

Stable Navigation Solutions for Robots in Complex Environments[†]

C. Scraper, R. Madhavan[‡], S. Balakirsky

National Institute of Standards and Technology (NIST)

Gaithersburg, MD 20899-8230, USA.

{chris.scraper, raj.madhavan, stephen.balakirsky}@nist.gov

Abstract—During the initial phase of a disaster response it is essential for responders to quickly and safely assess the overall situation. The use of rescue robots that can autonomously navigate and map these environments can help responders realize this goal while minimizing danger to them. In order for rescue robots to be of service to the responders, they must be able to sense the environment, create an internal representation that identifies victims and hazards to responders, and provide an estimate of where they are and where they have been. Methods for developing a stable navigation solution are based on sensors that can be broadly classified into two approaches, absolute (exteroception) and relative (proprioception). Commonly, two or more of these approaches are combined to develop a stable navigation solution that is insensitive to and robust in the presence of the errors that plague partial solutions by taking into account errors in the vehicle’s pose, thus bounding the uncertainty in the navigation solution. Since the capabilities and limitations of these approaches vary, it is essential for developers of robotic systems to understand the performance characteristics of methodologies employed to produce a stable navigation solution. This paper will provide quantitative analysis of two proprioceptive approaches, namely Encoder-based Odometry and Inertial Navigation System, and an exteroceptive approach namely Visual Odometry that uses scan matching techniques.

Keywords: Navigation Solutions, Visual Odometry, Scan Matching, Dead-Reckoning, ICP, MOAST, USARSim, Search and Rescue.

I. INTRODUCTION

Urban Search And Rescue (USAR) is primarily concerned with the extrication of victims trapped in man-made structures such as collapsed buildings. During the initial phase of a structural collapse rescue, it is imperative that the first responders “size-up” the situation and establish an Incident Control System that allows information to flow regarding the nature of the problem. During this size-up, reconnaissance teams are dispatched to assess the magnitude of the situation, identify hazards, and locate areas that have the lowest cost-benefit ratio of danger to rescuers versus live victims [1]. Commonly, these environments contain unstable structures, undulating terrain, and hazardous or toxic debris. Recent advancements in

autonomous navigation and mapping provide responders with an invaluable tool that can allow them to safely and efficiently assess the overall situation.

A robotic system, consisting of one or more robots, capable of autonomous navigation and mapping must provide the mechanisms for answering three fundamental questions: “Where am I?”, “Where do I want to go?”, and “How do I get there?” [2]. In order for the system to intelligently answer the second two questions, the system must be able to formulate a stable navigation solution that can answer the first. In this paper, the notion of a *navigation solution* will be defined as *the systems ability to sense the environment, create internal representations of its environment, and estimate pose, consisting of position and orientation, with respect to a fixed coordinate frame.*

Mobility Open Architecture Simulation and Tools (MOAST) [3] is an open-source, turn-key hierarchical control system capable of autonomous navigation and mapping for a wide range of robotic platforms in a variety of different domains. Since the methods employed to formulate a stable navigation solution are heavily dependent on the type of environment the system is operating in, the sensor capabilities, and the conditions found in that environment [4], it is critical for MOAST to employ redundant methods of pose estimation in order to develop a robust and stable navigation solution.

Dead-reckoning is a widely used method for pose estimation that serves as the backbone for many navigation solutions. It is based on simple mathematical principles that estimate the current pose based on a previous pose. This method “advances” the pose estimate by recursively integrating motion, measured through a proprioceptive sensor, to compute a new heading and the distance traveled. Dead-reckoning is favorable because it provides a simple, cost-effective solution that is self-contained and is capable of computing pose estimates at a high frequency. The major drawback to dead-reckoning is two-fold: 1) because the system is self-contained, *systematic* and *non-systematic* dead-reckoning errors [5] are hard to eliminate and 2) the recursive nature of the algorithms allows errors to propagate and accumulate in a unbounded manner, thus causing the navigation solution to diverge [4].

Many navigation solutions use a landmark-based approach [2], [6], [7]. This approach geometrically computes an estimate based on the recognition of distinct features, occurring naturally or artificially placed, in the environment. The factors

[†]Commercial equipment and materials are identified in this paper in order to adequately specify certain procedures. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

[‡]Research Staff Member, Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA.

contributing to the successful performance and integrity of these methods is the reliable acquisition and extractions of features from sensory data and the ability to efficiently recognize and associate features with some navigational map [2]. While these methods, in general, provide an accurate pose estimate, they require either engineering the environment to provide an adequate set of features, or efficient recognition of features to use as landmarks [8]. In addition, these methods often rely on geometric primitives or models of the environment that are not guaranteed to exist in all environments.

