# Range-based Relative Navigation for a Swarm of Centimetre-scale Femto-Spacecraft

Thomas Timmons*, James Beeley†, Gilles Bailet‡ and Colin R. McInnes§

*James Watt School of Engineering, University of Glasgow, Scotland, UK. G12 8QQ*

**In-orbit relative navigation between a networked swarm of centimetre-scale femto-spacecraft would add considerable value to a range of space mission concepts and applications, such as for multi-point sensing and distributed sparse aperture interferometry. For a swarm of networked femto-spacecraft, relative position determination would be possible by inferring coarse range estimates from the received signal strength indication associated with the communication link between swarm members. This is particularly advantageous for highly resource-constrained devices. In this paper, algorithms for swarm relative positioning using inter-spacecraft range estimates are presented that can be applied to centralised, decentralised and distributed network configurations. Relative navigation filters for initial relative orbit determination (IROD) and state estimation are presented for femto-spacecraft swarm deployment and dispersal scenarios. The algorithms presented could also find use in terrestrial applications, in static and dynamic wireless sensor networks.**

## Nomenclature

| | | |
|---|---|---|
| $A$ | = | system matrix |
| $\mathbf{a}_k$ | = | position of anchor spacecraft $k$ |
| $e_i$ | = | zero column vector with the value of 1 at point $i$ |
| $e_{ij}$ | = | zero column vector with the value of 1 at point $i$ and -1 at point $j$ |
| $\mathcal{F}$ | = | relative coordinate frame |
| $H$ | = | observation matrix |
| $h$ | = | observation function |
| $I$ | = | identity matrix |
| $J$ | = | Jacobian matrix |
| $K$ | = | Kalman gain |

*PhD Candidate, Space and Exploration Technology Group, t.timmons.1@research.gla.ac.uk

†Research Associate, Space and Exploration Technology Group

‡Research Associate, Space and Exploration Technology Group

§James Watt Chair, Professor of Engineering Science, Space and Exploration Technology Group

| | | |
|---|---|---|
| $k$ | = | discrete time step |
| $N$ | = | number of range estimates |
| $n$ | = | number of femto-spacecraft |
| $P$ | = | state covariance estimate |
| $Q$ | = | process noise covariance |
| $R$ | = | rotation matrix **or** sensor noise covariance |
| $R_o$ | = | orbital radius |
| $r_{ij}$ | = | true range between the $i^{\text{th}}$ and $j^{\text{th}}$ femto-spacecraft |
| $\hat{r}_{ij}$ | = | estimated range between the $i^{\text{th}}$ and $j^{\text{th}}$ femto-spacecraft |
| $\bar{r}$ | = | mean range of $N$ range estimates |
| $T$ | = | translation matrix |
| $T_o$ | = | orbital period |
| $t$ | = | time |
| $U, V$ | = | unitary matrices |
| $\mathbf{w}_k$ | = | process noise at discrete time step $k$ |
| $\mathbf{x}_i$ | = | position of the $i^{\text{th}}$ femto-spacecraft |
| $\hat{\mathbf{x}}_{\mathbf{k}}^-$ | = | a priori state estimate |
| $\hat{\mathbf{x}}_{\mathbf{k}}^+$ | = | a posteriori state estimate |
| $x, y, z$ | = | Cartesian elements of femto-spacecraft position |
| $\dot{x}, \dot{y}, \dot{z}$ | = | Cartesian elements of femto-spacecraft velocity |
| $\ddot{x}, \ddot{y}, \ddot{z}$ | = | Cartesian elements of femto-spacecraft acceleration |
| $X$ | = | matrix of $n$ femto-spacecraft positions $[\mathbf{x}_1, ..., \mathbf{x}_n]$ |
| $Z$ | = | symmetric positive semidefinite matrix |
| $\mathbf{z}_k$ | = | state measurement at discrete time step $k$ |
| $\alpha_{ij}$ | = | squared distance error between the $i^{\text{th}}$ and $j^{\text{th}}$ femto-spacecraft |
| $\mu$ | = | standard gravitational parameter of an orbital body |
| $\nu_{\mathbf{k}}$ | = | observation noise at discrete time step $k$ |
| $\sigma_r^2$ | = | mean squared error in range estimate |
| $\sigma_p^2$ | = | mean squared error in position |
| $\mathbf{\Phi}$ | = | state transition matrix |
| $\omega_n$ | = | mean motion of the femto-spacecraft deployer |

## I. Introduction

Due to continued miniaturisation in technology, it is now practical to develop active femto-spacecraft (mass under 100 g) with an inertial measurement unit (IMU), an attitude determination and control system (ADCS) and wireless radio frequency (RF) communications contained within a single printed circuit board (PCB) with a side length of only several centimetres. A wide range of potential applications are possible if many of these devices were to be deployed from a larger carrier spacecraft, such as a CubeSat, and dispersed into orbits neighbouring the carrier. This concept was pioneered by the KickSat project, which recently deployed more than one hundred 'ChipSats' from a 3U CubeSat [1]. Several research groups now have femto-spacecraft projects under development, including the Space and Exploration Technology Group at the University of Glasgow [2, 3]. A swarm of femto-spacecraft would render large-scale, simultaneous and spatially distributed measurements feasible for the improved investigation of planetary atmospheres, space weather monitoring, magnetospheric characterisation, gravity field mapping, distributed sparse aperture interferometry and other novel applications [3–8].

An underpinning motive behind the femto-spacecraft concept is to discover what functionality and new applications can be delivered at the smallest of spacecraft length-scales. Scaling the technology to a networked swarm of such devices dispersed over a large volume of space is a desirable extension to current capabilities. Not only would this enable compelling new applications for a range of mission scenarios, but would also enhance the capability of these resource-constrained devices as sensor platforms. As a swarm, objectives could be accomplished that are well beyond the capacity of any individual femto-spacecraft. Other motives for this approach include robustness and redundancy against failure with modular and dis-aggregated operations. Unlike traditional constellation and cluster architectures, a swarm could operate in an emergent way to achieve tasks beyond what its individual members are capable of, and do so without structure or hierarchy [9, 10]. Moreover, for centimetre-scale femto-spacecraft we can expect the number of devices in a swarm potentially to be extremely large. This utilisation essentially supposes a femto-spacecraft swarm as an ad-hoc space-based wireless sensor network (WSN).

Determining the location of femto-spacecraft relative to one another is essential in adding value to the data gathered in many mission applications and in enabling swarm members to operate in close proximity to each other. This can maximise the utility of each swarm member, not only for scientific investigation, but to also navigate in orbit without relying on an Earth-based ground station [11–13]. It is anticipated that absolute navigation would be enabled by the known position of the carrier spacecraft. Network structure and computation sharing would vary widely depending on the mission application, so we do not limit our discussion to one particular approach.

Concepts for the on-orbit relative navigation between at least two spacecraft can be broadly grouped into GPS-based, vision-based and RF-based approaches [10]. GPS-based approaches operate on the principle of differencing two absolute positions to obtain the relative positions of each satellite with respect to the other(s). Equipping femto-spacecraft with GPS transceivers is in principle feasible but it is power intensive and of course restricted to low Earth orbit [7, 14]. In expanding a swarm from dozens to thousands of femto-spacecraft there is a need to consider how relative localisation is achieved for the situation where few or no swarm members have access to GPS [15]. Recent work on vision-based methods for on-orbit relative navigation proposes using cameras and optical sensors for relative pose estimation via image processing and/or computer vision techniques for small satellites [16, 17]. Femto-spacecraft could be equipped with small COTS cameras, but the limited computational, ADCS and power resources required for relative state estimation makes approaches like this presently impractical.

Range-based relative-navigation methods have been implemented as part of a chain of available resources in a satellite's sensor suite, used to accompany or back-up other measurements, as was developed for the GRACE [18] and PRISMA [19] formation-flying satellites. These approaches only consider one-to-one communication for a pair of large satellites. With femto-spacecraft however we can consider a large number of limited devices using only range estimates for on-orbit relative navigation. This idea is discussed in [20], where the author presents how range data could be applied for the initial relative orbit determination problem (IROD) of small satellite formations, with no a-priori information on the formation's state. In this paper, we propose using swarm range estimates from communication within the network to calculate relative positions of swarm members directly. This would be of utility in the following circumstances:

1) Where there is an estimate of the a-priori swarm state from the known ejection impulse/time from a deployer spacecraft and a model of the relative dynamics of the swarm. Processing ranging estimates to determine swarm relative positioning could be used to improve the swarm's state estimate over time and bound growing uncertainties.
2) Where there is little or no understanding of the a-priori swarm state in a given scenario. Relative positioning information would characterise the dispersal of the swarm and the spatial density of the swarm. This information could then be post-processed by the deployer spacecraft to enhance the utility of sensed data from the swarm.

We propose that range-based relative positioning methods would utilise the wireless communication link between members to infer ranges, using the received signal strength indication (RSSI) as a proxy for a direct range measurement between two femto-spacecraft using omni-directional antennas. A signal sent from a device transmitting at a known power can be converted to a range estimate with an understanding of the path loss between the two devices. This is an appealing solution for femto-spacecraft as it would not require additional sensors when resources are already constrained, is usable in essentially any orbit scenario, and is available by virtue of the swarm carrying out its primary mission application when passing data packets between swarm members.

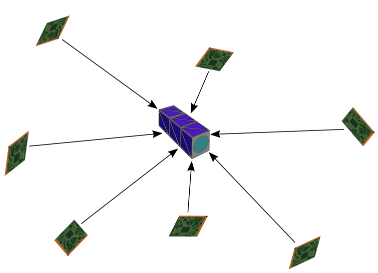
The rest of this paper is organised as follows. Section II explains the relative positioning problem and the development of two range-based relative positioning algorithms. Simulation-based results for the performance of both algorithms under the presence of varying levels of measurement noise are presented. Section III describes the relative dynamics and navigation filtering in scenarios where the swarm is deployed from a carrier spacecraft and disperses over several orbits. Simulation-based results of the relative navigation filtering are presented. Conclusions are given in Section IV.

## II. Range-based Relative Positioning Algorithms for a Femto-spacecraft Swarm

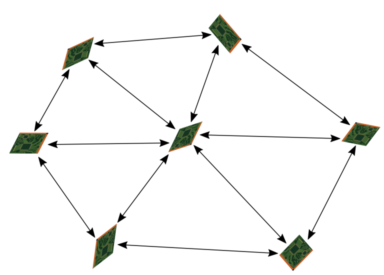
In this section we present two range-based relative positioning algorithms for a femto-spacecraft swarm; firstly a centralised and then a distributed algorithm designed to work in different network configurations. For this purpose, we suppose that a ranging metric is available from the RF communication link between networked femto-spacecraft in the form of RSSI data. We demonstrate that both algorithms work in simulation with a degree of random normally-distributed measurement noise added to the true ranges, representative of real-world inaccuracies that would be present. In simulation we assume the presence of a suitably accurate ranging metric that can be characterised in this way (e.g. via UHF omni-directional antennas). In free space, assuming isotropic radiation and the presence of no error sources, the Friis transmission formula [21] describes how signal strength decays with the square of the distance travelled, in simple terms allowing the range between a receiver and a transmitter to be calculated if the transmission power is known.

