
The Localization of Rigid Bodies via Multilateration
And Methods for the Estimation of Such in the Presence of Noise

PREPARED BY
JONATHAN COX

FOR
DARBHA SWAROOP
Texas A&M University



Introduction

The accurate localization of rigid a rigid with the minimum amount of available information remains one of the most pressing general problems in the field of engineering. Sensor implementations will always be limited, be it by current technological capabilities, specific application circumstances, or budget. Therefore a method is required that not only localizes rigid bodies, but also generates the highest quality localizations from the lowest quality inputs. This paper explores a prospective method for doing so, which consists of a trilateration arrangement augmented with a numerical least-squares based error reduction scheme. Additionally, supplementary velocity measurements are explored as a means of further error-correction via dead-reckoning.

Contents

1 Theory	3
1.1 Notes on Notation	3
1.2 Localizing the Body	4
1.2.1 Trilateration of a Single Point	4
1.2.2 Retrieving Rigid Body Orientation From Trilaterated Beacon Locations	5
1.3 Noise Handling	6
1.3.1 Modeling Noise	6
1.3.2 Localizing the Body in the Presence of Noise	6
1.3.3 Incorporating Velocity Measurements for Secondary Noise Reduction	8
1.4 Evaluating Estimation Methods	9
1.4.1 Positional Accuracy	9
1.4.2 Rotational Accuracy	10
2 Simulation Implementation Details	11
2.1 Utilized Standard Libraries	11
2.1.1 Scipy ^[3]	11
2.1.2 Numpy ^[3]	11
2.1.3 Random ^[4]	11
2.1.4 Matplotlib ^[5]	11
2.2 Euler-Angle Convention	11
2.3 Generated Input Values	12
2.4 Selection of Initial Orientations for Dead Reckoning and Gauss-Newton Optimization	13
3 Simulation Results	14
3.1 3-Anchor Trilateration Scheme	14
3.1.1 Characterization of a Single Simulation Run	14
3.1.2 Emergent Patterns from Repeated Simulation	15
3.2 4-Anchor Non-Planar Multilateration Scheme	16
3.2.1 Characterization of a Single Simulation Run	16
3.2.2 Emergent Patterns from Repeated Simulation	17
4 Discussion	18
4.1 Simulation Performance	18
4.1.1 Error Characterization	18
4.1.2 Estimation Method Comparison	18
4.2 Recommendations	19
4.3 Future Work	19
A Appendix	21
A.1 Supplementary Materials	21
A.2 Scripts	21

Section 1

Theory

1.1 Notes on Notation

This report strives to properly introduce notation as it appears and maintain consistent notation throughout its entirety. However, to alleviate any unavoidable ambiguity and to serve as a resource to the reader, this section will briefly and pre-emptively introduce common elements of the adopted standard notation.

- All symbols with a hat represent unit vectors. For example, \hat{a} .
- \mathbf{A}_n and \mathbf{B}_n denote the position vectors of the n^{th} anchor and beacon, respectively. Individual components of these vectors will typically be denoted with x, y, z subscripts, for example, \mathbf{B}_{1x} . However, on one occasion it is necessary to index the vector component via number. When two indices are present, the first index will always correspond to the anchor or beacon number, while the second will always refer to the vector component.
- Trilateration ranges are always denoted with the symbol l .
- The ground-frame and body-frame are denoted \mathcal{F}_g and \mathcal{F}_b , respectively.
- The generalized coordinate vector which fully defines \mathcal{F}_b is always denoted \mathbf{X} and takes the form $\mathbf{X} = [x_b \ y_b \ z_b \ \phi \ \theta \ \psi]^T$.
- The position vector of the origin of \mathcal{F}_b (in \mathcal{F}_g coordinates) is always denoted \mathbf{P} .
- Rotation matrices are always denoted \mathbf{R} .
- Rotation matrix components are always denoted r_{ij} .
- Coordinates or vector components expressed in \mathcal{F}_b are denoted with over-bars. For example, $\overline{\mathbf{B}_1x}$.
- Noisy values, or values derived from them, are denoted with prime notation. For example, \mathbf{B}'_1x or l'_{12} .

1.2 Localizing the Body

1.2.1 Trilateration of a Single Point

At base, trilateration (in 3D space) is a method for localizing a single particle, which we will refer to as the “beacon”, given its distance from at least three known locations, which we will refer to as “anchors”. The method has an intuitive geometric interpretation. The position of each anchor and its accompanying distance to the beacon together define a sphere. The points at which every such sphere intersect represent the only possible locations of the beacon. We can represent this relationship mathematically as a system of n equations

$$l_n^2 = (\mathbf{B}_x - \mathbf{A}_{nx})^2 + (\mathbf{B}_y - \mathbf{A}_{ny}) + (\mathbf{B}_z - \mathbf{A}_{nz}) \quad (1.1)$$

where the vector \mathbf{B} represents the position of the beacon, and each vector \mathbf{A}_n represents the position of the n^{th} anchor. The analytic solution for such a system can quickly become intractable. However, for the base case of three anchor points, a careful selection of the coordinate system results in a relatively simple standard solution. We select this coordinate system according to the following criteria:

- \mathbf{A}_1 lies on the origin.
- \mathbf{A}_2 lies along the x -axis.
- The y -axis is selected such that \mathbf{A}_3 lies within the xy -plane.