In lieu of the landmark-based approaches, the iconic approaches attempt to utilize whatever sensor data is available to compute a navigation solution by working directly with raw sensor data. This eliminates the need to decide what constitutes a feature by minimizing the discrepancies between the raw sensor data and a model of the environment. Using a maximum likelihood alignment to find the best fit between two sets of data points, this method is capable of providing a computationally efficient pose estimate in complex, unstructured environments. Examples of iconic-based methods in the literature are [9], [10], [11].

The motivation of this research is to develop a stable navigation solution that meets the navigational requirements for MOAST without any unnecessary overhead or computational complexities. This research uses a high-fidelity robotic simulation environment, known as *Urban Search and Rescue Simulation* (USARSim) [12], to explore the performance characteristics of different navigational solutions. In the interest of space, Section II will only present an overview of MOAST and USARSim. For further details please refer to system manual for MOAST [3] and USARSim [12]. Section III will detail the principles and known deficiencies of three navigation solutions and will provide an overview of the implementation of these systems in MOAST and USARSim. Although the navigation solutions presented here are usually combined to provide a more stable navigation solution, this paper will treat them as an independent solution in order to better understand the performance characteristics of each. This will facilitate a better understanding of the capabilities and, more importantly, the limitations of each solution. In Section IV, the methods employed and the results of the independent evaluation of each of the solutions will be presented. Section V will provide a discussion of the results obtained and how each of the partial solutions can be used to develop a more robust and stable navigation solution.

II. MOAST AND USARSIM

Robotic simulation systems, such as [13], [14], [15], are commonly used in the development of the autonomous systems and advanced robotic algorithms. They provide a cost-effective tool that enables developers to customize repeatable testing scenarios to test specific aspects of autonomous navigation and mapping systems. In order to provide convincing arguments about a system's performance and reliability, the simulation systems must be capable of capturing the stochastic nature

of a real world environment. USARSim [12] is an open-source package that provides a high-resolution, physics-based simulation that solves many of the practical problems faced by robotic simulators [16]. Initially developed to support development of robotic algorithms in the Urban Search and Rescue environment, USARSim has expanded its core functionality to provide the general-purpose, multi-agent simulation system with a set of unique characteristics unmatched by other simulation systems [17], [16].

MOAST [3] is an open-source, turn-key hierarchical control system that was originally developed to promote the research of advanced robotic algorithms [18]. Based on the *4-D Real-time Control System* (4D/RCS) architecture [19], MOAST provides a modularized hierarchical framework that allows for the transparent transference of data between a matrix of real and virtual components. This framework is glued together through well-defined interfaces and communications protocols, and detailed specifications on individual subsystem input/output (I/O) that allows developers to freely swap components. Internal tools provide developers with state-by-state, time-stamped snapshots that allow researchers to quantitatively measure and classify the performance characteristics of new algorithms and the means to analyze the overall impact on the system's performance by means of comparison.

Since the validity of the results obtained from such algorithms are directly related to the accuracy and realism of the underlying simulation models, it is important that the sensors provide realistic data. Significant efforts on the validation of simulated models in USARSim have resulted in close correspondence between simulated data extracted from USARSim and their real world counterparts [20], [21]. Therefore, integration of these high-fidelity models with MOAST allows researchers to develop advanced robotic algorithms, classify their performance characteristics, and evaluate the overall impact of the algorithms on a robotic system before implementation on real robotic hardware.

III. NAVIGATION SOLUTIONS

This section will present three navigation solutions; namely, an Inertial Navigation System solution, an Encoder-based Odometry solution, and a Visual Odometry solution that is based on a 2D laser range-finder. All of these solutions provide light-weight navigation solutions that are being considered for the navigation strategy employed by MOAST. Each of the following subsections will provide an overview of the solutions, identify their deficiencies, and provide an overview on how they are implemented in MOAST and USARSim.