In practice, undirected antennas would exhibit some directive losses relative to the orientation between a receiving and transmitting device. While this would not render RSSI as a range estimate ineffective for the majority of relative orientations between two communicating devices in three-dimensional space, it would still be a source of error if alignment is unfavourable and if using this metric alone without any relative attitude knowledge. Were the relative attitude between two devices known (as would be possible from an on-board ADCS), this could in practice reduce the effect of this error by implementing a scale factor for particular relative attitude configurations between devices. Practical testing and implementation of undirected antennas in three-dimensional space, the potential of improved RSSI ranging with relative attitude knowledge, and the derivation of suitable path loss models for the scenarios described in this paper will be the subject of future work. For this paper we assume the ranging metric is present with a degree of measurement noise in order to develop the underlying methodology for both algorithms.

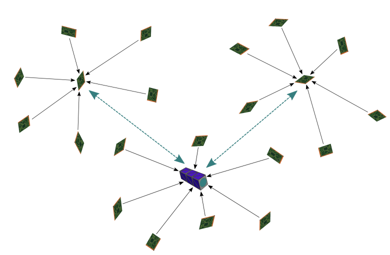
There are broadly three main network configurations for communication and computation, as shown in Fig. 1. The



**(a) Centralised**



**(b) Distributed**



**(c) Decentralised**

**Fig. 1 Network configurations**

arrowheads indicate communication between spacecraft for the purposes of localisation (sending range estimates to be processed). In a centralised configuration (Fig. 1a), femto-spacecraft send ranging data back to a central node (e.g. the swarm deployer) that handles all computation and optimises relative position estimates for the entire swarm. In a distributed configuration (Fig. 1b), this computation is shared; the femto-spacecraft share ranging information and attempt to localise relative to one another. In a decentralised configuration (Fig. 1c), distributed 'cluster heads' may act in a centralised way with nearby femto-spacecraft, but relate to other cluster heads in a distributed fashion. Different strategies suit various applications and swarm sizes. Developing both a centralised and a distributed algorithm, either of which may be more suited depending on the utility, scale and application, is therefore the approach taken. As presented, the algorithms developed in this paper may also be adapted for terrestrial applications. Such applications share the need for relative localisation in 2-D and 3-D with WSNs that share the high resource constraints and numbers of nodes that a femto-spacecraft swarm has, such as environments where GPS is unavailable.

### A. Centralised Relative Positioning Algorithm

The top-level challenge of this approach to femto-spacecraft swarm localisation is to find where individual members are located with only an estimate of their distances to one another. For WSNs this is commonly referred to as the sensor positioning problem [22]. The inputs to this problem are the unknown sensor positions (considering a femto-spacecraft being an individual sensor within the network, Fig. 2), the known sensor positions (if any, e.g. the deployer spacecraft), commonly referred to as 'anchors', and the estimated ranges between sensors. The outputs are the relative sensor positions. Applying this problem, we consider $n$ femto-spacecraft with position vectors $\mathbf{x}_1(x_1, y_1, z_1)$ to $\mathbf{x}_n(x_n, y_n, z_n)$, and $k$ anchor spacecraft with position vectors $\mathbf{a}_1(x_1, y_1, z_1)$ to $\mathbf{a}_k(x_k, y_k, z_k)$. The range between the $i^{\text{th}}$ and $j^{\text{th}}$ femto-spacecraft is given by $r_{ij}$, and the range between the $k^{\text{th}}$ anchor spacecraft and $j^{\text{th}}$ femto-spacecraft is given by $r_{kj}$. Arranging the femto-spacecraft position vectors into a $3 \times n$ matrix $X$, the objective is to find:

$$\mathbf{X} = \begin{bmatrix} x_1 & \dots & x_n \\ y_1 & \dots & y_n \\ z_1 & \dots & z_n \end{bmatrix} \tag{1}$$

subject to:

$$\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 = r_{ij}^2 \quad \& \quad \left\|\mathbf{a_k} - \mathbf{x}_j\right\|^2 = r_{kj}^2 \tag{2}$$

In practice, the objective of an algorithm that solves this problem is to minimise the difference between the true relative positions and noisy estimates. By relaxing the problem constraints to satisfy convex optimisation bounds, geometric constraints between femto-spacecraft can be represented by linear matrix inequalities (LMIs) combined to form a single semidefinite program (SDP), as shown in [23, 24]. A linear function is minimised in an SDP subject to the LMI constraint that a linear combination of symmetric matrices is positive semidefinite (the $n \times n$ symmetric real

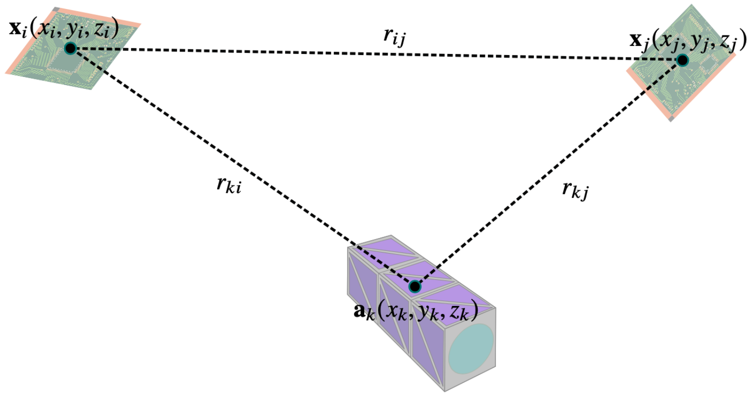



**Fig. 2 Relative positioning problem**

matrix $M$ is positive semidefinite if for a non-zero scalar $x$, $x^T Mx \geq 0$). The feasible regions of SDPs are spectahedra, and this requires the constraints to be convex functions. The sensor positioning problem is reformulated as finding the symmetric positive semidefinite matrix Z containing the matrix $X$ with all the femto-spacecraft positions. A solution that will minimise the sum of the errors matching the noisy distance estimates is found subject to most of the original constraints of the sensor positioning problem (non-convex constraints such as minimum range cannot be used). The formulation is then to find the semi-definite matrix:

$$Z = \begin{bmatrix} I_3 & X^T \\ X & X^T X \end{bmatrix} \tag{3}$$

to minimise:

$$\sum_{(i,j)\in N_1} (\alpha_{ij}^+ + \alpha_{ij}^-) + \sum_{(k,j)\in N_2} (\alpha_{jk}^+ + \alpha_{jk}^-) \tag{4}$$

subject to:

$$\begin{pmatrix} e_{ij} \\ 0 \end{pmatrix}^T Z \begin{pmatrix} e_{ij} \\ 0 \end{pmatrix} - \alpha_{ij}^+ + \alpha_{ij}^- = \hat{r_{ij}}^2 \quad \forall\ (i,j) \in N_1 \tag{5}$$

$$\begin{pmatrix} e_i \\ a_k \end{pmatrix}^T Z \begin{pmatrix} e_i \\ a_k \end{pmatrix} - \alpha_{ik}^+ + \alpha_{ik}^- = \hat{r_{ik}}^2 \quad \forall\ (i,k) \in N_2 \tag{6}$$

where:

$$\mathbf{Z}, \alpha_{ij}^+, \alpha_{ij}^-, \alpha_{jk}^+, \alpha_{jk}^- \succcurlyeq 0 \tag{7}$$

and where $\alpha_{ij} = \alpha_{ij}^+ + \alpha_{ij}^-$ and $\alpha_{jk} = \alpha_{jk}^+ + \alpha_{jk}^-$ are the errors in the ranging measurements. The positive and negative notation in these variables addresses the issue of purely using error magnitudes to account for over and under estimations. $I_3$ is a $3 \times 3$ identity matrix, $e_{ij}$ is a zero column vector with the value of 1 at point $i$ and the value of -1 at point $j$, and $e_i$ is a zero column vector with the value of 1 at point $i$. The measured range between femto-spacecraft $i$ and femto-spacecraft $j$ is given by $\hat{r_{ij}}$ , and $\hat{r_{jk}}$ is the measured range between femto-spacecraft $j$ and anchor $k$. The set $N_1$ contains the pairs of femto-spacecraft $(i, j)$ that have a range estimate $\hat{r_{ij}}$ between them. The set $N_2$ contains the pairs of femto-spacecraft $i$ and anchor $k$ that have a range estimate $\hat{r_{ik}}$ between them. Equation 7 indicates variables or matrices that are positive semidefinite.

This SDP problem formulation is the basis for the three-dimensional centralised positioning algorithm presented, implemented in MATLAB using the convex optimisation solver cvx [25]. This approach is necessarily 'centralised' because it would require one device to optimise for the entire swarm given all the range estimates and problem constraints. Such computation could be handled by the swarm deployer (e.g. a carrier CubeSat). In this role, the central unit would gather swarm range estimates communicated to it and use this information to form the SDP constraints, optimise and then extract the swarm relative positions. In practice, the space-based WSN that the femto-spacecraft forms would be dynamic, gradually drifting away from the deployer. The algorithm would therefore be sampled at regular intervals to update the overall swarm state estimates dynamically.

As will be discussed later in Section II.B, five femto-spacecraft can be localised in an arbitrary reference frame relative to one another to unambiguously describe their relative positions in three-dimensional space. This can provide the algorithm with a coordinate system if anchors are unavailable. Otherwise, anchors could be provided from the swarm deployment process, with the known ejection impulses and times used for initial state estimation, or by a small number of femto-spacecraft with GPS or other sensors.

We now quantify the algorithm's performance by generating a test case of 20 femto-spacecraft placed randomly in a cubic volume space, assigning four of these femto-spacecraft as anchors. The algorithm is then used with this same set of random positions with different error levels in the range estimates supplied to the solver. We assign four anchors for this test case as this is the minimum number of reference points found from testing that the algorithm can be expected to operate reliably with. Note here that while for this test anchors are points known in absolute space for investigating the algorithm's performance quantitatively, in practice these could be roles taken by any set of femto-spacecraft to transform a solution into a single relative frame.