Such a selection is unique for three given anchors. In practice, this “trilateration frame” can be related to the desired ground frame via its corresponding homogeneous transformation matrix, enabling this method to be readily applied for arbitrary sets of three anchors regardless of their ground coordinates. For brevity, this derivation continues under the assumption that \mathcal{F}_g is coincident with this defined trilateration frame. Under these conditions, the system of equations which constrains the beacon location reduces to the following

$$l_1^2 = \mathbf{B}_x^2 + \mathbf{B}_y^2 + \mathbf{B}_z^2 \quad (1.2)$$

$$l_2^2 = (\mathbf{B}_x - \mathbf{A}_{2x})^2 + \mathbf{B}_y^2 + \mathbf{B}_z^2 \quad (1.3)$$

$$l_3^2 = (\mathbf{B}_x - \mathbf{A}_{3x})^2 + (\mathbf{B}_y - \mathbf{A}_{3y})^2 + \mathbf{B}_z^2. \quad (1.4)$$

This system has the two solutions, which are listed below.

$$\mathbf{B}_x = \frac{\mathbf{A}_{2x}^2 + r_1^2 - r_2^2}{2\mathbf{A}_{2x}} \quad (1.5)$$

$$\mathbf{B}_y = \frac{\mathbf{A}_{3x}^2 + \mathbf{A}_{3y}^2 - 2\mathbf{A}_{3x}\mathbf{B}_x + r_1^2 - r_3^2}{2\mathbf{A}_{3y}} \quad (1.6)$$

$$\mathbf{B}_z = \pm \sqrt{r_1^2 - \mathbf{B}_x^2 - \mathbf{B}_y^2}. \quad (1.7)$$

We note that the two solutions differ only in the sign of the z coordinate. This corresponds to a symmetry across the plane containing the anchors. Such a symmetry will occur for any number of coplanar anchors in 3D space. This can be explained via the geometric intuition of the trilateration process. We note that every sphere is symmetric about the plane containing the anchors. In the event that no anchor lies off of this plane, then the entire system, including the intersection points of the spheres, must be symmetric about this plane as well, providing an ambiguous solution. To resolve this, we restrict our coordinates to include only the positive z direction, and discard the solution which lies on the negative- z side of the plane.¹

¹Although this assumption is valid for a wide variety of applications, care must be taken when invoking it. If the desired application allows for the beacon to cross the trilateration plane, additional information is required to properly resolve the ambiguity.

1.2.2 Retrieving Rigid Body Orientation From Trilaterated Beacon Locations

The orientation of a rigid body in 3D space, including its position and rotation matrix, can be fully determined from three points, given that their locations are known both in the body frame, and in the frame of interest. The fully general formulation for this is presented below, although a fully general analytic solution is non-trivial. In light of this, a special case, for which the analytic solution is readily available, is presented afterwards as an example.

General Formulation

We have at our disposal two competing means through which to describe the orientation of a rigid body²:

- The set of six independent generalized coordinates $\mathbf{X} = [x_b \ y_b \ z_b \ \phi \ \theta \ \psi]^T$
- The translation vector and rotation matrix representation, $\mathbf{P} = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix}$, $\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$

Although these forms are mathematically equivalent, they present unique benefits in terms of workability. The generalized coordinate representation carries the benefit that each individual value more directly encodes information about the body's orientation, as they are all independent values. The rotation matrix, on the other hand, consists of non-independent values, but is more closely related to the behavior of the body. In this way, the generalized coordinates can be considered to be an abstraction of the vector-matrix form. The rotation matrix form is used in this formulation for brevity. To identify the orientation of the body in vector-matrix form, we desire to create a system of equations which will allow us to identify all components of \mathbf{X}_0 and \mathbf{R} , which constitutes twelve unknown values.

For each beacon, we possess both the ground-frame, $\mathcal{F}_g := \{\hat{I}, \hat{J}, \hat{K}\}$, and body-frame, $\mathcal{F}_b := \{\hat{i}, \hat{j}, \hat{k}\}$, coordinates as follows. We denote body-frame coordinate values with an over-bar.

$$\mathbf{B}_n = \overline{\mathbf{B}_{nx}}\hat{i} + \overline{\mathbf{B}_{ny}}\hat{j} + \overline{\mathbf{B}_{nz}}\hat{k} \quad (1.8)$$

$$\mathbf{B}_n = \mathbf{B}_{nx}\hat{I} + \mathbf{B}_{ny}\hat{J} + \mathbf{B}_{nz}\hat{K} \quad (1.9)$$

From this, we can construct the following matrix equation.

$$\mathbf{B}_n = \mathbf{R}\overline{\mathbf{B}_n} + \mathbf{P} \quad (1.10)$$

Given three beacons, this provides us with nine equations. The remaining three required equations can be derived from \mathbf{R} 's status as an orthonormal matrix. Each column within \mathbf{R} must represent a unit vector, providing us with three constraint equations of the form

$$1 = r_{1n}^2 + r_{2n}^2 + r_{3n}^2. \quad (1.11)$$

We now have a system of twelve equations which constrain the values of \mathbf{X} and \mathbf{R} . However, as noted, an explicit solution to this system is likely intractable.

²A third, closely related, representation exists in the form of homogeneous transformation matrices. This matrix-centric notation can provide computational efficiency gains on specialized hardware, but is not considered in this paper. For high performance implementations, all methods presented in this paper may be translated into homogeneous coordinate form without additional modification.

Special Case: Origin and Unit-Vector Beacons

We consider now the case where the three beacons have the following coordinates in \mathcal{F}_b

$$\begin{aligned}\overline{\mathbf{B}_1} &= 0\hat{i} + 0\hat{j} + 0\hat{k} \\ \overline{\mathbf{B}_2} &= 1\hat{i} + 0\hat{j} + 0\hat{k} \\ \overline{\mathbf{B}_3} &= 0\hat{i} + 1\hat{j} + 0\hat{k}.\end{aligned}$$

We note that trivially, $\mathbf{P} = \mathbf{B}_1$. Additionally, it follows that $\hat{i} = \mathbf{B}_2 - \mathbf{B}_1$ and $\hat{j} = \mathbf{B}_3 - \mathbf{B}_1$. We can now retrieve \hat{k} via $\hat{k} = \hat{i} \times \hat{j}$. Finally, we note that \mathbf{R} is the matrix that results from horizontally stacking the ground-frame expressions of the \hat{i} , \hat{j} , and \hat{k} column vectors, providing.

$$\mathbf{R} = \begin{bmatrix} \hat{i}_{\hat{i}} & \hat{j}_{\hat{i}} & \hat{k}_{\hat{i}} \\ \hat{i}_{\hat{j}} & \hat{j}_{\hat{j}} & \hat{k}_{\hat{j}} \\ \hat{i}_{\hat{k}} & \hat{j}_{\hat{k}} & \hat{k}_{\hat{k}} \end{bmatrix} \quad (1.12)$$

This showcases that a careful selection of beacon locations on the body can greatly simplify orientation calculations.