A. Inertial Navigation System

An *Inertial Navigation System* (INS) is a dead-reckoning navigation solution based on Newton's laws of motion. It assumes that an object will remain in uniform motion unless it is acted on by an outside force. Forces acting on the system will produce accelerations in an inertial reference frame that can be measured and integrated. Change in position and orientation can then be computed with respect to a

previously determined pose [22]. The output of an Inertial Navigation System is a 3D pose estimate at time k , consisting of a location $[x_{v_k}, y_{v_k}, z_{v_k}]$ and an orientation $[\theta_{v_k}, \psi_{v_k}, \phi_{v_k}]$. Unfortunately, inertial sensors are not able to dissociate accelerations due to external forces, such as gravity, from kinematic accelerations [23]. The failure of the systems to misclassify or compensate for non-kinematic noise will produce error in the inertial measurements. The double integration of this error will produce a gradual degradation of the navigation solution known as drift.

A method for simulating an Inertial Navigation System has been developed for USARSim. In an effort to mimic the performance characteristics of an INS sensor, this method uses a recursive pose estimate that uses error models to introduce noise into inertial measurements.¹ Given an initial estimation of location $[\hat{x}_{v_k}, \hat{y}_{v_k}, \hat{z}_{v_k}]$ and an initial estimation of orientation $[\hat{\theta}_{v_k}, \hat{\psi}_{v_k}, \hat{\phi}_{v_k}]$, the algorithmic process of simulating the INS sensor in USARSim is as follows:

- 1) Compute angular velocities $[\omega_{x_{k+1}}, \omega_{y_{k+1}}, \omega_{z_{k+1}}]$, for current time step, $k + 1$, using ground truth:

$$\begin{bmatrix} \omega_{x_{k+1}} \\ \omega_{y_{k+1}} \\ \omega_{z_{k+1}} \end{bmatrix} = \frac{1}{\Delta t} \left(\begin{bmatrix} \theta_{v_{k+1}} \\ \psi_{v_{k+1}} \\ \phi_{v_{k+1}} \end{bmatrix} - \begin{bmatrix} \theta_{v_k} \\ \psi_{v_k} \\ \phi_{v_k} \end{bmatrix} \right) \quad (1)$$

- 2) Use angular velocities computed in Eq. 1 and a Gaussian noise model, $G(\mu, \sigma)$, that is based on a mean, μ , and variance, σ , to update the estimation of orientation, given the initial estimate. :

$$\begin{bmatrix} \hat{\theta}_{v_{k+1}} \\ \hat{\psi}_{v_{k+1}} \\ \hat{\phi}_{v_{k+1}} \end{bmatrix} = \begin{bmatrix} \hat{\theta}_{v_k} \\ \hat{\psi}_{v_k} \\ \hat{\phi}_{v_k} \end{bmatrix} + \begin{bmatrix} \omega_{x_{k+1}} \\ \omega_{y_{k+1}} \\ \omega_{z_{k+1}} \end{bmatrix} G(\mu, \sigma) \Delta t \quad (2)$$

- 3) Calculate an estimate of the Euclidean distance traveled over the past time step using ground truth and noise model, $G(\mu, \sigma)$:

$$\hat{V}_{dist} = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} G(\mu, \sigma) \quad (3)$$

- 4) Use results obtained from Equation 2 and 3 to update the current location estimate:

$$\begin{bmatrix} \hat{x}_{v_{k+1}} \\ \hat{y}_{v_{k+1}} \\ \hat{z}_{v_{k+1}} \end{bmatrix} = \begin{bmatrix} \hat{x}_{v_k} \\ \hat{y}_{v_k} \\ \hat{z}_{v_k} \end{bmatrix} + \begin{bmatrix} \hat{V}_{dist} \cos \hat{\phi}_{v_{k+1}} \cos \hat{\psi}_{v_{k+1}} \\ \hat{V}_{dist} \sin \hat{\phi}_{v_{k+1}} \cos \hat{\psi}_{v_{k+1}} \\ \hat{V}_{dist} \sin \hat{\phi}_{v_{k+1}} \end{bmatrix} \quad (4)$$

B. Encoder-based Odometry

Encoder-based Odometry is a commonly used dead-reckoning approach that uses encoders to measure wheel rotation and steering angle to kinematically compute a pose estimate based on wheel radius, wheel separation, and wheel base. The output of an odometric sensor is usually a 2D pose at time k , expressed as a triplet consisting 2D location and orientation, $[x_{v_k}, y_{v_k}, \phi_{v_k}]$. The major drawback to this solution is that *systematic* and *non-systematic* errors [5] are

hard to eliminate and can accumulate over time. Systematic errors accumulate constantly over time due to unequal wheel diameters, misalignment of wheels, encoder sampling rates, drifts associated with time etc., whereas non-systematic errors are as a result of wheel slippage and uneven terrain conditions that may occur unexpectedly. For a vehicle navigating on uneven terrain the non-systematic errors are more dominant than the systematic errors.