The true ranges between femto-spacecraft are distorted with additive white Gaussian noise (AWGN), varying its standard deviation ($\sigma_r$) to examine how the algorithm performs. The mean squared error in range estimates, ${\sigma_r}^2$, is given by:

$$ {\sigma_r}^2 = \sum_{i=1}^{N} \frac{(\hat{r}_i - r_i)^2}{N} \tag{8} $$

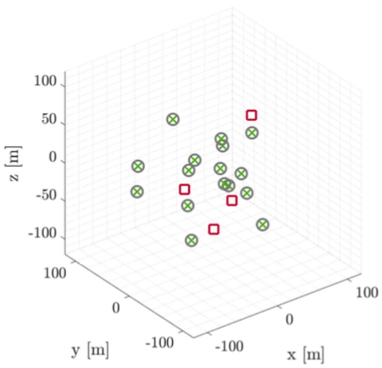
where $N$ is the number of range estimates, $\hat{r}_i$ is the range estimate supplied to the algorithm and $r_i$ is the true range. The algorithm is supplied a series of these range estimates between all femto-spacecraft. We vary $\sigma_r$ in simulation according to the average ranges between all the femto-spacecraft in the random network configuration, with the expectation that the ranging error in practice using an RSSI range estimate would be proportional to the magnitude of the range. In order to assess the performance of the algorithm in a systematic manner, we therefore vary $\sigma_r$ from 0-20% of $\bar{r}$, where $\bar{r}$ is the average range between all femto-spacecraft. For the example scenario presented in Fig. 3, $\bar{r} = 85.0\ m$ within a cubic volume of 200 $m^3$. We use the mean squared error of the localised femto-spacecraft (${\sigma_p}^2$) to compare the algorithm's localisation to the true positions, such that:

$$ {\sigma_p}^2 = \sum_{i=1}^{n} \frac{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2 + (\hat{y}_i - y_i)^2}{n} \tag{9} $$

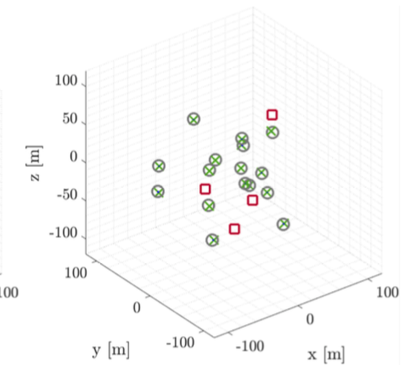
where $n$ is the number of femto-spacecraft, position $P_i$ has true coordinates $(x_i, y_i, z_i)$ and the algorithm estimates these coordinates to be $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$. Table 1 displays the algorithm performance for the scenario described above. The results are presented in Fig. 3.

**Table 1 Centralised algorithm performance in random scenario**

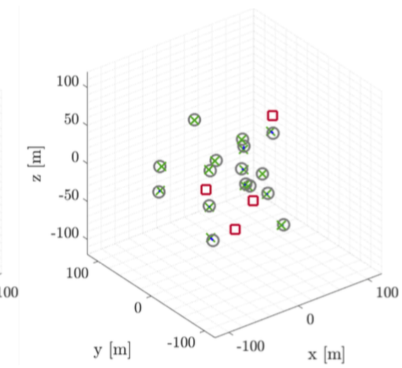
| $\sigma_r$ | $\sigma_p$ [m] |
|---|---|
| 0 | 0.01 |
| $0.01\bar{r}$ (0.92 m) | 1.42 |
| $0.02\bar{r}$ (1.84 m) | 3.01 |
| $0.03\bar{r}$ (2.75 m) | 2.69 |
| $0.04\bar{r}$ (3.67 m) | 8.77 |
| $0.05\bar{r}$ (4.59 m) | 9.43 |
| $0.06\bar{r}$ (5.51 m) | 11.89 |
| $0.07\bar{r}$ (6.43 m) | 14.78 |
| $0.08\bar{r}$ (7.34 m) | 15.33 |
| $0.09\bar{r}$ (8.26 m) | 17.73 |
| $0.10\bar{r}$ (9.18 m) | 16.03 |
| $0.20\bar{r}$ (18.36 m) | 30.08 |



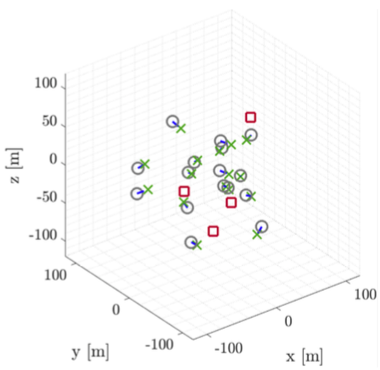
**(a)** $\sigma_r = 0$



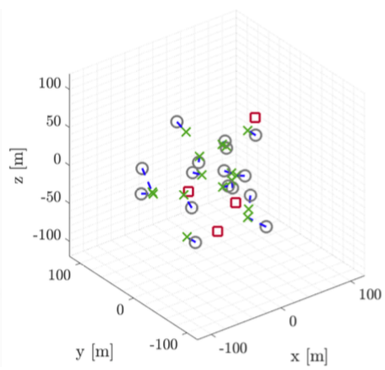
**(b)** $\sigma_r = 0.01\bar{r}$



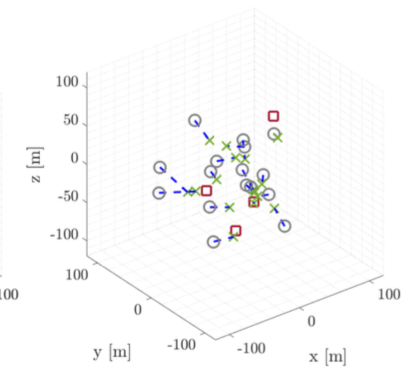
**(c)** $\sigma_r = 0.03\bar{r}$



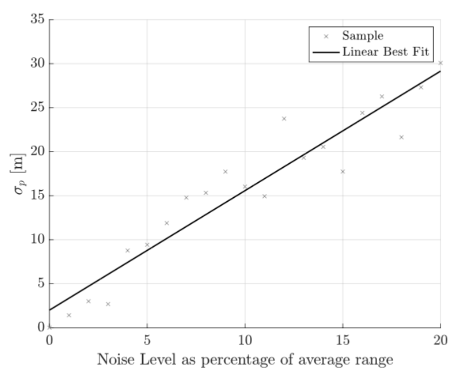
**(d)** $\sigma_r = 0.05\bar{r}$



**(e)** $\sigma_r = 0.10\bar{r}$



**(f)** $\sigma_r = 0.20\bar{r}$




**(g)** $\sigma_p$ **as noise levels increase**

**Fig. 3 Centralised algorithm performance in random scenario**

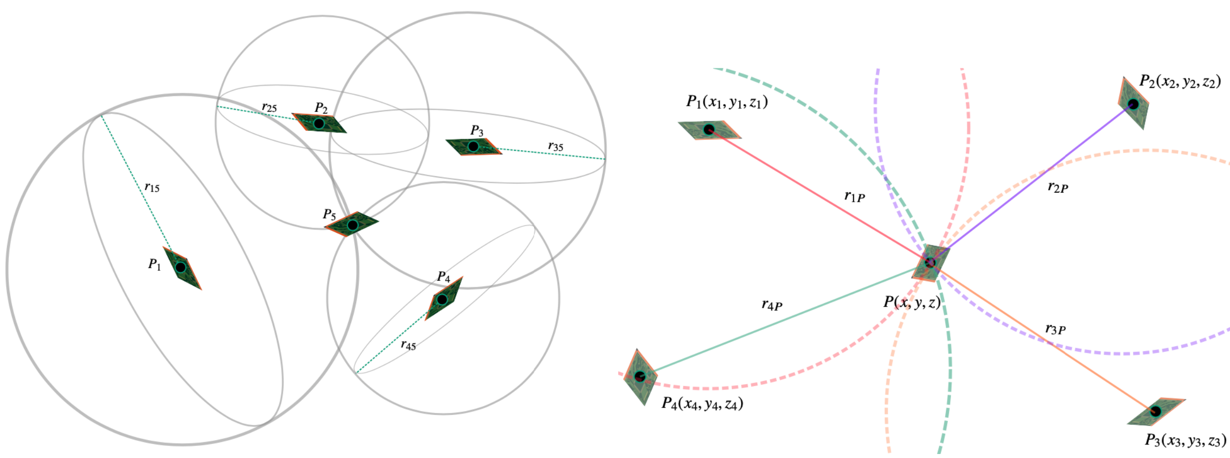
In Fig. 3, the anchor coordinates are marked by red squares, the true coordinates (unknown to the algorithm) are marked by grey circles and the algorithm estimates are marked by green crosses. The blue dashed lines indicate the difference between an estimate and the true position for that point. As can be seen from these results, the algorithm can comfortably handle a significant degree of inaccuracy and noise in the range estimates it is provided with and still localise the points. Figure 3f presents the case approaching the upper limit of this process, where the algorithm's performance is significantly degraded when supplied range estimate errors with a standard deviation of 20% of the average range between points. This demonstrates the performance of the algorithm in a controlled volume of space. In

Section III we consider the practical performance of this algorithm for femto-spacecraft in realistic dynamic scenarios as the swarm drifts away from its deployer spacecraft.

### B. Distributed Relative Positioning Algorithm

A centralised approach relies on one unit being supplied all information necessary to optimise the estimate of the relative positions for the swarm members. With a decentralised or distributed approach, we suppose varying degrees of shared data and computation between swarm members. Using a distributed positioning algorithm we consider relative trilateration of many unknown points using only the ranges between them to determine relative positions. The challenge for this strategy is developing an algorithm that is robust to measurement noise and the resulting potential ambiguities in the solutions. The key advantage of this method however is its scalability and ability to work in an anchorless way. The structure of the algorithm in computational implementation and in communication within the network could vary considerably depending on the application or swarm size. In the simplest case, a single femto-spacecraft would be able to determine its position relative to at least four of its neighbours.

With trilateration in three-dimensional space, knowledge of the ranges between four other known points that are not co-planar is in principle sufficient to uniquely identify a fifth point as the only possible intersection, as shown with femto-spacecraft in Fig. 4.

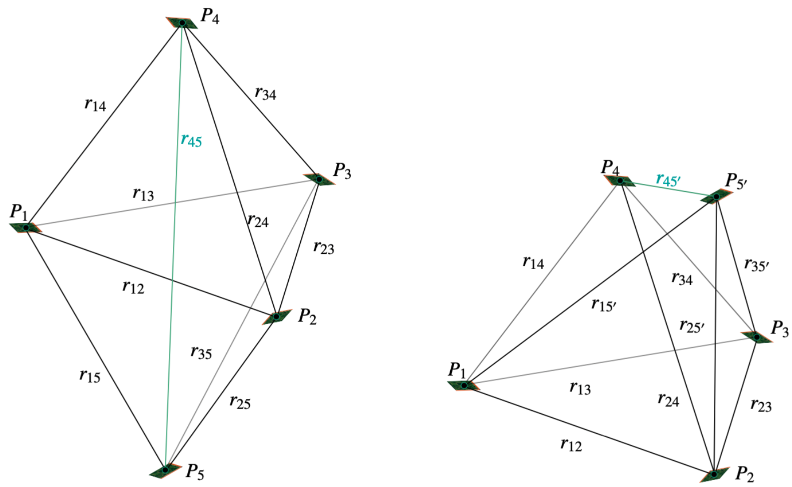



**(a) 4 ranging spheres to 5$^{\text{th}}$ femto-spacecraft** **(b) Unique intersection of 5$^{\text{th}}$ femto-spacecraft**

**Fig. 4 Relative trilateration in three-dimensional space**

The fifth femto-spacecraft $P_5$ is shown as the only possible intersection of the other four ranging spheres. A least-squares approach (or similar) is required to estimate position with this method in practice due to measurement noise. In its usual implementation, trilateration also requires the absolute coordinates of the four points to be known. With a relative approach, we reverse this idea to determine relative positions where there are many unknown points but there are estimates of the ranges between them.