1.3 Noise Handling

1.3.1 Modeling Noise

All measurements in this analysis are considered to be subject to normally distributed noise which is proportional to the measured value such that it can be said with 95% confidence that the true value of the measurement lies within $\pm x\%$ of the measured value, where x is a user-defined value supplied to the simulation to control the simulated measurement accuracy.

This model captures the typical behavior of measurement uncertainty, that it typically represents some proportion of the true value and that repeated measurements can be assumed to be normally distributed around the true value. The mathematical expression defining this behavior is provided below, where $N(\mu, \sigma)$ denotes a randomly generated value from the normal distribution with mean μ and standard deviation σ .³

$$M_{\text{reported}} = N\left(M_{\text{true}}, \frac{xM_{\text{true}}}{z_{0.975}^+}\right) \quad (1.13)$$

1.3.2 Localizing the Body in the Presence of Noise

Incorporating the Rigid Body Assumptions

Upon introducing random noise into the trilateration range measurements, we may immediately note that it is possible, and in fact almost certain, that the trilaterated beacon locations resulting from the noisy measurements will not maintain the positional relationships to one another that they possess within the body frame. This observation motivates the following reasoning.

- The raw trilaterated points violate the rigid body assumption.
- The true body orientation properly maintains the rigid body assumption.
- Being normally distributed, we expect the noise from each range measurement to tend to “cancel” in some capacity as we take their sum.

We desire now to identify the proper body orientation which minimizes some measure of disagreement with the noisy ranges, under the assumption that the individual noise from each measurement will tend to cancel. The body orientation may be expressed via six independent coordinates, and so this task represents a nonlinear optimization problem in at least six dimensions. From here, we turn to numerical methods to enable continued analysis.

³ $z_{0.975}^+$ is the z -value corresponding to 95% confidence in a two-tailed z -test, which normalizes the distribution to the 95% confidence level.

The Gauss-Newton Algorithm

The Gauss-Newton is iterative numerical method for approximating the solution to n-dimensional optimization problems.[1] Given some vector function $\vec{\rho}(\mathbf{X})$ representing the error or residuals of a system, it minimizes the scalar quantity

$$S(\mathbf{X}) = \sum_i \rho_i(\mathbf{X})^2. \quad (1.14)$$

Starting with an initial guess, \mathbf{X}^0 , it updates the estimate after each iteration according to

$$\mathbf{X}^{n+1} = \mathbf{X}^n - (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \vec{\rho}(\mathbf{X}^n) \quad (1.15)$$

where \mathbf{J} is the Jacobian of $\vec{\rho}(\mathbf{X})$ with respect to \mathbf{X} .

$$\mathbf{J}_{ij}(\mathbf{X}) = \frac{\partial \rho_i(\mathbf{X})}{\partial \mathbf{X}_j} \quad (1.16)$$

The derivatives which comprise the individual elements of \mathbf{J} must also be numerically estimated. For this implementation, the derivatives have been evaluated using the complex-step method, which produces higher accuracy estimations than finite difference schemes for holomorphic functions when only the real-valued derivative is desired.[1]⁴

Selecting the Residuals Function, $\vec{\rho}(\mathbf{X})$

The accuracy, computational efficiency, and flexibility of the Gauss-Newton implementation hinge on the selection of the cost function $\vec{\rho}(\mathbf{X})$. Below, three different alternatives for the residuals are presented and discussed.

Selection 1. $\rho_i(\mathbf{X}) = ||\mathbf{B}_i(\mathbf{X}) - \mathbf{B}'_i||$ The naïve selection for residuals is the magnitude of the difference between each measured beacon location and its corresponding location in the estimated body orientation. Unfortunately, this method falters under scrutiny. In order to converge to a single solution, the Gauss-Newton algorithm requires that $\vec{\rho}$ be of equal or greater rank than \mathbf{X} . We see here that for the three beacon case, this provides us with $\text{rank}(\vec{\rho}) = 3$, while $\text{rank}(\mathbf{X}) = 6$. However, we also know that the three beacon case does fully define an orientation in the absence of noise. It must be possible to identify a better selection of $\vec{\rho}$.

Selection 2. $\rho_{ij}(\mathbf{X}) = |\mathbf{B}_{ij}(\mathbf{X}) - \mathbf{B}'_{ij}|$ To increase the rank of $\vec{\rho}$, we instead consider the residuals to be the disagreement between the individual components of the measured and estimated beacon locations. This provides $\text{rank}(\vec{\rho}) = 3 \cdot (\# \text{ of beacons})$, which is sufficient for convergence. However, this selection unfortunately also contains major pitfalls, although they are much more subtle. Upon inspection, we note that the mapping from measured ranges to Cartesian coordinates, $\mathcal{T} : \{l\}^3 \rightarrow \mathbf{B}$, is nonlinear. While the error is normally distributed about the true value in $\{l\}^3$ coordinates, we know that the corresponding distribution in cartesian coordinates is non-normal, and is not guaranteed to have an expected value corresponding to the true Cartesian coordinates of the beacon. In other words, we cannot rule out that the mapping $\mathcal{T} : \{l\}^3 \rightarrow \mathbf{B}$ may introduce bias error without substantial statistical analysis. In order to avoid this, we consider a third selection.

⁴For applications where highly specialized hardware or software is available, algorithmic differentiation is recommended instead.

Selection 3. $\rho_i(\mathbf{X}) = |l_i(\mathbf{X}) - l'_i|$ This selection of residuals aims to minimize the disagreement between each of the measured noisy trilateration ranges and the true value that that range would take on if the body were to occupy the orientation \mathbf{X} . This maintains the desired cancellation behavior from the normally distributed values of the trilateration ranges. Additionally, transforming the beacon coordinates into their associated trilateration ranges fully sidesteps the need to perform the reverse mapping altogether. This provides the estimation schema with an unparalleled level of extensibility.

