PARTICLE FILTERING RANGE DATA FOR POSE ESTIMATION UNDER TORQUE-FREE MOTION

Stephen P. Russell, Stephen M. Rock
Aerospace Robotics Laboratory
Department of Aeronautics and Astronautics
Stanford University
Durand Building, Room 250
496 Lomita Mall
Stanford, California 94305
email: {sprussell,rock}@stanford.edu

ABSTRACT
Autonomous rendezvous and capture is a current research interest with varied mission applications ranging from servicing to sample return. Especially interesting is the problem of rendezvous with an uncooperative tumbling target in orbit. One key to this process is the ability to estimate the relative pose and motion of the target object. This paper presents a particle filter method which utilizes a continuous stream of range data to extract the relative pose and rotation rate of the target. This approach circumvents problems encountered by previous techniques, especially the issue of data smearing caused by target motion during data collection. This has implications for estimating faster rotations than previously demonstrated by existing solutions. Furthermore, the proposed method can utilize simpler sensor patterns compared to previous experiments and still correctly estimate the target state. The method is verified via simulation and results are presented.

KEY WORDS
Robot sensing and data fusion, pose estimation, particle filter, rendezvous and capture, simulation, aerospace

1 Introduction

This paper investigates a particle filter approach for continuously incorporating range data and extracting an estimate of target pose and motion. This filter framework incorporates each measurement as it is taken, resulting in more effective estimation of rapid rotational motion. Several algorithms have been proposed for solving this estimation problem but each has limitations as discussed in the subsequent sections of this introduction.

1.1 Background

There is a current desire for spacecraft to perform autonomous rendezvous and capture in orbit. This need arises in servicing missions for in situ satellite repair or in deorbiting operations for satellites at end of life. Current servicing is performed almost universally via manned missions, which are both costly and have safety of life concerns. This motivates the need for a service craft capable of docking with a target autonomously. Future demand for more satellites and lower operating costs drives the development of these new technologies.

Additionally, NASA has expressed an interest in eventual sample return missions from Mars or other extraterrestrial sites of interest. Autonomous rendezvous likely would be needed to interface the sample capsule with a return motor. The latency and bandwidth involved in performing teleoperation for such remote missions is prohibitive. Increased autonomous capability is vital for the realization of these future space endeavors.

Several missions have been conducted or are planned in the near future to attempt demonstrations of autonomous rendezvous capabilities. ETS-VII was launched by the National Space Development Agency of Japan in 1997 to conduct automated rendezvous and demonstrate robotic arm technology applicable to servicing. ETS-VII was the first unmanned spacecraft to successfully perform autonomous rendezvous in orbit [1]. The Demonstration of Autonomous Rendezvous Technology (DART) was an ambitious attempt to perform autonomous orbit matching and proximity maneuvers around an obsolete communications satellite that ended unsuccessfully and confirmed the challenge of autonomous docking [2].

More recently NASA and DARPA had success with Orbital Express, where autonomous docking and fuel/hardware transfer were performed [3]. Both DART and Orbital Express used the Advanced Video Guidance Sensor (AVGS), a laser image sensor designed to detect a pattern of retroreflectors on the target to determine its pose [4]. Currently, DARPA is working on FREND (to be followed by SUMO) for further demonstration of estimation algorithms and end effector technology capable of autonomously capturing generic satellites not equipped with sensing targets or dedicated grapple points [5].

1.2 Related Work

Many researchers have investigated the problem of estimating the pose of a target spacecraft, e.g. [6] [7] [8]. Though vision algorithms are very compelling, the vulnerability to illumination changes encourages the use of a sensor more suited to the harsh sensing environment of space. Batches of range measurements from LIDAR sensors have been used to solve for relative position but it is desirable to find a method better tailored to observing a moving target. Previous work has assumed relatively benign target motion and as the motion rate increases, range sensor information can become harder to interpret. This prompts a solution that can utilize sparse range data and effectively deal with the difficulties encountered when observing a moving object.

Sensing the relative pose of an object is a problem that has been tackled in many ways. A camera is an obvious sensor for studying a target object. These sensors are ubiquitous, cheap, and come with an extensive body of research and methods for their application. A great deal of work has been done in extracting 3D information from 2D images. The computer vision community has developed algorithms for reconstructing camera pose along with the relative environment, a field commonly referred to as structure from motion (SFM) [9] [10]. Recently, concurrent work in the Aerospace Robotics Laboratory at Stanford University has investigated using SFM methods along with fast-SLAM techniques in an architecture referred to as SPEAR (SLAM-inspired Pose Estimation And Reconstruction) [11].

