 2024.
[106] Ishveena Singh, The Secret to DJI's Drone Market Dominance: Revealed, DroneDJ, 2024.
[107] Gina Chon, DJI Is a more Elusive U.S. Target than Huawei, Reuters, 2021.
[108] Brad Dress, China's Dominant Drone Industry Is a Step Ahead of Congress, The Hill, 2024.
[109] Whitepaper: AUVSI partnership for drone competitiveness, 2024, AUVSI Partnership for Drone Competitiveness.
[110] Paul Mozur, Valerie Hopkins, Ukraine's war of drones runs into an obstacle: China, New York Times, 2024, (2023; Hannah Beech & Paul Mozur, Drones Changed This Civil War, and Linked Rebels to the World, New York Times.
[111] Heather Somerville, Why first responders don't want the U.S. to ban chinese drones, Wall Str. J. (2024).
[112] Lars Schönander, Securing the Skies: Chinese Drones and U.S. Cybersecurity Risks, Foundation for American Innovation, 2023.
[113] Timeline of U.S. Federal Government Activity Identifying and Addressing Unsecure UAS, Association for Uncrewed Vehicle Systems International, 2024.
[114] Eric Holdeman, Federal Government Will Require Purchase of 'Made in America' Drones, Government Technology, 2024.
[115] American Security Drone Act Of 2023, General Services Administration (last accessed 2024); Gallagher, Colleagues Introduce Bipartisan American Security Drone Act, 2023, The Select Committee on the Chinese Communist Party.
[116] Jaron Schneider, U.S. department of the interior says anti-DJI regulation hurt its operations, 2024, PetaPixel.
[117] David Shepardson, US Considers Potential Rules to Restrict or Bar Chinese Drones, Reuters, 2025.
[118] Agence France Presse, 2024, U.S. Drone Maker Says China Sanctions to Hit Supply Chain, Barron's.
[119] Heather Somerville, American drone startup notches rare victory in Ukraine, Wall Str. J. (2024).
[120] Heather Somerville, Brett Forrest, How American drones failed to turn the tide in Ukraine, Str. J. (2024).
[121] Joyu Wang, Taiwan wants a drone army — but China makes the drones it wants, Wall Str. J. (2024).
[122] Chris Buckley, Amy Chang Chien, Taiwan and U.S. work to counter China's drone dominance, 2024, New York Times.
[123] Ed Alvarado, Commercial use of drone swarms, 2024, Drone Industry Insights.
[124] Zachary Kallenborn, Swarm Clouds on the Horizon? Exploring the Future of Drone Swarm Proliferation, Modern War Institute, 2024.
[125] Military and security developments involving the People's Republic of China 2024, 2024, U.S. Department of Defense.
[126] G.E. Elsinga, F. Scarano, B. Wieneke, B.W. van Oudheusden, Tomographic particle image velocimetry, Exp. Fluids 41 (15) (2006) 933–947.
[127] G.T. Herman, A. Lent, Iterative reconstruction algorithms, Comput. Biol. Med. 6 (1976) 273–294.
[128] A. Schroder, R. Geisler, G.E. Elsinga, F. Scarano, U. Dierksheide, Investigation of a turbulent spot and a tripped turbulent boundary layer flow using time-resolved tomographic PIV, Exp. Fluids 44 (2008) 305–316.
[129] B. Wieneke, Volume self-calibration for 3D particle image velocimetry. 549-556, Exp. Fluids 45 (2008) 549–556.
[130] Andres A. Aguirre-Pablo, Meshal K. Alarfaj, Er Qiang Li, J.F. Hernandez-Sanchez, Sigurdur T. Thoroddsen, Tomographic particle image velocimetry using smartphones and colored shadows, Sci. Rep. 7 (2017) 3714, http://dx.doi.org/10.1038/s41598-017-03722-9.
[131] C. Atkinson, J. Soria, An efficient simultaneous reconstruction technique for tomographic particle image velocimetry, Exp. Fluids 47 (2009) 553–568.

[132] S. Discetti, A. Ianiro, T. Astarita, G. Cardone, On a novel low cost high accuracy experimental setup for tomographic particle image velocimetry, Meas. Sci. Technol. 24 (2013) 075302.
[133] P. Geoghegan, N. Buchmann, J. Soria, M. Jermy, Time-resolved PIV measurements of the flow field in a stenosed, Compliant Arter. Model. Exp. Fluids 54 (2013) 1–19.
[134] C. Willert, B. Stasicki, J. Klinner, S. Moessner, Pulsed operation of high-power light emitting diodes for imaging flow velocimetry, Meas. Sci. Technol. 21 (2010) 075402.
[135] N.A. Buchmann, C.E. Willert, J. Soria, Pulsed, high-power LED illumination for tomographic particle image velocimetry, Exp. Fluids 53 (2012) 1545–1560.