Consider the case where, to increase estimation accuracy, a fourth anchor point is supplied. Calling on the geometric interpretation of trilateration, it is trivial to envision a scenario in which noisy measurements result in a set of four ranges which correspond to no location in \mathbb{R}^3 . In fact, it is almost guaranteed that such a set of measurements will be read. Suddenly, the introduction of information, intended to increase positional accuracy, has instead rendered the simple trilateration schema ineffective. The task of locating individual beacons in such an over-constrained system requires its own non-linear optimization algorithm. By avoiding the mapping $\mathcal{T} : \{l\}^3 \rightarrow \mathbf{B}$, we also avoid the need for such layered estimations. Furthermore, we also avoid the need for the initial \mathcal{F}_g to trilateration-frame coordinate conversion, meaning that \mathbf{X} may be estimated directly in the ground frame regardless of the anchor positions. With a system which easily accepts variable numbers of beacons and anchors we now achieve rank $(\vec{\rho}) = (\# \text{ of anchors}) \cdot (\# \text{ of beacons})$.

1.3.3 Incorporating Velocity Measurements for Secondary Noise Reduction

Dead Reckoning Estimation

In addition to multilateration schema, velocitymeters and gyrometers may be attached to the body to provide real time data about its translational and rotational velocities. Although estimations based on velocity data alone are subject to gradual error accumulation, they may be used as a supplement to the multilateration schema to increase accuracy and stability.

Dead Reckoning for Translation Data For translational dead reckoning of the body, we assume that the velocity of the body-frame origin remains constant between measurements. This provides the following expression for the new position of the body-frame origin, given its position and velocity at the previous time.

$$\mathbf{P}^{t_{s+1}} = \mathbf{P}^{t_s} + \Delta t \mathbf{V}^{t_s} \quad (1.17)$$

Dead Reckoning for Rotational Data For rotational dead reckoning of the body, we assume that the angular velocity of the body-frame remains constant between measurements. Each measurement then provides us with the angular velocity vector in body-frame coordinates.

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (1.18)$$

From this we can extract the angular speed, $\dot{\alpha}$, and the unit-vector corresponding to the axis of rotation, \hat{a} .

$$\dot{\alpha} = \|\omega\| \quad \hat{a} = \frac{\omega}{\|\omega\|} \quad (1.19)$$

We note that the total angle of rotation is $\Delta\alpha = \dot{\alpha}\Delta t$. We can now calculate the rotation matrix associated with rotation the body undergoes during the elapsed time using the Rodrigues rotation formula, where $[\hat{a}]_\times$ is the skew-symmetric cross product matrix associated with \hat{a} .

$$\Delta \mathbf{R}^{t_s} = \mathbf{I} + \sin(\Delta\alpha) [\hat{a}]_\times + (1 - \cos^2(\Delta\alpha)) [\hat{a}]_\times^2 \quad (1.20)$$

Finally, we can see that

$$\mathbf{R}^{t_{s+1}} = \Delta \mathbf{R}^{t_s} \mathbf{R}^{t_s}. \quad (1.21)$$

Reconciling Competing Estimation Methods

We now have two competing estimations of \mathbf{X} at each time step, the least-squares estimate, \mathbf{X}_{ls} , and the dead reckoned estimate, \mathbf{X}_{dr} . Reconciling the translations is simple. We simply define

$$\mathbf{P}_{rec} = \frac{1}{2} (\mathbf{P}_{ls} + \mathbf{P}_{dr}). \quad (1.22)$$

Reconciling the rotations requires a more sophisticated approach, however, as the elementwise average of two rotation matrices does not produce another rotation matrix. If we define this element-wise average as⁵

$$\mathbf{T}_{av} = \frac{1}{2} (\mathbf{R}_{ls} + \mathbf{R}_{dr}) \quad (1.23)$$

we may define \mathbf{R}_{rec} as the rotation matrix which is ‘nearest’ to \mathbf{T}_{av} in the Frobenius norm sense. That is to say, \mathbf{R}_{rec} is the orthogonal matrix which minimizes the quantity $\|\mathbf{T}_{av} - \mathbf{R}_{rec}\|$. This is the orthogonal Procrustes problem, and the solution may be identified via the singular value decomposition of \mathbf{T}_{av} .[2]

For any square matrix, A , it is possible to express that matrix in the form

$$A = U\Sigma V^T \quad (1.24)$$

where all three matrices are of the same shape as A , U and V are both orthonormal, and Σ is diagonal. The elements of Σ are referred to as the singular values of A . We now note that because U and V are orthonormal, their product will be as well. This product, the result of setting the singular values of A to 1, represents the “orthonormal component” of the matrix A , which is also closest orthonormal matrix to A in the Frobenius sense. We now see that when

$$\mathbf{T}_{av} = U\Sigma V^T. \quad (1.25)$$

The desired rotation matrix takes the form

$$\mathbf{R}_{rec} = UV. \quad (1.26)$$

1.4 Evaluating Estimation Methods

Now that we have developed three competing methods for estimating the orientation and position of the body, we now require a set of metrics by which to measure their efficacy. Two are proposed below.

1.4.1 Positional Accuracy

The proposed metric for positional accuracy is a percentage-error of the form

$$\%err_{\mathbf{P}} = 100 \left(\frac{\|\mathbf{P}_{true} - \mathbf{P}_{est}\|}{\|\mathbf{P}_{true}\|} \right). \quad (1.27)$$

This compares the magnitude of the “error vector” which connects the estimated body position to the true body position. Representing this measure as a percentage predisposes it for comparison to the percent error value used to mask the multilateration ranges, which makes it an excellent naïve heuristic for evaluating whether a given method increases or reduces the uncertainty of the position relative to that of the raw measurements.

⁵ \mathbf{T}_{av} is used to denote the average instead of \mathbf{R}_{av} to maintain clarity that it is not a rotation matrix.

1.4.2 Rotational Accuracy

Sensible methods of defining the rotational accuracy are more abstract. Being both cyclical in nature, and non-unique, Euler angle representations are unfit for use in calculating normalized accuracy metrics. Instead we must use the matrix representation of the rotation directly, which is unique for a specified ground frame. The proposed metric is thus

$$\text{err}_{\mathbf{R}} = \|\mathbf{R}_{\text{true}} - \mathbf{R}_{\text{est}}\|. \quad (1.28)$$

This represents a valid measure of “distance” between matrices for any valid matrix norm. However, its particular behavior will differ significantly based on the norm selected. For this application, the Frobenius norm is selected for its simplicity and its analogous properties to the euclidean vector norm used in the characterization of positional error.