Encoder-based Odometry was the first navigation sensor developed for USARSim. The current implementation, which only simulates odometric sensors for skid-steered platforms, relies on the simulation of physical interactions to introduce non-systematic errors. Since systematic error models have not been incorporated in this implementation of the sensor, the simulated version in USARSim will provide a more accurate pose estimate than actual Encoder-based Odometry found on real robotic platforms.

This simulation of the Odometry sensor in USARSim is as follows:

Given an initial pose estimate at time k , $[\hat{x}_{v_k}, \hat{y}_{v_k}, \hat{\phi}_{v_k}]$, and the distance traveled by the left and right tires/tracks over the past time step, $[U_{l_{k+1}}, U_{r_{k+1}}]$, compute a new pose estimate based on a skid-steered kinematic model shown in Equation 5, where ℓ represents the wheel separation.

$$\begin{bmatrix} \hat{x}_{v_{k+1}} \\ \hat{y}_{v_{k+1}} \\ \hat{\phi}_{v_{k+1}} \end{bmatrix} = \begin{bmatrix} \hat{x}_{v_k} \\ \hat{y}_{v_k} \\ \hat{\phi}_{v_k} \end{bmatrix} + \begin{bmatrix} \frac{U_{l_{k+1}} + U_{r_{k+1}}}{2} \cos \hat{\phi}_{v_k} \\ \frac{U_{l_{k+1}} + U_{r_{k+1}}}{2} \sin \hat{\phi}_{v_k} \\ \tan^{-1} \frac{U_{l_{k+1}} - U_{r_{k+1}}}{\ell} \end{bmatrix} \quad (5)$$

C. Visual Odometry

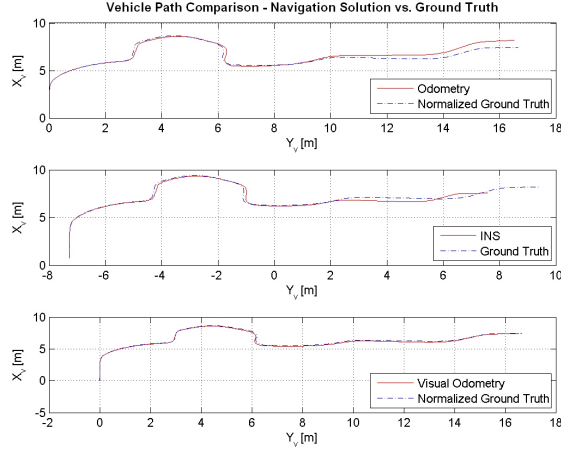
Visual Odometry (VO) is a commonly used scan matching technique [10], [9], [25], [26] that uses exteroceptive sensors. VO is an important intermediate step that will lead to an absolute navigation solution. Absolute navigation solutions have the advantage of being independent of the errors that arise in relative navigation solutions (i.e. INS and encoder-based schemas discussed in Sections III-A and III-B), thus providing a method for keeping the resulting errors bounded.

This iconic navigation solution is obtained by using a *fine range image registration method* [27] known as Iterative Closest Point (ICP) algorithm [28]. ICP uses pairs of consecutive sets of data points obtained from a 2D laser range finder (scans) to compute relative pose estimates. This technique provides a computationally efficient navigation solution in environments with minimal structure. However, poor point-correspondence and the lack of distinguishing features in the data sets can lead to erroneous pose estimates.

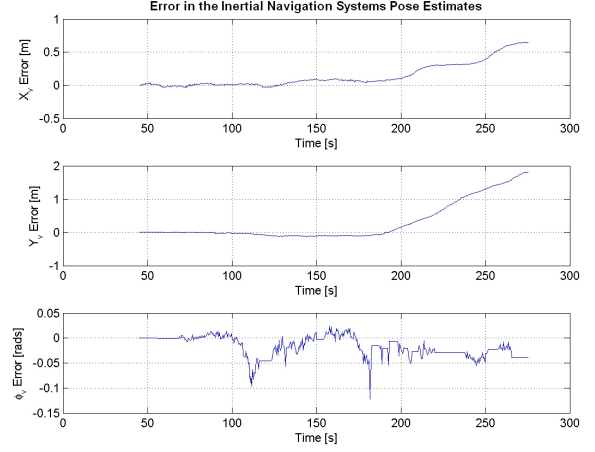
In its simplest form, the ICP algorithm can be described by the following steps:

1. For each point in data set \mathbf{D} , compute its closest point in data set \mathbf{M} .
2. Compute the incremental transformation (\mathbf{R}, \mathbf{T}) using Singular Value Decomposition (SVD) based on correspondences obtained in step 1.