The relative positions found must be robust against ambiguities up to a global translation, rotation and reflection (TRR). Translational ambiguity simply refers to the fact that a solution could be translated anywhere in space and remain valid, so solutions must not have such an ambiguity in the relative coordinate system the solution is found in. Rotational ambiguity is similar; suppose that the set of points form a structure consisting of lines between the femto-spacecraft that is able to be rotated in any direction about a central pivot common to the entire structure. Reflective ambiguity exists if a candidate solution point can be 'flipped' about a face shared by other points in a solution and still remain valid; effectively there is a potential 'ghost' solution that is (without further knowledge) equally correct. It is essential to avoid this ambiguity in constructing a network of further relative positions. Consider the scenario shown in Fig. 5.




**(a) Localisation** **(b) Alternate localisation**

**Fig. 5 Reflective ambiguity in three-dimensional space**

In Fig. 5a, consider that positions $P_1$ to $P_4$ are known in relative space, and that each ranging measurement to position $P_5$ is used to trilaterate and locate it relative to the first four positions. It is possible that in the presence of measurement noise both the solutions shown in Figs. 5a and 5b are valid without considering the range $r_{45}$ between $P_4$ and $P_5$. The solution shown in Fig. 5b, where the fifth position is above the plane formed by points $P_1$ to $P_3$, is also possible. Even when $r_{45}$ is considered, it is possible that the difference between $r_{45}$ and the potential $r_{45'}$ is not sufficient in the presence of noise to rule out one or the other. While Fig. 5 illustrates a case where the estimate $r_{45}$ would need to be completely anomolous to mistake the correct configuration (as the difference is so large), this is not always the case for random geometries, particularly in the presence of noise. In developing a distributed algorithm that uses relative trilateration it is essential to be robust against this kind of uncertainty and rule out candidate solutions that exhibit it. This is especially important for the first set of points used to start the algorithm and trilaterate new solutions to that first solution.

Consider now Fig. 5a to explain how relative trilateration would work as the basis of this algorithm. We can arbitrarily assign a femto-spacecraft at position $P_1$ as the origin of a new relative Cartesian coordinate system, position $P_2$ with the x-coordinate $r_{12}$ and positions $P_3$ and $P_4$ using basic trigonometry (forming a relative tetrahedron of four femto-spacecraft). This is described later (see Eq. (10)). We can perform that same operation with positions $P_1$, $P_2$, $P_3$, and $P_5$, and then confirm the relationship between both tetrahedra with the range measurement $r_{45}$. With this structure formed subject to strict ranging conditions that avoid reflective ambiguity, we can freely trilaterate new femto-spacecraft to this cluster in a simple way as we now know a sufficient number of true relative locations.

It is important to note here that five points connected by ten links is the smallest rigid structure in three-dimensional space that can be found up to a global TRR. Were we to use tetrahedra as a stitching mechanism for new points to trilaterate onto, many solutions could be found that exhibit ambiguities. Existing two-dimensional distributed algorithms use two-dimensional quadrilaterals as a stitching mechanism. The two-dimensional analogue of a tetrahedron is a triangle, with a quadrilateral being four points connected by six links in two-dimensional space, as has been described for two-dimensional WSN localisation [26]. The third dimension introduces a relative up/down direction for this scenario meaning that five points are required in three-dimensional space [27].

The first five femto-spacecraft start the algorithm by defining a relative orientation and position that newly trilaterated positions are found relative to. This process continues until confidence limits of new femto-spacecraft positions are reached based on an estimate of the size of the measurement noise. At this stage, the process would then restart for new unknown femto-spacecraft positions. This results in clusters in different relative coordinate systems that need to be transformed (translated and rotated) into a single relative reference frame. There are several ways of achieving this [28], provided the clusters share a degree of overlap with localised femto-spacecraft in common (at least four in three-dimensional space). Singular value decomposition (SVD) [29] has been found to be the most stable in computation, so we use this method for frame transformations in our distributed algorithm.

To describe how the algorithm determines the swarm's relative positions we first summarise the process and then explain each step in detail. The fundamental steps of our algorithm's logic are:

1) Trilaterate the first set of femto-spacecraft that pass volumetric and ranging tests against positioning ambiguity as positions $P_1$ - $P_5$ in the relative frame $\mathcal{F}_1$.
2) Trilaterate further positions onto this cluster using a non-linear least squares iteration process:
    a) Subject new configurations to the same volumetric and ranging tests for positioning ambiguity
    b) If a configuration passes volumetric tests, start with a 'first guess' linear least-squares estimate that solves the range equations
    c) Refine through iteration of a non-linear case with the Newton-Raphson method
    d) If the position found agrees with ranging test results against flip ambiguity, the femto-spacecraft is localised onto the cluster.

3) Continue to add new femto-spacecraft as in Step 2 until a (pre-determined) threshold on the error propagation in newly trilaterated positions is reached for the cluster.
4) Repeat Steps 1-3 with a new cluster of positions for the entire swarm, ensuring a degree of overlap in clusters for subsequent frame transformations.
5) Transform $n$ clusters in $n$ relative frames $\mathcal{F}_1 - \mathcal{F}_n$ into a single swarm localisation in one relative frame using singular value decomposition.

As described in step 1, to start a localisation cluster, the femto-spacecraft must pass both a volumetric test and a ranging (flip ambiguity) test to work in the presence of ranging error. For these tests, we employ a similar method in three-dimensional space to those presented in [27]. The two tests used work as follows:

(a) Volumetric test: prevents poor geometry and measurement noise allowing trilateration of non-robust structures (e.g. in two-dimensional space, the equivalent would be three ranges failing the triangle inequality). If the probability that a tetrahedron formed by four femto-spaceraft encloses a negative volume is above a pre-determined value (set at 1% in simulation), then the femto-spacecraft are not localised.

(b) Ranging test: prevents flip/reflective ambiguity by using the otherwise redundant tenth range estimate $r_{45}$ between five femto-spacecraft (Fig. 5). A statistical two-tailed z-test is used to determine within a 95% confidence interval that their positions are robust against reflective ambiguity.

These tests ensure robust determination of femto-spacecraft relative positions. The relative localisation in the arbitrary local frame $\mathcal{F}_1$ of the first 5 positions of the cluster ($P_1(x_1, y_1, z_1)$ to $P_5(x_5, y_5, z_5)$) can then be assigned from trigonometry:

$$P_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} P_2 = \begin{bmatrix} r_{12} \\ 0 \\ 0 \end{bmatrix} P_3 = \begin{bmatrix} \frac{r_{13}(r_{23}^2 - r_{12}^2 - r_{13}^2)}{-2r_{12}r_{13}} \\ \sqrt{r_{13}^2 - x_3^2} \\ 0 \end{bmatrix} P_4 = \begin{bmatrix} \frac{r_{14}^2 - r_{24}^2 + r_{12}^2}{2r_{12}} \\ \frac{r_{14}^2 - r_{34}^2 + x_3^2 + y_3^2 - 2x_3x_4)}{2y_3} \\ \sqrt{r_{14}^2 - x_4^2 - y_4^2} \end{bmatrix} P_5 = \begin{bmatrix} \frac{r_{15}^2 - r_{25}^2 + r_{12}^2}{2r_{12}} \\ \frac{r_{15}^2 - r_{35}^2 + x_3^2 + y_3^2 - 2x_3x_5}{2y_3} \\ \pm\sqrt{r_{15}^2 - x_5^2 - y_5^2} \end{bmatrix} \quad (10)$$

where the relative orientation of $\pm z_5$ is determined by the flip ambiguity test. Further femto-spacecraft are then trilaterated onto this cluster using a non-linear least squares refinement process. Using Newton's method, this starts with a first iteration 'guess' from a linear least-squares solution of the spherical ranging equations which is refined with non-linear least squares. First, we solve the trilateration problem with a system of linear equations involving the new femto-spacecraft to be localised $P(x, y, z)$ and four other femto-spacecraft already trilaterated within the cluster (e.g. $P_1$ to $P_4$). Note here that the notation $P_i$ to $P_{i+3}$ (where $i = 1 - n$) would describe the general case for this scenario with four arbitrary relative positions used, but we use the notation $P_1$ to $P_4$ in the following equations for clarity in the expressions. Also note that this method uses the ranges between the femto-spacecraft used that are already trilaterated. This requires only the four ranges between each of the localised femto-spacecraft and the new femto-spacecraft to be localised to add this point to the structure. Its formation is still subject to the same volumetric and ranging tests as used for cluster formation. From Fig. 4b the four equations:

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = r_{iP}^2 \qquad (i = 1 - 4) \tag{11}$$

can be rearranged into a system of 3 linear equations of the form:

$$H\mathbf{x} = \mathbf{b} \tag{12}$$

where:

$$H = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ x_4 - x_1 & y_4 - y_1 & z_4 - z_1 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x - x_1 \\ y - y_1 \\ z - z_1 \end{bmatrix} \qquad \mathbf{b} = \frac{1}{2}\begin{bmatrix} (r_{1P}^2 - r_{2P}^2 + r_{12}^2) \\ (r_{1P}^2 - r_{3P}^2 + r_{13}^2) \\ (r_{1P}^2 - r_{4P}^2 + r_{14}^2) \end{bmatrix} \tag{13}$$

This can be solved in a linear least-squares sense as:

$$\mathbf{x} = (H^T H)^{-1} H^T \mathbf{b} \tag{14}$$

Solving for $\mathbf{x}$ we can now find the newly trilaterated femto-spacecraft position $P(x, y, z)$. This is refined using a non-linear least squares method that as a first iteration starts with the linear least squares solution. This method minimises the sum of the squares of the range errors, which is achieved by minimising:

$$F(x, y, z) = \sum_{i=1}^{n} (\hat{r}_{iP} - r_{iP})^2 = \sum_{i=1}^{n} f_i(x, y, z)^2 \tag{15}$$

where $\hat{r_{iP}}$ is the estimated range between femto-spacecraft $i$ and the femto-spacecraft at position $P$, $r_{iP}$ is the true range, and $f_i(x, y, z)$ is given by:

$$f_i(x, y, z) = \hat{r_i} - r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - r_i \tag{16}$$

Finding the partial derivatives of Eq. (15) with respect to $x$, $y$ and $z$ yields:

$$\mathbf{g} = 2J^T \mathbf{f} \tag{17}$$

where $\mathbf{g}$ is the vector of partial derivatives and $J$ is the Jacobian matrix:

$$\mathbf{g} = \begin{pmatrix} \frac{\partial F}{\partial x} \\ \frac{\partial F}{\partial y} \\ \frac{\partial F}{\partial z} \end{pmatrix} \qquad J = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial z} \end{bmatrix} \qquad \mathbf{f} = \begin{bmatrix} \hat{r_{15}} - r_{15} \\ \hat{r_{25}} - r_{25} \\ \hat{r_{35}} - r_{35} \\ \hat{r_{45}} - r_{45} \end{bmatrix} \tag{18}$$

Then, iterating with Newton's method for point $P$:

$$P_{k+1} = P_k - (J_k^T J_k)^{-1} J_k \mathbf{f}_k \tag{19}$$

This procedure can be repeated for a set number of iterations or until convergence within a given tolerance level. The number of attempts that a femto-spacecraft of unknown position would have to localise onto the cluster would vary. However, we assume that if extra available configurations are available in a sampling cycle through additional range estimates to different femto-spacecraft within the cluster, that these would be used, which we assume in simulation. Other femto-spacecraft can continue to be localised onto a cluster until the solutions exceed noise bounds or confidence levels. At this stage new femto-spacecraft yet to be localised would find another cluster to localise to, or would start another cluster altogether.