Section 2

Simulation Implementation Details

2.1 Utilized Standard Libraries

2.1.1 Scipy^[3]

The Gauss-Newton algorithm used for solving the noisy localization optimization problem is implemented via Scipy's `scipy.optimize.least_squares` function, which handles the numerical differentiation internally as well. This function is directed to use the complex-step method to compute the Jacobian matrix, and is also fed the following constraints to force consistency with the modeling assumptions and notation.

$$z_b \geq 0 \quad -\pi \leq \phi \leq \pi \quad -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2} \quad -\pi \leq \psi \leq \pi$$

2.1.2 Numpy^[3]

`numpy` is utilized heavily for a wide variety of matrix and vector manipulations. Most notably of these, the singular value decomposition of \mathbf{T}_{av} is performed with `numpy.linalg.svd`.

2.1.3 Random^[4]

Two functions from `random` are used. `random.gauss` is used to generate normally distributed noise for masking measured values and `random.random` is used in the generation of randomly valued piecewise constant functions, which are used as the simulation inputs.

2.1.4 Matplotlib^[5]

`matplotlib` is utilized heavily to generate all plots and diagrams included in this report, as well as the accompanying supplementary animated simulation representations.

2.2 Euler-Angle Convention

In the sections above, any discussion of converting between \mathbf{X} and \mathbf{R} is conspicuously absent. This is because while \mathbf{R} is unique for any given inter-frame relationship, the Euler angles included in \mathbf{X} are not, and may take different values depending on convention. For the simulation accompanying this report, the Euler angles were taken to be the angles of rotation about the x , y , and z axes, in that order, so to describe $\mathbf{R}(\phi, \theta, \psi)$, we have

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad \mathbf{R}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad \mathbf{R}_z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$\mathbf{R}(\phi, \theta, \psi) = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.2)$$

$$\mathbf{R}(\phi, \theta, \psi) = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix}. \quad (2.3)$$

From here, we can develop a method for retrieving the $x - y - z$ Euler-angles from a given rotation matrix. An expression for θ is immediately apparent from r_{31} .

$$\theta = \sin^{-1}(-r_{31}) \quad (2.4)$$

Notably, this places the constraint on θ that $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$. Now knowing the value of θ , we can retrieve the values of ϕ from r_{32} and r_{33} as follows.

$$\phi = \text{atan2}(r_{32}, r_{33}) \quad (2.5)$$

Similarly, we note that

$$\psi = \text{atan2}(r_{21}, r_{11}). \quad (2.6)$$

This constraints ϕ and ψ to either of the ranges $[0, 2\pi]$ or $(-\pi, \pi]$, depending on the implementation of `atan2`. For this implementation, the range $(-\pi, \pi]$ is taken.

2.3 Generated Input Values

The velocity profiles of the body, both linear and angular, are randomly generated at the beginning of each simulation run based on a set of user-defined values. One independent profile is generated for each generalized coordinate. The user supplies four inputs:

- An interval which defines the range of possible values the function can take.
- A list of points spanning the desired domain at which the function may experience a discontinuity.
- The desired likelihood that a discontinuity will occur at each supplied point.
- A decimal value representing the magnitude of Gaussian noise to be applied, interpreted in accordance with section 1.3.1.

An example generated velocity profile is shown below.

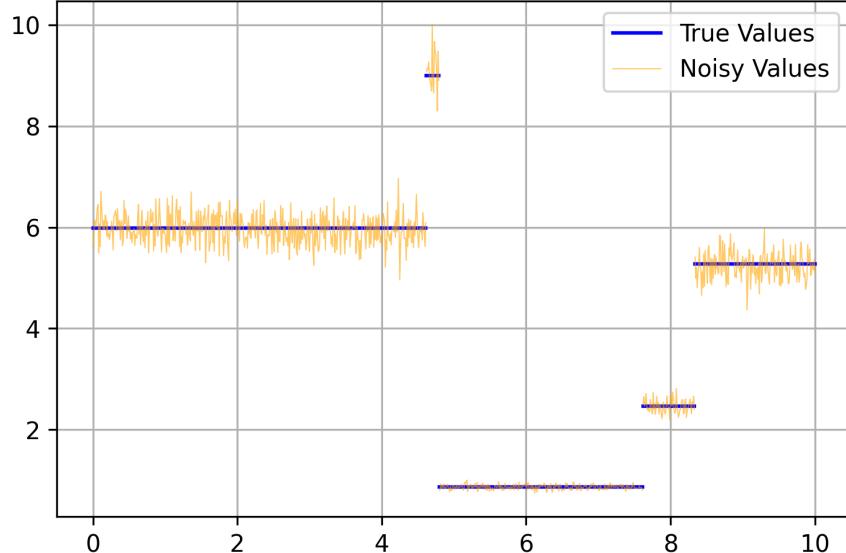


Figure 2.1: Example Random Velocity Profile

2.4 Selection of Initial Orientations for Dead Reckoning and Gauss-Newton Optimization

Dead reckoning methods require an initial “known” orientation from which to extrapolate future values. Somewhat similarly, iterative optimization algorithms require an initial estimate for the true solution on which to iterate. In both cases, the quality of the supplied initial value has a significant effect on the quality of the results. In this implementation the dead reckoning and gauss-newton algorithms are supplied with the same “initial value” at each time step. The starting orientation of the body is taken to be known, and so at $t = 0$, both methods are given the true location of the body. For each subsequent time step, each method is supplied the reconciled estimate value (defined in section 1.3.3) from the nearest preceding time step, as this is considered to be the nearest available estimate for both the body’s previous, in the case of dead reckoning, and current, in the case of iterative optimization, position.

Section 3

Simulation Results

3.1 3-Anchor Trilateration Scheme

The simulation was first conducted with three trilateration anchors, all located within the xy -plane. Noise on all measured quantities was set to 5%, in accordance with the convention introduced in section 1.3.1.

3.1.1 Characterization of a Single Simulation Run

The below figures showcase the evolution of both the linear and orientation error over time, as calculated by equations 1.27 and 1.28, respectively. Linear error is reported as a percentage-error with respect to the true position, while orientation error is reported in absolute terms. Average values are included in the figure legends.