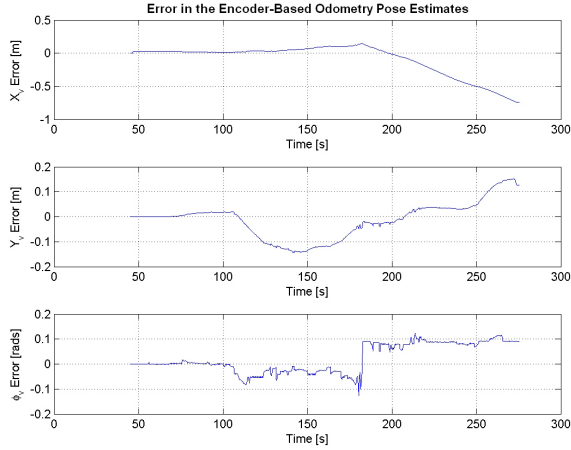
¹The current implementation in USARSim does not take accelerations into account. However, we are currently investigating other methods to model the errors in simulated INS sensor, using equations defined by [24].



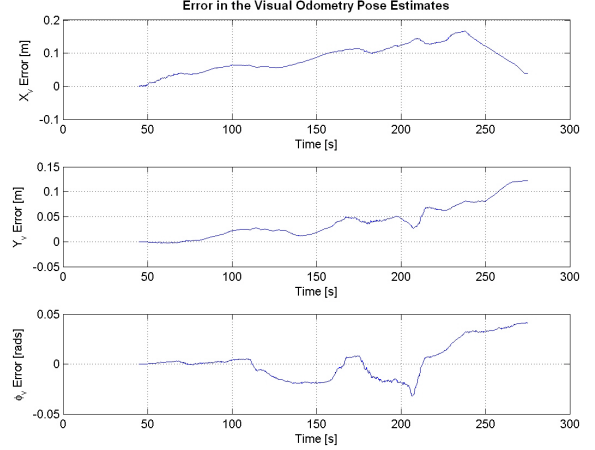
(a) Comparison of the vehicle path produced by navigation solutions and ground truth information in a flat, feature-rich world. As seen, the Visual Odometry navigation solution provides a better estimate of the vehicle's location.



(b) The decomposition of the INS navigation solution shows the accumulation of the errors and the slow degradation of the individual elements on the pose estimate over time (drift).



(c) Quantitative analysis of errors found in the Encoder-based Odometry solutions shows that this solution provides a relatively accurate solution in environment with flat flooring and other factors that cause non-systematic errors.



(d) Quantitative analysis of errors in the Visual Odometry navigation solution. As seen, this exteroceptive approach provides an extremely accurate estimation of the pose that out performs the other two dead-reckoning solutions.

Fig. 1: Estimated vehicle paths and associated errors obtained from the three navigation solutions. It is evident that the “absolute” Visual Odometry solution is superior to the encoder-based Odometry and INS solutions

3. Apply the incremental transformation from step 2 to \mathbf{D} .
4. If relative changes in \mathbf{R} and \mathbf{T} are less than a predetermined threshold or a tolerable number of iterations is exceeded, terminate. Else go to step 1.

In step 1, the set $\{x_i, y_i, d_i\}$ is computed consisting of original points in \mathbf{M} , their nearest neighbor in set \mathbf{D} , and the Euclidean distance between the two. To deal with spurious points/false matches and to account for occlusions and outliers, we modify and weight the least-squares objective function such that [29]:

$$\min_{(\mathbf{R}, \mathbf{T})} \sum_i w_i \|\mathbf{M}_i - (\mathbf{R}\mathbf{D}_i + \mathbf{T})\|^2 \quad (6)$$

If the Euclidean distance between a point x_i in one set and its closest point y_i in the other, denoted by $d_i \triangleq d(x_i, y_i)$, is

bigger than the maximum tolerable distance threshold \mathcal{D}_{max} , then w_i is set to zero in Equation (6). This means that an x_i cannot be paired with a y_i since the distance between reasonable pairs cannot be very big. The value of \mathcal{D}_{max} is set adaptively in a robust manner by analyzing distance statistics. The adaptive threshold is implemented with respect to two observations [29]:

- 1) If \mathcal{D}_{max} is too small, then several iterations are required for the algorithm to converge and several good matches will be discarded, and
- 2) If \mathcal{D}_{max} is too big, then the algorithm may not converge at all since many spurious matches will be included.