When cluster positions are combined, a suitable amount of overlap in relative positions is required for the transformation as described above. Therefore, it is necessary to know some femto-spacecraft positions in at least two reference frames for transformations. Singular value decomposition frame transformation can be used given the coordinates of a minimum of the same four non-coplanar positions known in two separate frames of reference, such that a transformation (rotation matrix and translation vector) between the two frames can be found with the steps below. Consider the two sets of positions $q$ and $m$, representing two $3 \times n$ matrices of the same $n$ positions expressed in different reference frames:

1) Find the centroids of each set of positions: $\bar{\mathbf{q}} = \frac{\sum_{i=1}^{n} q_i}{n}$ & $\bar{\mathbf{m}} = \frac{\sum_{i=1}^{n} m_i}{n}$, where $n$ is the number of positions
2) Find the centred vectors: $q_{ci} = \mathbf{q_i} - \bar{\mathbf{q}}$ & $m_{ci} = \mathbf{m_i} - \bar{\mathbf{m}}$ (represented as arrays storing all vector combinations)
3) Find the 3×3 covariance matrix: $H = \mathbf{q_{ci}} \mathbf{m_{ci}}^T$
4) Find the singular value decomposition: $H = U \sum V^T$
5) The desired rotation is calculated as $R = VU^T$
6) The desired translation is calculated as $T = \bar{\mathbf{m}} - R\bar{\mathbf{q}}$
7) The two sets of positions are then related by the transformation $q = Rm + T$

In the singular value decomposition, $U$ and $V$ are $3 \times 3$ unitary matrices and $\sum$ is a $3 \times 3$ diagonal matrix. With this rotation and translation, further positions can be transformed into either reference frame as required. This algorithm is implemented in MATLAB. Using the same performance criteria for the centralised algorithm reported in Section II.A, the following results show single-cluster performance for the distributed algorithm in the same scenario used in Section II.A. This is anchorless, so all points are localised and the output is transformed into an absolute frame of reference to compare its performance with the centralised algorithm.

Table 2 displays the algorithm performance. For this algorithm we also quantify the number of femto-spacecraft localised to a cluster on their first attempt to do so, and the number localised in total after all potential range configurations

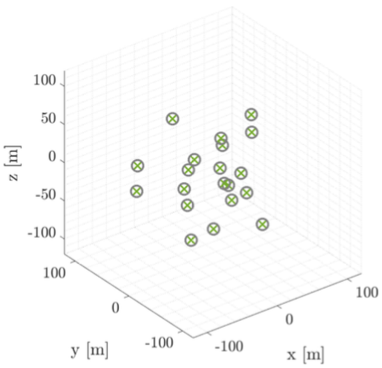
**Table 2 Distributed algorithm performance in random scenario (single cluster)**

| $\sigma_r$ | $\sigma_p$ $[m]$ | No. Localised on 1st Attempt | No. Localised |
|---|---|---|---|
| 0 | 0 | 20 | 20 |
| $0.01\bar{r}$ (0.92 m) | 2.66 | 13 | 20 |
| $0.02\bar{r}$ (1.84 m) | 4.33 | 11 | 20 |
| $0.03\bar{r}$ (2.75 m) | 5.42 | 16 | 17 |
| $0.04\bar{r}$ (3.67 m) | 11.32 | 16 | 19 |
| $0.05\bar{r}$ (4.59 m) | 11.64 | 9 | 15 |
| $0.06\bar{r}$ (5.51 m) | 12.06 | 6 | 17 |
| $0.07\bar{r}$ (6.43 m) | 7.61 | 8 | 10 |
| $0.08\bar{r}$ (7.34 m) | 21.37 | 9 | 17 |
| $0.09\bar{r}$ (8.26 m) | 18.12 | 8 | 14 |
| $0.10\bar{r}$ (9.18 m) | 12.84 | 7 | 7 |
| $0.20\bar{r}$ (18.36 m) | 17.52 | 6 | 6 |

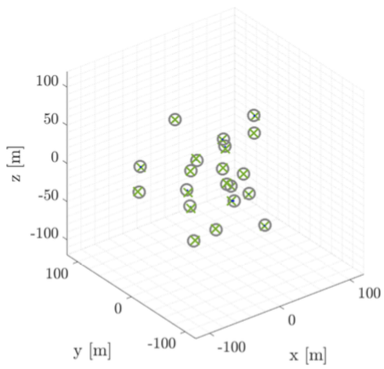
between each swarm member localised have been attempted for a single cluster. The results shown in Fig. 6 demonstrate the ability of the distributed algorithm to localise the majority of spacecraft in single-run attempts up to a ranging noise of 9% $\bar{r}$. This represents the best performance that can be expected in simulation using the thresholds described for the volumetric and ranging tests to prevent incorrect localisation of femto-spacecraft.

As ranging noise increases, there is an expected general trend towards fewer first attempt localisations (i.e. femto-spacecraft localised using range estimates between the first four femto-spacecraft attempted), as shown in Table 2. In general this leads to fewer total localisations for a single algorithm run as noise increases. In practice, frequent sampling of the algorithm would update the relative position estimates of the swarm to account for the swarm state changing in space with time. Femto-spacecraft that fail to localise on a particular algorithm cycle would have the opportunity to do so on the next cycle.

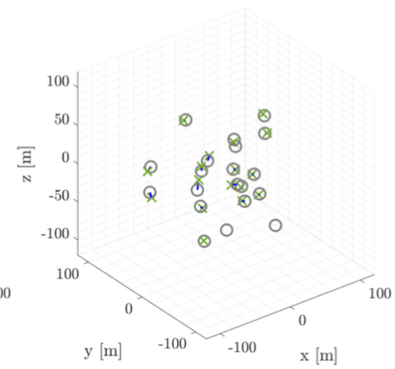
Analysing the generalised performance of both algorithms in this scenario, as shown in Figs. 3g and 6g, we find that the distributed algorithm localises to a comparable accuracy with the centralised algorithm as noise levels in the range estimates increase (from linear best fit at a AWGN level of $10\%\bar{r} = 8.5\ m$, in the centralised algorithm $\sigma_p = 15.5\ m$ and in the distributed algorithm $\sigma_p = 14\ m$). However, this is only when considering the femto-spacecraft that manage to localise, as the distributed algorithm would not necessarily localise all femto-spacecraft at higher noise levels. This emphasises the contrasting approaches; while the centralised algorithm works on one spacecraft that is provided with all swarm range data, in the distributed algorithm individual femto-spacecraft have less range data to localise with.



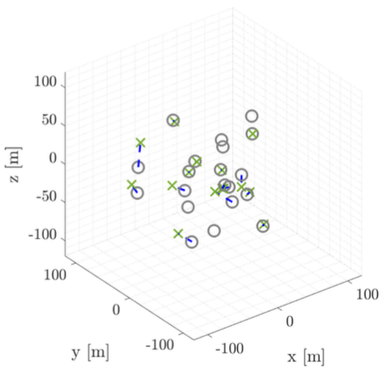

**(a)** $\sigma_r = 0$



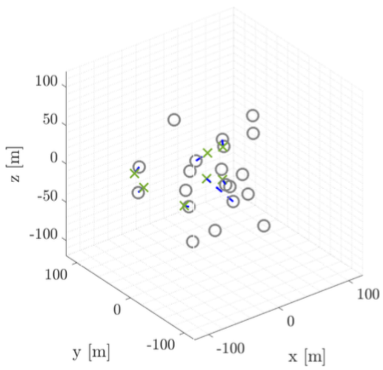

**(b)** $\sigma_r = 0.01\bar{r}$



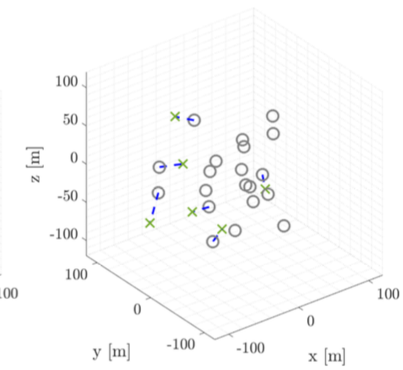

**(c)** $\sigma_r = 0.03\bar{r}$



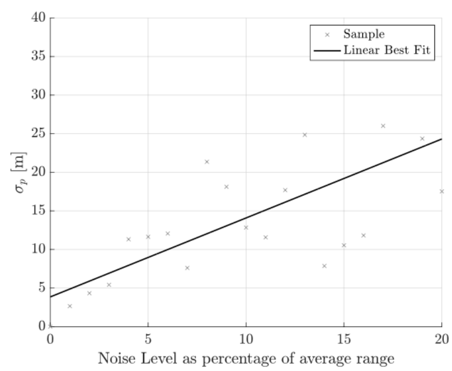

**(d)** $\sigma_r = 0.05\bar{r}$




**(e)** $\sigma_r = 0.10\bar{r}$




**(f)** $\sigma_r = 0.20\bar{r}$




**(g)** $\sigma_p$ **as noise levels increase**

**Fig. 6 Distributed algorithm performance in random scenario**

## III. Relative Navigation for a Femto-Spacecraft Swarm

In this section we apply the relative positioning methodology described Section II within realistic femto-spacecraft swarm dispersal scenarios using a model of the relative dynamics of the swarm with respect to its deployer. This enables the development of a simple relative navigation filter as an example of the utility of the algorithms in practice. For these purposes we assume the femto-spacecraft have no means of controlling their relative positions and the initial conditions for the dynamics are defined by the deployer.

### A. Relative Dynamics

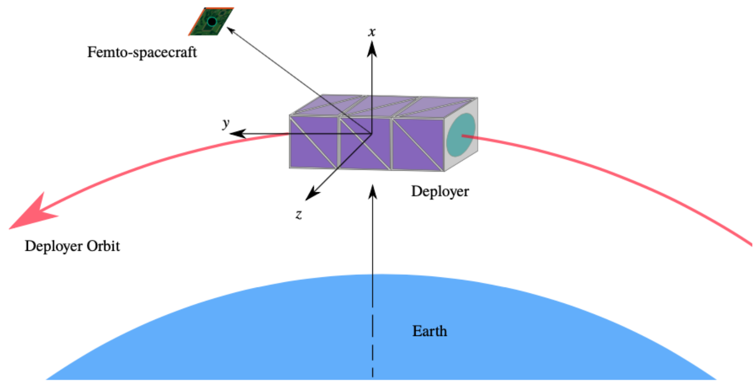



**Fig. 7 Clohessy-Wiltshire Reference Frame**

The Clohessy-Wiltshire (CW) equations [30] provide a linearised approximation of the relative motion of a chase spacecraft with respect to a target spacecraft in a target-centred reference frame. We will use the terms 'deployer' and 'femto-spacecraft' to describe the target and chase spacecraft respectively. The deployer is assumed to be in a circular orbit while the femto-spacecraft drift passively relative to it. Perturbations to the two-body relative motion problem are neglected in this analysis. The CW equations are defined as:

$$\ddot{x} - 2\omega_n \dot{y} - 3\omega_n^2 x = 0 \tag{20}$$

$$\ddot{y} + 2\omega_n \dot{x} = 0 \tag{21}$$

$$\ddot{z} + \omega_n^2 z = 0 \tag{22}$$

where:

$$\omega_n = \sqrt{\frac{\mu}{R_o^3}} = \frac{2\pi}{T_o} \tag{23}$$

The mean motion of the deployer $\omega_n$ is expressed in terms of the standard gravitational parameter of the central body $\mu$ and the orbital radius $R_o$ of the deployer's orbit, which has an orbital period $T_o$. In this deployer-centred reference frame, the x-axis points outwards along the radius vector of the deployer spacecraft (radial motion), the y-axis points forwards along the velocity vector (along-track motion), and the z-axis completes the right-handed set by pointing along the deployer's orbital angular momentum vector (cross-track motion). This means that the central orbital body (e.g. Earth) is towards the negative x-direction of the deployer. This is illustrated in Fig. 7. Expressing the CW equations in state space form allows a closed form solution to be expressed such that:

$$\dot{x}(t) = Ax(t) \tag{24}$$

Here the state vector $x(t)$ and the system matrix $A$ are given by:

$$x(t) = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3\omega_n^2 & 0 & 0 & 0 & 2\omega_n & 0 \\ 0 & 0 & 0 & -2\omega_n & 0 & 0 \\ 0 & 0 & -\omega_n^2 & 0 & 0 & 0 \end{bmatrix} \tag{25}$$