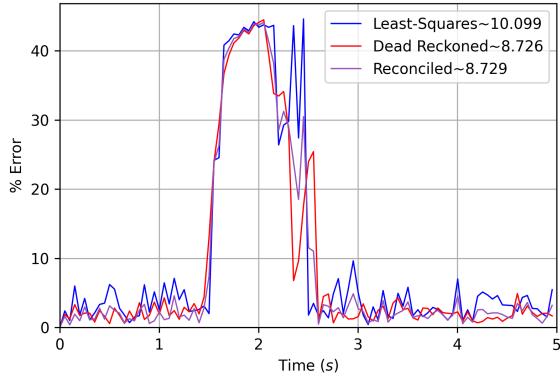


Figure 3.1: Percent Positioning Error for 3-Anchor Arrangement

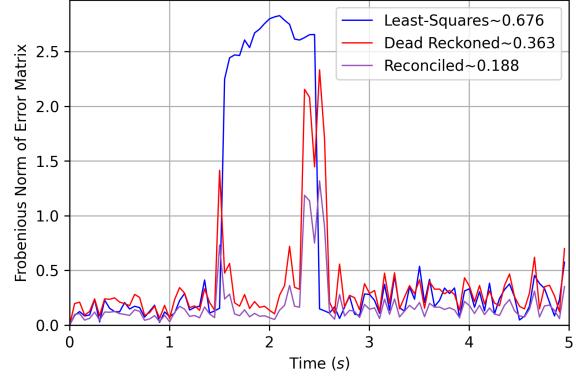


Figure 3.2: Rotational Error for 3-Anchor Arrangement

Several supplementary plots of the time-evolution of select generalized coordinates have been included below to provide deeper insight into the sharp spikes in both linear and rotational error.

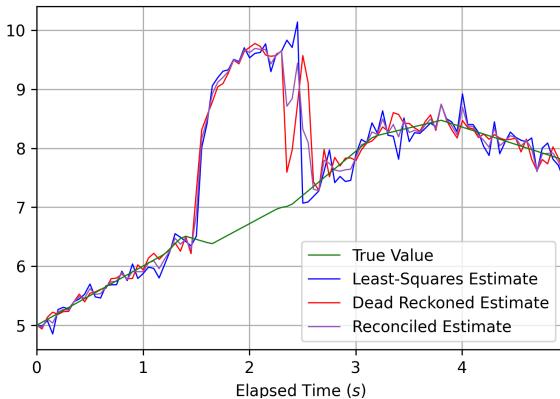


Figure 3.3: Time Evolution of the x Position Coordinate

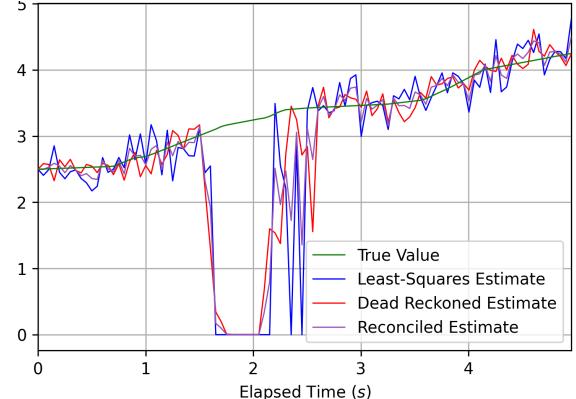


Figure 3.4: Time Evolution of the z Position Coordinate

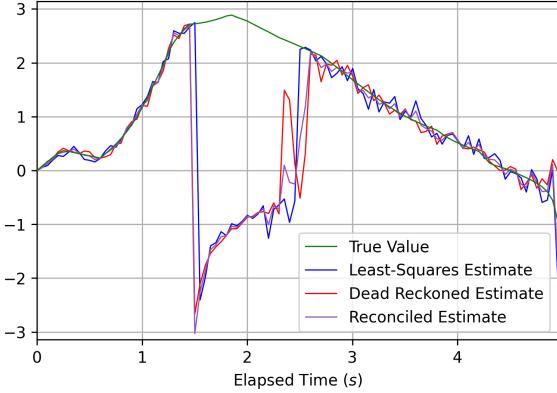


Figure 3.5: Time Evolution of the x -Axis Euler Angle

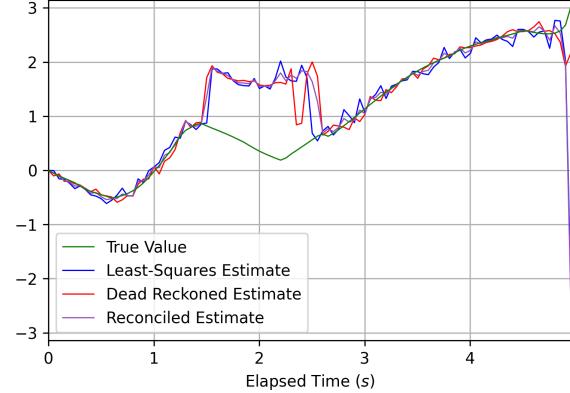


Figure 3.6: Time Evolution of the z -Axis Euler Angle

3.1.2 Emergent Patterns from Repeated Simulation

The simulation was repeated multiple times with different random inputs. Full plots describing the body and estimate behavior are omitted for brevity, but the average error values for each estimation method from each trial are tabulated below.

Table 3.1: 3-Anchor Simulation Average Errors

Trial#	Average % Positional Error			Average Rotational Error		
	LS	DR	RE	LS	DR	RE
1	10.099	8.726	8.729	0.676	0.363	0.188
2	3.862	2.286	2.288	0.208	0.253	0.129
3	9.505	8.042	8.045	0.596	0.392	0.203
4	3.188	1.846	1.847	0.167	0.194	0.097
5	3.531	2.079	2.080	0.168	0.200	0.100
6	4.620	3.095	3.102	0.296	0.342	0.178
7	4.738	3.423	3.428	0.285	0.244	0.126
8	3.612	2.162	2.163	0.168	0.202	0.102
9	10.147	8.841	8.846	0.590	0.265	0.134
10	3.132	1.798	1.800	0.171	0.198	0.099
Net Averages	5.643	4.230	4.233	0.333	0.265	0.136

3.2 4-Anchor Non-Planar Multilateration Scheme

Large spikes in error were a common occurrence throughout the 3-anchor simulation trials. Possible explanations are discussed in section 4.1. To explore a remedy for this, simulations were conducted with four anchors in a non-planar arrangement, each subject to the same level of noise. The results are presented below.