At the end of step 1, we have two corresponding point sets, $\mathbf{P}_M: \{\mathbf{p}_i\}$ and $\mathbf{P}_D: \{\mathbf{q}_i\}$. The incremental transformation in

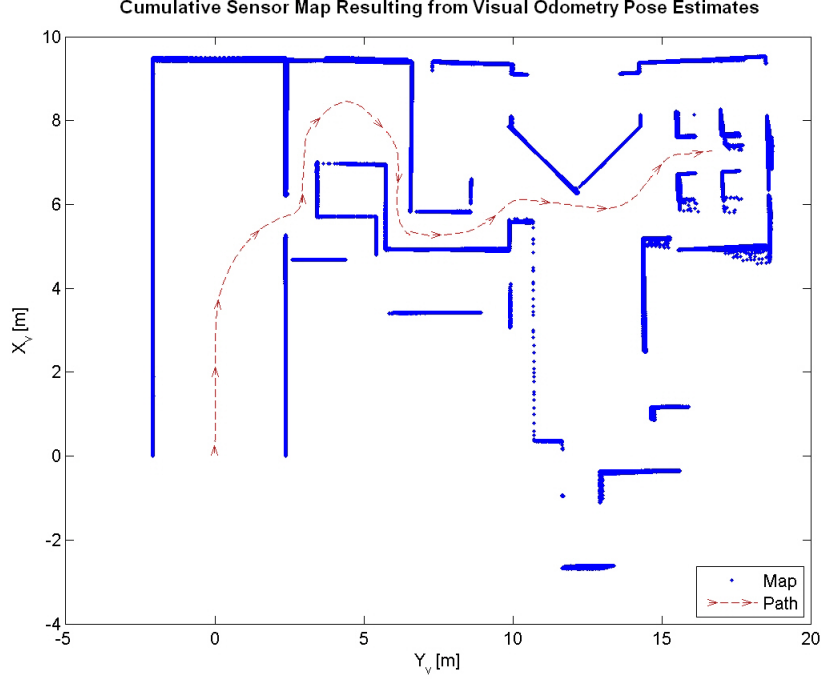


Fig. 2: Cumulative sensor map is based on the pose estimates produced by the VO navigation solution. This solution utilizes the ICP algorithm to compute the rotation and translation of the vehicle during a time steps and uses the pose estimate to translate the range scans into a coordinate frame relative to the vehicle's starting pose.

step 2 is obtained as follows [30]:

- Calculate $\mathbf{H} = \sum_{i=1}^{N_D} (\mathbf{p}_i - \mathbf{p}_c)(\mathbf{q}_i - \mathbf{q}_c)^T$; $(\mathbf{p}_c, \mathbf{q}_c)$ are the centroids of the point sets $(\mathbf{P}_M, \mathbf{P}_D)$.
- Find the Singular Value Decomposition (SVD) of \mathbf{H} such that $\mathbf{H} = \mathbf{U}\mathbf{\Omega}\mathbf{V}^T$ where \mathbf{U} and \mathbf{V} are unitary matrices whose columns are the singular vectors and $\mathbf{\Omega}$ is a diagonal matrix containing the singular values.
- The rotation matrix relating the two point sets is given by $\mathbf{R} = \mathbf{V}\mathbf{U}^T$.
- The translation between the two point sets is given by $\mathbf{T} = \mathbf{q}_c - \mathbf{R}\mathbf{p}_c$.

This process is iterated as stated in step 4 until the mean Euclidean distance between the corresponding point sets \mathbf{P}_M and \mathbf{P}_D is less than or equal to a predetermined distance or until a given number of iterations is exceeded.