These equations can then be solved in terms of a state transition matrix $\Phi$ and the initial conditions $x(t_0)$ such that:

$$x(t) = e^{A(t-t_0)}x(t_0) = \ \Phi x(t_0) \tag{26}$$

Abbreviating $\sin(\omega_n t) = s$ and $\cos(\omega_n t) = c$, for $t_0 = 0$ the solutions are:

$$\begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} 4-3c & 0 & 0 & \frac{s}{\omega_n} & \frac{2}{\omega_n} - \frac{2c}{\omega_n} & 0 \\ 6s - 6\omega_n t & 1 & 0 & \frac{2c}{\omega_n} - \frac{2}{\omega_n} & \frac{4s}{\omega_n} - 3t & 0 \\ 0 & 0 & c & 0 & 0 & \frac{s}{\omega_n} \\ 3\omega_n s & 0 & 0 & c & 2s & 0 \\ 6\omega_n c - 6\omega_n & 0 & 0 & -2s & 4(c-3) & 0 \\ 0 & 0 & -\omega_n s & 0 & 0 & c \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ \dot{x_0} \\ \dot{y_0} \\ \dot{z_0} \end{bmatrix} \tag{27}$$

With this closed form solution we can propagate the state of each femto-spacecraft using the state transition matrix. The CW equations are now used to consider how femto-spacecraft ejected from their deployer may disperse and drift with time, and how the relative positioning algorithms proposed can estimate the location of each swarm member. The initial state vector of the femto-spacecraft deployed will be coincident with the deployer, with some initial ejection velocity for each femto-spacecraft relative to the deployer.

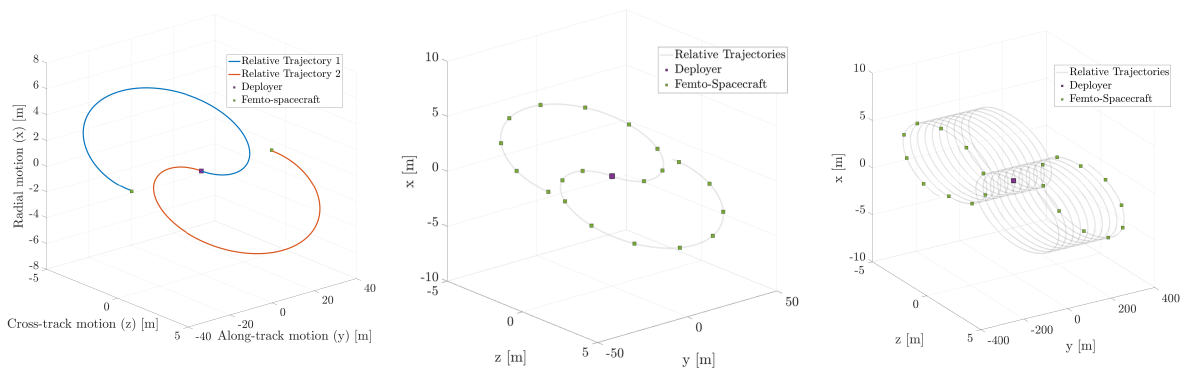
We will consider two swarm deployment and dispersal methods. Firstly, a controlled, sequential ejection of the swarm with fixed impulses in front of and behind the deployer, and secondly, a randomly scattered, instantaneous ejection of the entire swarm around the deployer. These could be used in applications such as sparse aperture interferometry or massively-parallel space environment sensing. For both scenarios we consider the deployer to be in a circular low Earth orbit of altitude 400 km. We also consider only the ejection impulse contributing to the swarm dispersal and that no other perturbing forces affect the swarm dynamics.

*Sequential Swarm Ejection*

Here we simulate the sequential ejection of a swarm of femto-spacecraft over one Earth orbit for the deployer, where each femto-spacecraft is ejected with the same impulse. The initial state of each femto-spacecraft relative to the deployer is assumed to be coincident in position with the deployer and an ejection velocity given by:

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} m \qquad \begin{bmatrix} \dot{x_0} \\ \dot{y_0} \\ \dot{z_0} \end{bmatrix} = \pm \begin{bmatrix} 0 \\ 2 \\ 4 \end{bmatrix} \times 10^{-3} m/s \tag{28}$$

If two femto-spacecraft are ejected with the same impulse relative to their deployer, one in front and one behind the deployer, over one full Earth orbit for the deployer, their trajectories relative to the deployer take the form as shown in Fig. 8a. By deploying the swarm evenly and sequentially over an orbit the swarm members would be located at different phases along this same relative trajectory as shown in Fig. 8b, where a swarm of 20 femto-spacecraft (10 ahead and 10 behind) are deployed, which would then gradually drift farther from the deployer in following orbits, as shown in Fig. 8c.




**(a) Trajectories over 1 orbit** **(b) Swarm state at $t = T$** **(c) $t = 10T$**

**Fig. 8 Sequential swarm ejection: swarm state with time**

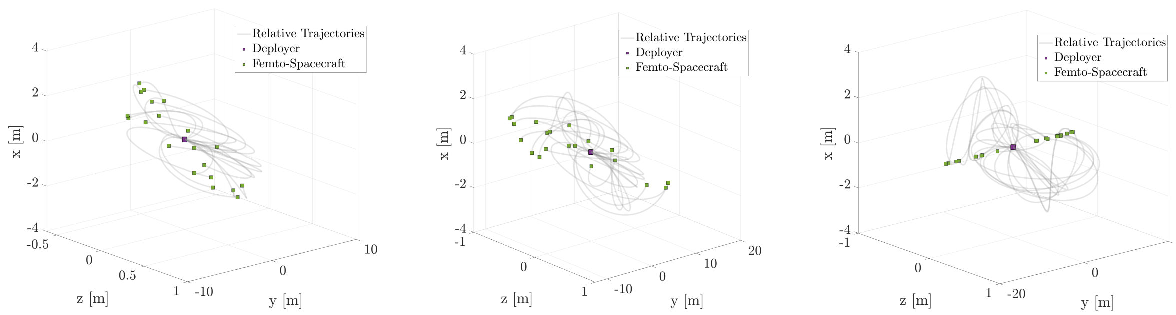
With this deployment sequence the swarm drifts away from the deployer, bounded in the radial and cross-track directions, dispersing approximately $\pm 50$ $m$ from the deployer after 1 orbit, to $\pm 500$ $m$ from the deployer after 10 orbits.

*Random and Instantaneous Swarm Ejection*

Rather than a controlled sequential ejection of the swarm to drift away from the carrier, the swarm could be ejected instantaneously in random directions around the deployer at the same speed, with the initial state for the entire swarm of:

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} m \qquad |v_0| = 1 \times 10^{-3} m/s \tag{29}$$

This ejection is modelled with the velocity vectors pointing in uniformly distributed random directions around the deployer, each with the same magnitude. In practice, the available deployment directions would be limited. Figure 9 displays a swarm of 20 femto-spacecraft deployed randomly around their deployer, at different stages throughout an orbit:




**(a)** $t = \frac{1}{2}T$ **(b)** $t = \frac{3}{4}T$ **(c)** $t = T$

**Fig. 9 Random swarm ejection: swarm state with time**

After each relative orbit, the swarm returns to a state where each femto-spacecraft is located either in front of or behind the deployer, drifting further over the course of several orbits.

### B. Relative Navigation Filter

We now use a Kalman filter (KF) [31] to demonstrate a relative navigation system for the swarm, combining the algorithm outputs and relative dynamics in the above scenarios to filter the relative state estimation over time. In the following scenarios we consider a simple centralised filter operating on the same spacecraft that is running the centralised relative positioning algorithm. The discrete time model used is:

$$\mathbf{x_k} = \Phi \mathbf{x_{k-1}} + \mathbf{w_{k-1}} \tag{30}$$

$$\mathbf{z_k} = \mathbf{x_k} + \nu_\mathbf{k} \tag{31}$$

where $\mathbf{x_k}$ is the state at discrete time step $k$, $\Phi$ is the state transition matrix from the CW equations, $\mathbf{w}$ is the process noise, $\mathbf{z}$ is the state measurement and $\nu$ is the observation noise. We use the relative positions from the algorithm to be a partial linear observer of the femto-spacecraft state in the measurement model. In this implementation, the algorithm is sampled at every discrete time step, with the swarm relative positions output by the centralised positioning algorithm (as detailed in Section II.A) providing the input for the filter's measurement update.