3.2.1 Characterization of a Single Simulation Run

The below figures showcase the evolution of both the linear and orientation error over time, as calculated by equations 1.27 and 1.28, respectively. Linear error is reported as a percentage-error with respect to the true position, while orientation error is reported in absolute terms. Average values are included in the figure legends.

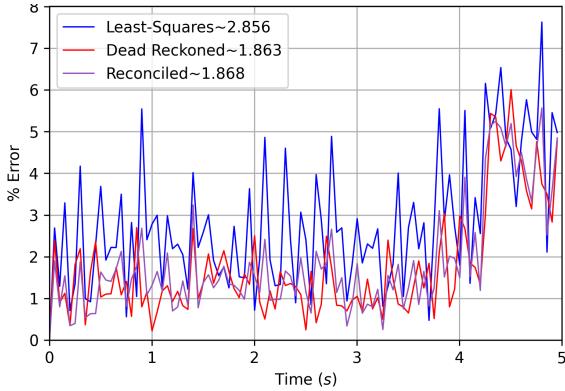


Figure 3.7: Percent Positioning Error for 4-Anchor Arrangement

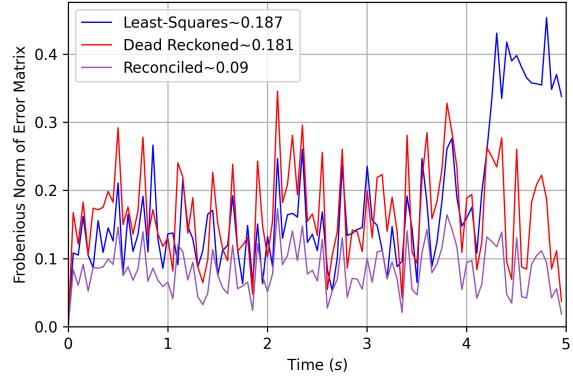


Figure 3.8: Rotational Error for 4-Anchor Arrangement

Several supplementary plots of the time-evolution of select generalized coordinates have been included below to provide comparison to the 3-anchor case.

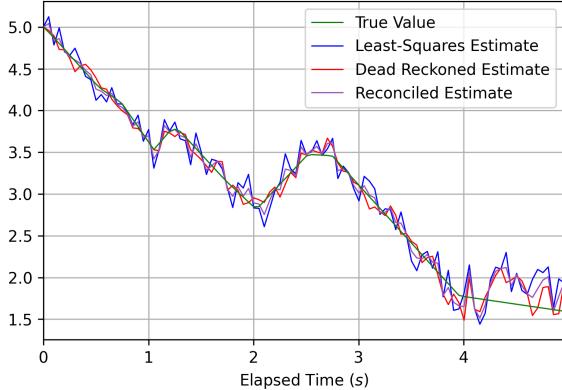


Figure 3.9: Time Evolution of the x Position Coordinate

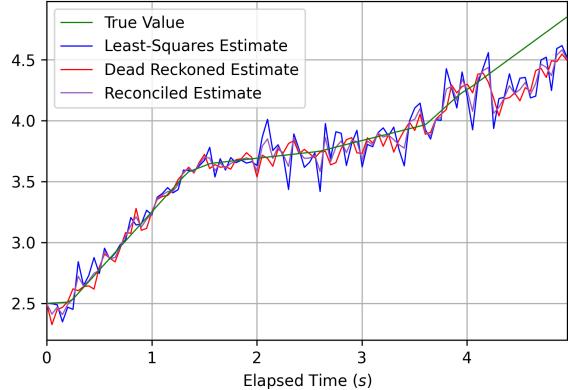


Figure 3.10: Time Evolution of the z Position Coordinate

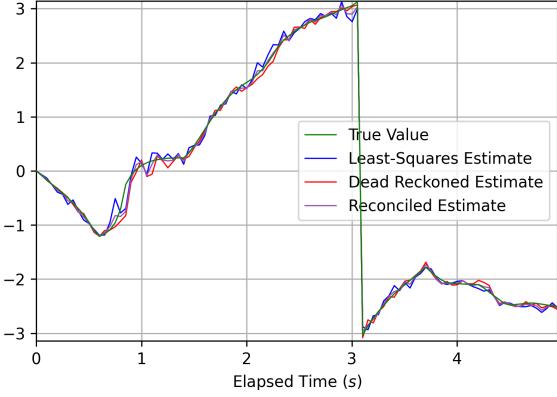


Figure 3.11: Time Evolution of the x -Axis Euler Angle

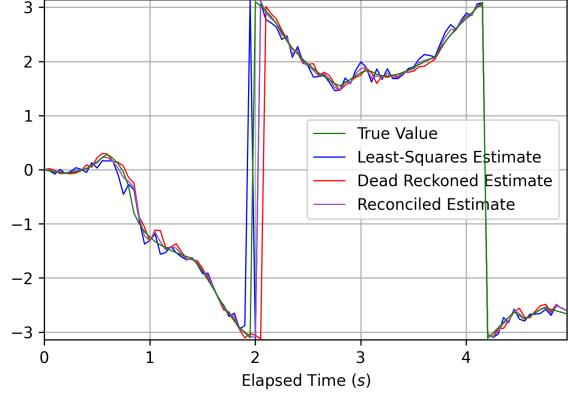


Figure 3.12: Time Evolution of the z -Axis Euler Angle

3.2.2 Emergent Patterns from Repeated Simulation

The simulation was repeated multiple times with different random inputs. Full plots describing the body and estimate behavior are omitted for brevity, but the average error values for each estimation method from each trial are tabulated below.