IV. EXPERIMENTAL RESULTS

The navigation solutions discussed in this paper were evaluated using a simulated P2DX that was equipped with a simulated INS sensor, discussed in Section III-A, a simulated Odometry sensor, discussed in Section III-B, and a simulated 2D laser range finder. The laser range finder, configured to provide a field of view of 180 degrees with a beam separation of one degree, served as the basis for the Visual Odometry solution presented in Section III-C. The frequency of data returns for each sensor, limited by the computational complexities of the simulation environment, was approximately 5 Hz. Ground truth information, extracted through USARSim,

served as the basis of comparison to quantitatively evaluate each of the navigation solutions described in Section III.

The simulated P2DX was teleoperated using MOAST in an elemental mapping test world in USARSim that is being used in the 2007 Virtual RoboRescue Competition to test teams that autonomously map the environment. The *DM-Mapping_250* world (available on the USARSim home page) consists of several sections with varying degrees of complexity in terms of features and mobility characteristics. Data was logged using MOAST during a run that lasted over 4.5 minutes and traveled almost 26 meters at an average speed of 0.097 m/s. During the run, data streams were logged and processed and compared to ground truth information to quantifiably measure the performance characteristics of the two proprioceptive navigation solutions (i.e INS and Odometry) and the exteroceptive navigation solution (i.e. Visual Odometry) as depicted in Figure 1.

In this text, a *cumulative sensor map* refers to a composite map consisting of raw sensor data mapped into a relative coordinate frame using the pose estimate from the navigation solution (no filtering data or pruning of the map). Close examination of the cumulative map produced by the Visual Odometry solutions, shown in Figure 2, illustrates the integrity and robust nature of an exteroceptive approach to formulating a navigation solution with only marginal errors being produced in the top-right corner of the map. In this map there are slight discontinuities that may have led to these errors. Examining Figures 1(b), 1(c), and 1(d), the errors in ϕ_v for all three

solutions show a significant spike between 150 and 200 seconds. This spike, occurring three-quarters into the run, suggests the presence of a non-systematic error (i.e. hitting a wall) that may have led to the discontinuities observed in the cumulative sensor map for Visual Odometry.

A closer examination of the vehicle path errors in 1(a), shows that Visual Odometry exhibits a more accurate representation of the actual path traveled by the vehicle. The path produced by the INS solution appears to mimic that of Visual Odometry, but examining error plots of the INS, shown in Figure 1(b), reveals a biased drift that is causing divergence in the pose estimate. This drift is representative of systematic errors found in Inertial Navigation Systems. The presence of systematic errors does not seem to be as noticeable in the error plots for the encoder-based Odometry solution, see Figure 1(c). Comparing the error plots for the two proprioceptive approaches, we see that the encoder-based Odometry provides a better estimate of pose, where the INS solution provides a better estimation of orientation.

Using the error curves found in Figures 1(b) and 1(c) as the basis for inferring trends suggests errors accumulating in these solutions may have grown without bounds. In contrast, the error curves for the Visual Odometry navigation solution, seen in 1(d), suggests that this method provides a robust solution that is resilient to the systematic errors found in dead-reckoning sensors.

V. CONCLUSION AND FUTURE WORK

This paper explored the performance characteristics of several approaches towards developing a stable navigation solution, detailing the capabilities and limitations of each approach, as well as the underlying mechanisms used to formulate these solutions. Through quantitative analysis of each solution, this paper showed that the exteroceptive navigation solution is more resilient to systematic and non-systematic errors that plague proprioceptive-based solutions and provide the mechanism for limiting the uncertainty in the navigation solution. Although Visual Odometry is a computationally efficient navigation solution and exhibits good performance in complex environments, it is vulnerable to failure due to the lack of redundancy and its inability to measure the amount of uncertainty in the system at any given time. Therefore, the Visual Odometry solution must be fused with other approaches within a probabilistic framework to take into account errors in the vehicles pose and uncertainty being introduced into the system in order to overcome vulnerabilities that might compromise the autonomous navigation and mapping capabilities of the robotic system.