The filter works through two stages iteratively, beginning with an initial state $\hat{\mathbf{x}}^-_{\mathbf{k-1}}$ and state covariance estimate $P^-_{k-1}$ from the initial conditions. Firstly, the time update stage:

$$\hat{\mathbf{x}}_{\mathbf{k}}^{-} = \Phi \hat{\mathbf{x}}_{\mathbf{k-1}}^{+} \tag{32}$$

$$P_k^- = \Phi P_{k-1}^+ \Phi^T + Q_{k-1} \tag{33}$$

where the -/+ superscripts denote the a-priori/a-posteriori estimates and $Q$ is the process noise covariance matrix. Then the measurement update stage:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{34}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \tag{35}$$

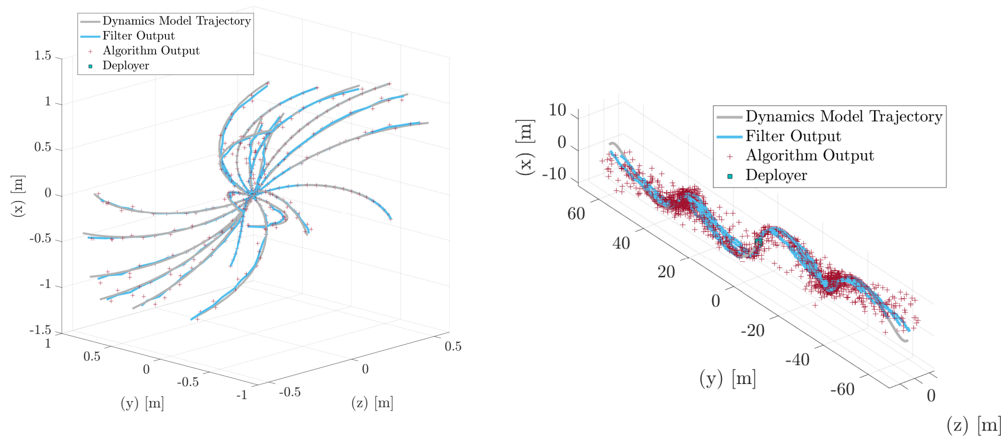
$$P_k^+ = (I - K_k H) P_k^- \tag{36}$$

where $K$ is the Kalman gain, $H = I$ is the observation matrix, and $R$ is the sensor noise covariance matrix. The filter then iterates through $k - 1 \leftarrow k$, using the output of the measurement update stage as part of the calculation in the next time update stage. We assume $Q$ is a diagonal matrix with the following elements for the CW equations [32]:

$$Q = \begin{bmatrix} Q_r & 0 \\ 0 & Q_v \end{bmatrix} Q_r = \begin{bmatrix} 2^2 & 0 & 0 \\ 0 & 2^2 & 0 \\ 0 & 0 & 2^2 \end{bmatrix} \times (10^{-2} m)^2 \;\; Q_v = \begin{bmatrix} 2^2 & 0 & 0 \\ 0 & 2^2 & 0 \\ 0 & 0 & 2^2 \end{bmatrix} \times (10^{-3} m/s)^2 \tag{37}$$

The relative navigation of the swarm would begin after release from the deployer. Ejection from the deployer would provide the initial state estimate of each femto-spacecraft within the swarm. The relative velocity at the point of ejection would introduce uncertainty.

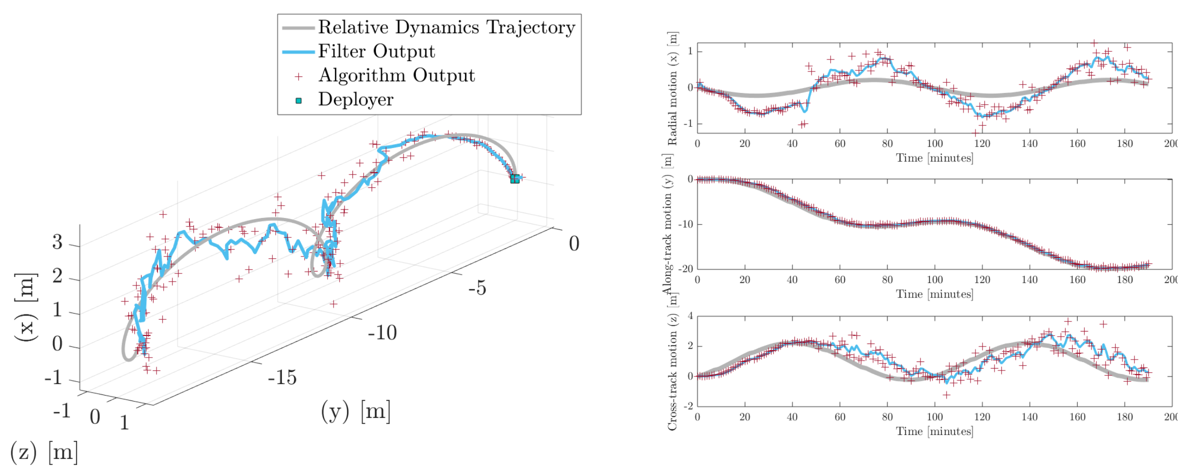
We now apply the filter in the scenarios described above to demonstrate relative positioning performance with the centralised algorithm. In this simulation we set a AWGN level of $\sigma_r = 0.02\bar{r}$, and we use a sampling interval of 1 minute. Figure 10a displays the filtering of a swarm 20 femto-spacecraft deployed randomly and instantaneously from a deployer (only shown to $t = \frac{T}{5}$ for clarity). The algorithm output samples are marked by the red crosses, while the blue lines indicates the Kalman filter output and the grey lines indicate the dynamics model trajectories for each femto-spacecraft from the deployer, which is located at the origin. Figure 10b displays the sequential ejection case over the course of an orbit (where the femto-spacecraft that are deployed last are re-tracing the trajectories of those deployed first). Note that the filtering in Fig. 10b occurs between the first and second orbit after all femto-spacecraft have been deployed. To analyse the filter performance in detail, we isolate the filtering of just one femto-spacecraft within the swarm over the course of 2 orbits (190 minutes) for both scenarios. This is displayed firstly for the random deployment




**(a) Random ejection (shown to t = $\frac{T}{5}$)** **(b) Sequential ejection**

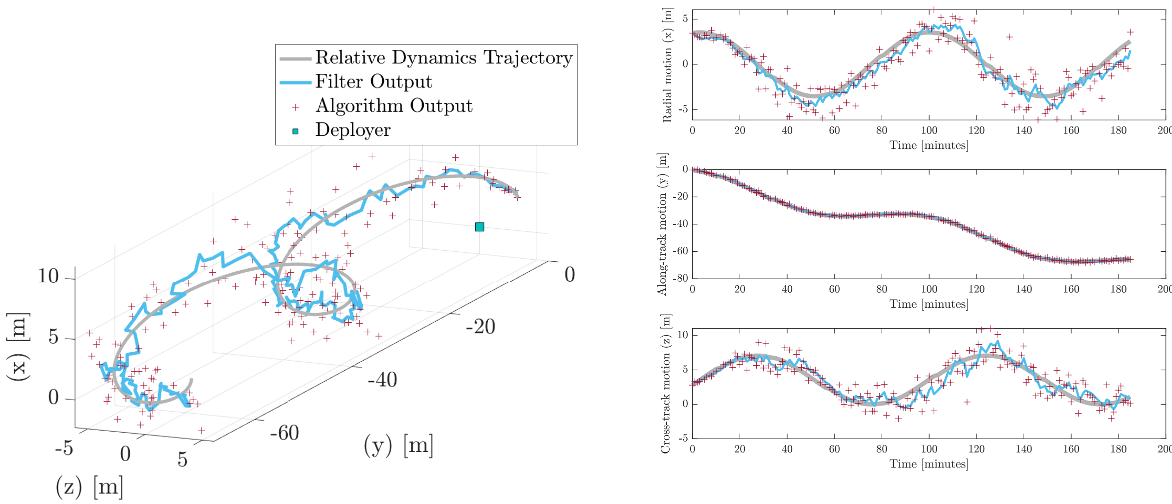
**Fig. 10 Swarm relative positioning filtering**

in Fig. 11 and then for the controlled sequential deployment in Fig. 12.




**(a) 3D** **(b) Individual axes**

**Fig. 11 KF positioning of femto-spacecraft 1 (random ejection)**




**(a) 3D** **(b) Individual axes**

**Fig. 12 KF positioning of femto-spacecraft 1 (sequential ejection)**

The algorithm output samples are again marked by the red crosses in Figs.11 and 12 indicate the sampled algorithm outputs at each discrete time step, while the blue line indicates the Kalman filtering of these samples. As can be seen in the plots of both the three-dimensional relative motion and the individual axes, the femto-spacecraft's relative position state estimate with time is smoothed by the filter.

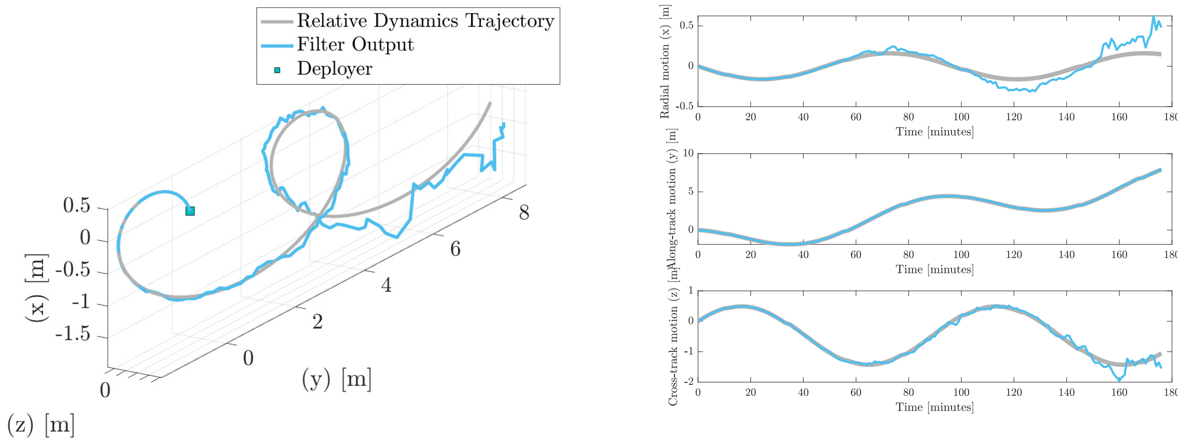
As an alternative approach, the outputs of the positioning algorithms can be used for initial relative orbit determination (IROD), with the localisation solutions generated providing initialisation for an extended Kalman filter (EKF) using RSSI values directly. This approach requires an EKF only because the range-based measurement model is non-linear. The state measurement model from Eq. (31) is adapted for the EKF:

$$\mathbf{z_k} = h(\mathbf{x_k}) + \nu_\mathbf{k} \tag{38}$$

where $h$ is the observation function of the state, and the observation matrix $H$ is now defined by the following Jacobian:

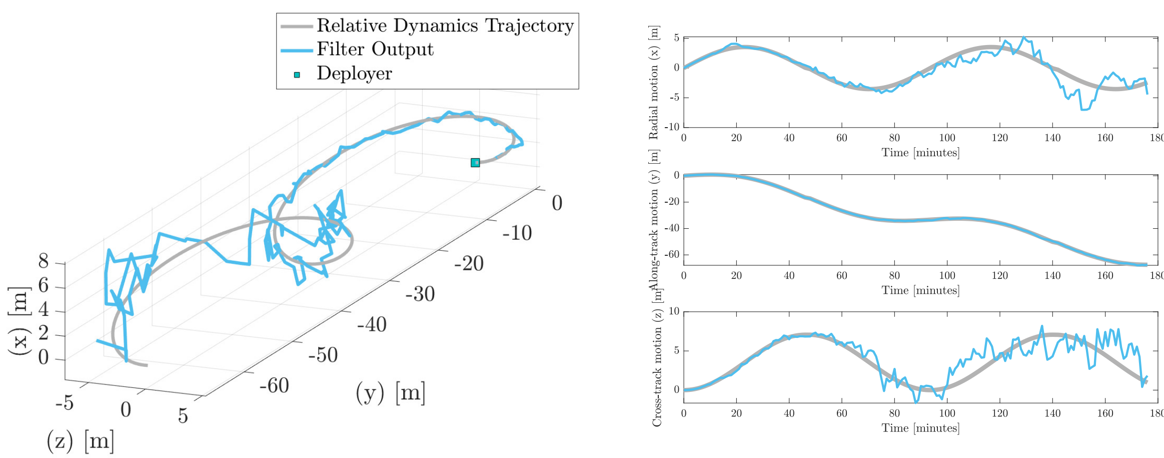
$$H_k = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{x}_k^-} \tag{39}$$

Otherwise, the EKF works in the same way as the KF with the steps in Eqs. (32 - 36). The outputs of this filtering approach are displayed in Figs. 13 and 14, again highlighting the filtering of a femto-spacecraft within the swarm over 2 orbits in both random and sequential deployment scenarios.