Table 3.2: 4-Anchor Simulation Average Errors

Trial#	Average % Positional Error			Average Rotational Error		
	LS	DR	RE	LS	DR	RE
1	2.856	1.863	1.868	0.187	0.181	0.090
2	2.797	1.603	1.605	0.138	0.162	0.081
3	3.079	2.155	2.158	0.275	0.373	0.205
4	2.454	1.474	1.477	0.177	0.197	0.099
5	2.527	1.476	1.478	0.154	0.179	0.090
6	2.737	1.557	1.558	0.140	0.166	0.083
7	2.350	1.377	1.379	0.159	0.187	0.093
8	2.643	1.626	1.632	0.169	0.175	0.087
9	2.867	1.870	1.872	0.169	0.154	0.077
10	2.695	1.549	1.550	0.133	0.160	0.080
Net Averages	2.701	1.65	1.658	0.170	0.230	0.099

Section 4

Discussion

4.1 Simulation Performance

4.1.1 Error Characterization

The behavior displayed in section 3.1.1 is emblematic of the shortcomings of iterative solvers in conjunction with coplanar anchor configurations. These sudden error spikes are the result of the iterative method closing in on the ambiguous trilateration solutions that lie mirrored across the anchor plane. Figure 3.1.1 serves as the most compelling evidence for this. The solver is constrained to operate only in the positive- z region; when it is captured by the ambiguous solution, it moves downwards until it reaches $z = 0$ and stabilizes there, not being permitted to reduce the z -coordinate any further. This behavior produces the flat-line $z = 0$ value that frequently accompanies these extreme-error regions. These error spikes most frequently occur shortly after the value of one of the Euler angles wraps around to the opposite sign, as these regions represent the domains in which the target value differs greatest from the supplied initial-guess, providing the greatest window of opportunity for the solver to drift to the false optimum location.

This problem is largely resolved in the 4-anchor configuration. The existence of a non-coplanar anchor eliminates the ambiguous solutions, resulting in a single global optimum for the least-squares method to converge to. Moderate-to-large spikes in error still occur, as the method may occasionally converge to incorrect local minimums, but both the frequency and recovery time of these spikes are greatly reduced. In conjunction with the generally higher accuracy granted by the additional three measurements per-estimate, this reduction in error spikes explains the significant reduction in average error between the 3-anchor and 4-anchor configuration as seen in tables 3.1.2 and 3.2.2.

4.1.2 Estimation Method Comparison

Tables 3.1.2 and 3.2.2 additionally reveal interesting trends about the relationships between the error rates of the different estimation methods as well. In both the 3 and 4-anchor case, it can be seen that the dead reckoned and reconciled positional error are nearly identical with both being consistently lower than the least-squares positional error. Upon initial inspection, this might suggest that it is unnecessary to calculate the reconciled positional error. This is unfortunately not the case, however, as the reconciled estimate for translation and its use as the basis for the dead reckoning estimate are necessary to prevent cumulative drift.

For rotational error, it can be seen that the reconciled error is significantly lower than both the least-squares and dead reckoned estimate with startling consistency. This seems to indicate that the true orientation is highly likely, or possibly guaranteed, to lie “between” these two estimates in some sense. Further investigation is warranted into the relationship between these values as a means for generating more accurate estimates.

4.2 Recommendations

- Multilateration noise reduction methods based on numerical optimization should utilize cost functions which map their inputs into $\{l\}^n$, the coordinate space of the measured multilateration ranges. This technique best leverages the normally distributed nature of measurement error while also providing a robust algorithm which can be unilaterally applied to multilateration configurations of arbitrary complexity.
- The inclusion of a fourth non-planar anchor in the multilateration scheme should be a high priority when the application is time-sensitive. Although mathematical tools exist to ensure the three beacon configuration will eventually converge to the proper global optimum, they represent significant additional computational expense, and may interfere with the ability of the system to interpret real-time data at an acceptable rate.
- Gyrometers should be considered as a budget alternative to higher-accuracy multilateration equipment for increasing the accuracy of the estimate body orientation, as the dead reckoning based corrections show significant improvement on orientational accuracy, particularly when the body is expected to move far away from the multilateration anchors. Velocitymeters can be used similarly to increase positional accuracy, although the improvement is less pronounced.

4.3 Future Work

Several lines of work present themselves as opportunities to further refine estimation methods:

- The effect of noise within multilateration ranges on the resultant located point coordinates depends heavily on the positioning of the multilateration anchors. Studies to identify optimal anchor locations could minimize the effect of measurement error, resulting in higher numerical stability without additional computation.
- The distance of beacons from the body-frame origin is inversely related to the magnitude of the effect that a given displacement of that beacon will have on the orientation of the body. Further investigation of this relationship could yield methods for beacon-placement or body-frame selection that increase the numerical stability of the estimated rotation without additional computation.
- The implemented scheme for reconciling competing rotation matrices is based on the orthogonal Procrustes problem, and provides equal weight to both measurements. Wahba’s problem represents a method for generating a “weighted average” rotation matrix. Investigation into implementing Wahba’s method as well as methods for selecting weights for the least-squares and dead reckoned estimates, likely based on their corresponding estimated uncertainties, could result in even greater reductions in orientation error.[6]

Bibliography

- [1] Joaquim R. R. A. Martins and Andrew Ning. *Engineering Design Optimization*. Cambridge University Press, 2021. ISBN: 9781108833417.
- [2] Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. 3600 University City Science Center, Philadelphia, PA 19104-2688: Society for Industrial and Applied Mathematics, 1997. ISBN: 0-89871-361-7.
- [3] The Scipy Community. 2022. URL: <https://docs.scipy.org/doc/scipy/>.
- [4] The Python Software Foundation. 2022. URL: <https://docs.python.org/3/>.
- [5] John Hunter et al. The Matplotlib Development Team. 2022. URL: <https://matplotlib.org/stable/index.html>.
- [6] Bisharah Libbus, Gordon Simons, and Yi-Ching Yao. “Rotating Multiple Sets of Labeled Points to Bring Them Into Close Coincidence: A Generalized Wahba Problem”. In: *The American Mathematical Monthly* 124 (), pp. 149–160. URL: <https://www.jstor.org/stable/10.4169/amer.math.monthly.124.2.149>.

Appendix

A.1 Supplementary Materials

Included alongside the attached scripts are two animations depicting the motion of the simulated bodies and their estimated positions from sections 3.1.1 and 3.2.1, which may aid in developing an intuitional understanding of the data from these sections.

A.2 Scripts

The scripts, as well as pre-generated demo outputs, are included in this report in the form of the below attachment. (Double click to open)

[Python Module for Multilateration Based Body Localization](#)