REFERENCES

- [1] *Rescue Systems I*, USFA/NFA-RS1-SM1993 ed., FEMA, 1993.
- [2] J. Leonard and H. Durrant-Whyte, "Mobile Localization by Tracking Geometric Beacons," in *IEEE Transactions on Robotics and Automation*, vol. 7, 1991, pp. 376–382.
- [3] "MOAST." [Online]. Available: www.sourceforge.net/projects/moast
- [4] R. Madhavan, *Unmanned Ground Vehicle Navigation in Unstructured Environments*, J. X. Liu, Ed. Nova Science Pub Inc, December 2005.
- [5] J. Borenstein, H. Everett, and I. Feng, *Navigating Mobile Robots: Systems and Techniques*. A.K. Peters Ltd., 1996.
- [6] A. Kleiner, J. Prediger, and B. Nebel, "RFID Technology-based Exploration and SLAM for Search And Rescue," in *Proceedings of the IEEE/RSJ International Conference on Robots and Systems*, October 2006, pp. 4054–4059.
- [7] L. J. Latecki and R. Lakaemper, "Polygonal Approximation of Laser Range Data Based on Perceptual Grouping and EM," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2006.
- [8] J. Gutmann, W. Burgard, D. Fox, and K. Konolige, "An Experimental Comparison of Localization Methods," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98)*, 1998.
- [9] T. Bailey, "Mobile Robot Localisation and Mapping in Extensive Outdoor Environments," Ph.D. dissertation, University of Sydney, 2001.
- [10] J. Nieto, T. Bailey, and E. Nebot, "Recursive scan-matching SLAM," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 39–49, 2007.
- [11] J. Gonzalez, A. T. Stentz, and A. Ollerio, "An Iconic Position Estimator for a 2D Laser RangeFinder," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '92)*, vol. 3, May 1992, pp. 2646 – 2651.
- [12] "USARSim." [Online]. Available: www.sourceforge.net/projects/usarsim
- [13] "Delta3D." [Online]. Available: www.delta3d.org
- [14] "Microsoft Robotics Studio." [Online]. Available: msdn.microsoft.com/robotics
- [15] "The Player Project." [Online]. Available: playerstage.sourceforge.net
- [16] M. Zaratti, M. Fratarcangeli, and L. Iocchi, "A 3D Simulator of Multiple Legged Robots based on USARSim," in *Robocup 2006: Robot Soccer World Cup X*, Springer, LNAI, 2006.
- [17] S. Carpin and M. Lewis and J. Wang and S. Balakirsky and C. Scrapper, "USARSim: a Robot Simulator for Research and Education," in *Proceedings of the IEEE 2007 International Conference on Robotics and Automation*, October 2007.
- [18] C. Scrapper, S. Balakirsky, and E. Messina, "MOAST and USARSim: A Combined Framework for the Development and Testing of Autonomous Systems," in *The International Society for Optical Engineering (SPIE '06)*, vol. 6230, 1988.
- [19] J. Albus and A. Meystel, *Engineering of Mind: An Introduction to the Science of Intelligent Systems*, J. Albus, A. Meystel, and L. Zadeh, Eds. John Wiley and Sons, Inc 2001, 2001.
- [20] S. Carpin and J. Wang and M. Lewis and A. Birk and A. Jacoff, "High Fidelity Tools for Rescue Robotics: Results and Perspectives," in *Robocup 2005: Robot Soccer World Cup X*, Springer, LNAI, vol. 4020, 2005.
- [21] J. Wang, M. Lewis, S. Hughes, M. Koes, and S. Carpin, "Validating USARSim for Use in HRI Research," in *Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting (HFES05)*, 2005.
- [22] S. Sukkarieh, "Low Cost, High Integrity, Aided Inertial Navigation Systems for Autonomous Land Vehicles," Ph.D. dissertation, University of Sydney, 2000.
- [23] I. Mandour and M. El-Dakiky, "Inertial Navigation System Synthesis Approach and Gravity-Induced Error Sensitivity," in *IEEE Transactions on Aerospace and Electronic Systems*, 1988.
- [24] "OXTS Interim+GPS." [Online]. Available: <http://www.oxts.co.uk/default.asp?pageRef=38&category=10&faq=30>
- [25] S. Thrun, D. Hähnel, D. Fergusson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker, "A System for Volumetric Robotic Mapping of Abandoned Mines," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
- [26] R. Madhavan, T. Hong, and E. Messina, "Temporal Range Registration for Unmanned Ground and Aerial Vehicles," in *2004 IEEE International Conference on Robotics and Automation*, 2004, pp. 3180 – 3187.
- [27] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A Review of Recent Range Image Registration Methods with Accuracy Evaluation," *Image Vision Comput.*, vol. 25, no. 5, pp. 578–596, 2007.
- [28] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.
- [29] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vision*, vol. 13, no. 2, pp. 119–152, 1994.
- [30] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 5, pp. 698–700, 1987.