**(a) 3D** **(b) Individual axes**

**Fig. 13 EKF positioning of femto-spacecraft 1 (random ejection)**




**(a) 3D** **(b) Individual axes**

**Fig. 14 EKF positioning of femto-spacecraft 1 (sequential ejection)**

The results of both these filtering approaches demonstrate the improved relative navigation that Kalman filtering provides over relative positioning algorithm outputs alone. Combining confidence in the algorithm sampling and the relative dynamics provides improved relative positioning when compared to discrete algorithm samples at each discrete time step. In the EKF case, where the algorithm is used for initialisation, the results show that using a range-based measurement model is also viable. As a femto-spacecraft swarm drifts further from its deployer, the uncertainties in the state estimation would grow with time, so this is important in bounding errors and providing improved relative positioning for a swarm. The results demonstrate that either of these approaches could be implemented for a centralised navigation filter for swarm relative positioning. Depending on operational constraints, such as available processing power, one method or combination of both KF and EKF methods could be implemented.

# IV. Conclusions

In this paper, novel methods for the relative navigation of a swarm of centimetre-scale femto-spacecraft using only range estimates available from the communication links between swarm members have been presented. Two relative positioning algorithms that utilise these range estimates have been detailed; firstly a centralised algorithm designed to optimise the relative positioning of the entire swarm using the computational resources of a single device, and a distributed algorithm designed to share this computational load throughout the swarm. Sample deployment and dispersal scenarios have been presented, and the algorithms operating with a basic navigation filter to improve state estimation over time has been shown.

The performance of both algorithms in simulated random scenarios under the presence of increasing measurement noise demonstrates their viability in uniquely localising femto-spacecraft relative to one another using coarse range estimates. The methods presented would provide a low-cost relative positioning system for a swarm of femto-spacecraft that while of limited computational resource individually can operate collectively for many space-based applications. The relative navigation demonstrated using the centralised algorithm and Kalman filtering in different swarm dispersal scenarios highlights opportunities for their practical implementation in low Earth orbit.

The range-based relative navigation approaches described in this paper could be implemented within femto-spacecraft swarms to enable novel applications in space with a unique design methodology that could provide enhanced space-based satellite utilities in the near future.

# Acknowledgments

This work was supported by the Royal Academy of Engineering under the Chair in Emerging Technologies Scheme (Colin McInnes). Thomas Timmons acknowledges the support of an EPSRC studentship (EP/R513222/1).

# References

[1] Manchester, Z., Peck, M., and Filo, A., "KickSat : A Crowd-Funded Mission To Demonstrate The World's Smallest Spacecraft," *Annual AIAA/USU Conference on Small Satellites*, Logan, Utah, 2013. URL `https://digitalcommons.usu.edu/smallsat/2013/all2013/111/`.

[2] Hu, Z., Timmons, T., Stamat, L., and McInnes, C., "Development of a 10g Femto-satellite with Active Attitude Control," *Proceedings of the 17th Reinventing Space Conference*, Belfast, 2019. URL `http://eprints.gla.ac.uk/204036/1/204036.pdf`.

[3] Hein, A. M., Burkhardt, Z., and Eubanks, T. M., "AttoSats: ChipSats, other Gram-Scale Spacecraft, and Beyond," *17th IAA Symposium on Visions and Strategies for the Future - 70th International Astronautical Congress*, Washington D.C., 2019. URL `http://arxiv.org/abs/1910.12559`.

[4] Timmons, T., Bailet, G., Beeley, J., and McInnes, C., "Mars Atmospheric Characterization with a ChipSat Swarm," *Journal of Spacecraft and Rockets*, March, 2021. https://doi.org/https://doi.org/10.2514/1.A34970.

[5] Hadaegh, F. Y., Chung, S. J., and Manohara, H. M., "On Development of 100-gram-class spacecraft for swarm applications," *IEEE Systems Journal*, Vol. 10, No. 2, 2016, pp. 673–684. https://doi.org/10.1109/JSYST.2014.2327972.

[6] Sneeuw, N., "Satellite clusters for future gravity field missions," *Gravity, Geoid and Space Missions: IAG International Symposium*, Springer, Porto, 2004, pp. 12–17. https://doi.org/ISBN:978-3-540-26930-4(print)978-3-540-26932-9(online).

[7] Manchester, Z. R., and Peck, M. A., "Stochastic space exploration with microscale spacecraft," *AIAA Guidance, Navigation, and Control Conference*, Portland, Oregon, 2011. https://doi.org/10.2514/6.2011-6648.

[8] Janson, S., and Barnhart, D., "The Next Little Thing : Femtosatellites," *Proceedings of the Small Satellite Conference, Technical Session VI*, Logan, Utah, 2013. URL https://digitalcommons.usu.edu/smallsat/2013/all2013/85/.

[9] Verhoeven, C. J., Bentum, M. J., Monna, G. L., Rotteveel, J., and Guo, J., "On the origin of satellite swarms," *Acta Astronautica*, Vol. 68, No. 7-8, 2011, pp. 1392–1395. https://doi.org/10.1016/j.actaastro.2010.10.002, URL http://dx.doi.org/10.1016/j.actaastro.2010.10.002.

[10] D'Errico, M. (ed.), *Distributed Space Missions For Earth System Monitoring*, 1$^{st}$ ed., Microcosm Press, Springer, El Segundo, California, 2013. URL ISBN:978-1-4614-4540-1.

[11] Savvides, A., Han, C. C., and Strivastava, M. B., "Dynamic fine-grained localization in ad-hoc networks of sensors," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, Rome, 2001, pp. 166–179. https://doi.org/10.1145/381677.381693.

[12] Phillips, V., "NASA Concepts: Swarm/SODA [Online]," Last accessed June 2021, 2017. URL https://www.nasa.gov/feature/swarmsoda.

[13] Marszalek, M., Kurz, O., Drentschew, M., Schmidt, M., and Schilling, K., *Intersatellite links and relative navigation: Pre-conditions for formation flights with pico- and nanosatellites*, Vol. 44, IFAC, 2011. https://doi.org/10.3182/20110828-6-IT-1002.02369, URL http://dx.doi.org/10.3182/20110828-6-IT-1002.02369.

[14] Adams, V. H., and Peck, M., "R-Selected Spacecraft," *AIAA Journal of Spacecraft and Rockets*, Vol. 57, No. 1, 2019. https://doi.org/10.2514/1.A34564.

[15] Magness, R., "Short-range RF Wireless (Proximity) Networks Technology Assessment for Space Applications," *ESA Tec-E Wireless Technology Dossier*, 2006.

[16] Bezouska, W., and Barnhart, D., "Decentralized Cooperative Localization with Relative Pose Estimation for a Spacecraft Swarm," *IEEE Aerospace Conference Proceedings*, Big Sky, Montana, 2019. https://doi.org/10.1109/AERO.2019.8742158.

[17] Sabatini, M., Volpe, R., and Palmerini, G. B., "Centralized visual based navigation and control of a swarm of satellites for on-orbit servicing," *Acta Astronautica*, Vol. 171, No. January, 2020, pp. 323–334. https://doi.org/10.1016/j.actaastro.2020.03.015.

[18] Tapley, B. D., Bettadpur, S., Watkins, M., and Reigber, C., "The gravity recovery and climate experiment : Mission overview and early results," Vol. 31, No. November 2002, 2004, pp. 1–4. https://doi.org/10.1029/2004GL019920.

[19] Aguttes, J. P., Grelier, T., Lestarquit, L., Péragin, E., Harr, J., Issler, J. L., Gerner, J. L., Garcia, A., Thevenet, J. B., Mehlen, C., Ensenat, C., Wilhelm, N., and Martinez, A. M. B., "A formation flying radio frequency technique and technology," *International Astronautical Federation - 59th International Astronautical Congress 2008, IAC 2008*, Vol. 11, No. August, 2008, pp. 6780–6787.

[20] Christian, J. A., "Relative navigation using only intersatellite range measurements," *Journal of Spacecraft and Rockets*, Vol. 54, No. 1, 2017, pp. 13–28. https://doi.org/10.2514/1.A33608.

[21] Friis, H., "A Note on a Simple Transmission Formula," *Proceedings of the IRE and Waves and Electrons*, 1946, pp. 254–256. https://doi.org/10.1109/JRPROC.1946.234568.

[22] Bachrach, J., and Taylor, C., *Handbook of Sensor Networks: Algorithms and Architectures*, 1st ed., Wiley, 2005. https://doi.org/10.1002/047174414X.ch9.

[23] Doherty, L., Pister, K. S., and El Ghaoui, L., "Convex position estimation in wireless sensor networks," *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 3, 2001, pp. 1655–1663. https://doi.org/10.1109/infcom.2001.916662.

[24] Biswas, P., and Ye, Y., "Semidefinite programming for ad hoc wireless sensor network localization," *Third International Symposium on Information Processing in Sensor Networks, IPSN*, Berkeley, California, 2004, pp. 46–54. https://doi.org/10.1145/984622.984630.

[25] Boyd, S., "CVX: Matlab Software for Disciplined Convex Programming [Online]," Available: http://cvxr.com/cvx/, 2020.

[26] Moore, D., "Robust Distributed Network Localization with Noisy Range Measurements," *Proceedings of ACM SenSys*, Baltimore, 2004, pp. 208–222. https://doi.org/10.1016/j.eaef.2017.03.002.

[27] Mautz, R., Ochieng, W., Brodin, G., and Kemp, A., "3D wireless network localization from inconsistent distance observations," *Ad-Hoc and Sensor Wireless Networks*, Vol. 3, No. 2-3, 2007, pp. 141–170. URL https://mautz.net/cv/Mautz_AHWSN.pdf.

[28] Fisher, R., "A Comparison of Four Algorithms for Estimating 3-D Rigid Transformations," *Proceedings of the British Machine Vision Conference*, Birmingham, 1995. https://doi.org/10.5244/C.9.24.

[29] Arun, K., Huang, T., and Blostein, S., "Least-squares fitting of two 3-D point sets," *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 9, No. 5, 1987, pp. 698–700. https://doi.org/10.1109/TPAMI.1987.4767965.

[30] Clohessy, W., and Wiltshire, R., "Terminal Guidance System for Satellite Rendezvous," *ARS Journal*, Vol. 31, No. 11, 1961, pp. 1516–1521. https://doi.org/10.2514/8.5842.

[31] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, Vol. 82, No. Series D, 1960, pp. 35–45. https://doi.org/10.1115/1.3662552.

[32] Ebinuma, T., Montenbruck, O., Lightsey, E. G., and De, E. M., "Precise Spacecraft Relative Navigation using Kinematic Inter-Spacecraft State Estimates," , No. January 2002, 2002, pp. 1–9. URL https://www.dlr.de/rb/Portaldata/38/Resources/dokumente/GSOC_dokumente/RB-RFT/ION_02B57.pdf.