[136] T.A. Casey, J. Sakakibara, S.T. Thoroddsen, Scanning tomographic particle image velocimetry applied to a turbulent jet, Phys. Fluids 25 (2013) 025102.
[137] W.-H. Tien, D. Dabiri, J.R. Hove, Color-coded three-dimensional micro particle tracking velocimetry and application to micro backward-facing step flows, Exp. Fluids 55 (2014) 1684.
[138] J. Xiong, et al., Rainbow particle imaging velocimetry for dense 3D fluid velocity imaging, ACM Trans. Graph. 36 (4) (2017) 36.
[139] T. Watamura, Y. Tasaka, Y. Murai, LCD-projector-based 3D color PTV, Exp. Therm. Fluid Sci. 47 (2013) 68–80.
[140] J. Klinner, C. Willert, Tomographic shadowgraphy for three-dimensional reconstruction of instantaneous spray distributions, Exp. Fluids 53 (2012) 531–543.
[141] M.J. McPhail, A.A. Fontaine, M.H. Krane, L. Goss, J. Crafton, Correcting for color crosstalk and chromatic aberration in multicolor particle shadow velocimetry, Meas. Sci. Technol. 26 (2015) 025302.
[142] C. Cierpka, R. Hain, N.A. Buchmann, Flow visualization by mobile phone cameras, Exp. Fluids 57 (2016) 1–10.
[143] F. Scarano, Tomographic PIV: principles and practice, Meas. Sci. Technol. 24 (2012) 012001.
[144] M. McPhail, M. Krane, A. Fontaine, L. Goss, J. Crafton, Multicolor particle shadow accelerometry, Meas. Sci. Technol. 26 (2015) 045301.
[145] Hicham Saaid, Jason Voorneveld, Christiaan Schinkel, Jos Westenberg, Frank Gijsen, Patrick Segers, Pascal Verdonck, Nico de Jong, Johan G. Bosch, Sasa Kenjeres, Tom Claessens, Tomographic PIV in a model of the left ventricle: 3D flow past biological and mechanical heart valves, J. Biomech. (ISSN: 0021-9290) 90 (2019) 40–49, http://dx.doi.org/10.1016/j.jbiomech.2019.04.024.
[146] H. Zhu, C. Wang, H. Wang, J. Wang, Tomographic PIV investigation on 3D wake structures for flow over a wall-mounted short cylinder, J. Fluid Mech. 831 (2017) 743–778, http://dx.doi.org/10.1017/jfm.2017.647.
[147] Kyle Patrick Lynch, Justin Wagner, High-speed tomographic PIV of cylinder wakes in a shock tube using a pulse-burst laser, 2019, United States. https://www.osti.gov/servlets/purl/1640645.
[148] Ning Liu, Yue Wu, Lin Ma, Quantification of tomographic PIV uncertainty using controlled experimental measurements, Appl. Opt. 57 (2018) 420–427, https://opg.optica.org/ao/abstract.cfm?URI=ao-57-3-420.
[149] Chuangxin He, Peng Wang, Yingzheng Liu, Lian Gan, Flow enhancement of tomographic particle image velocimetry measurements using sequential data assimilation, Phys. Fluids 34 (2022) 035101, http://dx.doi.org/10.1063/5.0082460.
[150] Jian Guo Liu, Philippa J. Mason, Essential Image Processing for GIS and Remote Sensing, Wiley-Blackwell, ISBN: 978-0-470-51032-2, 2009, p. 4.
[151] Alexander Chilton, The working principle and key applications of infrared sensors, 2013, AZoSensors.
[152] Tariq F., R. Haswell, P.D. Lee, D.W. McComb, Characterization of hierarchical pore structures in ceramics using multi-scale tomography, Acta Mater. 59 (5) (2011) 2109–2120, http://dx.doi.org/10.1016/j.actamat.2010.12.012.
[153] James D. Foley, Andries van Dam, John F. Hughes, Steven K. Feiner, Spatial-partitioning representations; surface detail, in: Computer Graphics: Principles and Practice, in: The Systems Programming Series, Addison-Wesley., ISBN: 0-201-12110-7, 1990.
[154] Sz. Chmielewski, P. Tompalski, Estimating outdoor advertising media visibility with voxel-based approach, Appl. Geogr. 87 (2017) 1–13, http://dx.doi.org/10.1016/j.apgeog.2017.07.007.
[155] W.F. Ames, Numerical Methods for Partial Differential Equations, second ed., Academic Press, 1977.
[156] O. Axelsson, Iterative Solution Methods, Cambridge University Press, 1994.
[157] T.I. Zohdi, A note on rapid genetic calibration of artificial neural networks, Comput. Mech. (2022) http://dx.doi.org/10.1007/s00466-022-02216-4.