Another related vision sensing technique, which exploits specific sensor patterns, is commonly called structured light. Rusinkiewicz et al. have shown the capability of tracking a freely moving object and reconstructing its shape in real-time at a 60 Hz rate [12]. Requiring this type of sensor, which depends on a specific infrastructure, is not always feasible though. In the context of the target pose estimation problem, visual sensing has the advantage of generating a large volume of simultaneous data. Measurements generated using SIFT features or structured light sensors can provide hundreds of points per range image. These visual techniques can be very powerful but rely on image data that are susceptible to changes in illumination, which is a concern in the space environment.

Another option for sensing relative information is to use a range sensor such as LIDAR. In many of the vision applications cited above, part of the algorithm involves solving for ranges to image features. Ranging sensors, on the other hand, simply provide a direct measurement of range. There are several current research efforts using range sensors to perform tasks applicable to pose estimation.

Object reconstruction, a field for which there exists a rich literature, is closely related to the goal of determining the pose of a target. Many of these methods solve for the registration of separate point clouds to merge them into a global target representation. The most well known example is the iterative closest point (ICP) method presented by Besl and McKay [13]. Upon solving this problem, the registration information can be used to compute where the sensor was as each point cloud was measured. Thus, one can determine a solution for the relative pose of the sensor with respect to the target. However, many of these cases assume the object is stationary. This allows the sensor to take a large volume of range data and produce a dense cloud of points, move to a different location and take another set of data, and so forth. These dense, rigidly constrained point clouds lend themselves well to ICP registration. The problem is complicated significantly if the target is moving while data are being taken.

Other recent work has attempted using range data in tracking a freely moving object. Blais and his colleagues used a scanning triangulation laser sensor to produce batches of range data. The sensor they used has a 2-axis steerable mirror such that it can generate a 2D scan pattern, e.g. a raster scan or a Lissajous curve. ICP was used to determine the sensor displacement between scans and iteratively refine the model estimate [14]. However, using ICP in this context has the added problem of solving for the warping of each point cloud; warping of the data is introduced because the object is not stationary during the scanning process. This is also commonly called data "smearing". The authors go on to do recursive warping estimation and again produce an accurate target model in the presence of oscillatory motion during the scanning process [15].

1.3 Warping Characterization

Though solutions for small motions have been demonstrated, the problem of data smearing is amplified with increased target motion. To characterize the severity of this problem, a simulation was conducted where a virtual 10 kHz scanning sensor was used to sweep out one Lissajous period containing 1024 beams. Scans were performed on two instances of a target (the same virtual Hubble model used in Section 4), one stationary and the other continuously rotating throughout the scan. The offsets between corresponding range measurements were averaged and plotted versus rotation rate. Figure 1 shows the results of 1000 Monte Carlo trials and indicates that motion of around 40° per second can induce average beam offsets of greater than 10cm. These offsets are 10-20 times the documented statistical error of commercial laser range sensors [16]. Thus, if considered a source of noise, this warping significantly increases the uncertainty of how accurately a batch scan represents the target geometry.

1.4 Problem Specification

The problem of interest is estimating target orientation and rotational motion using a scanning range sensor. The state

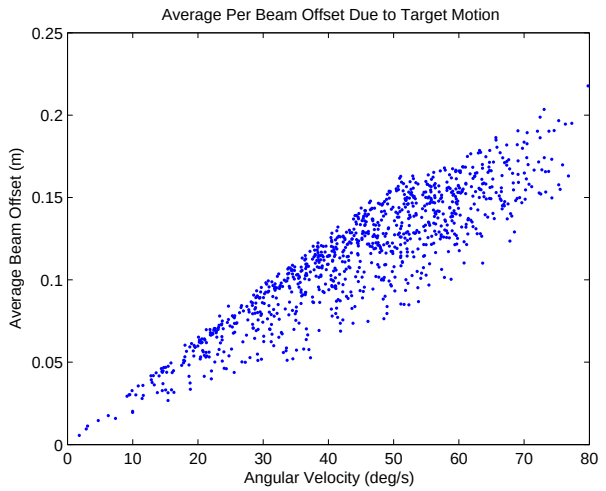

Figure 1. Average beam offsets due to scan warping (1000 Monte Carlo trials)

to be estimated is

$$[\phi, \theta, \psi, \omega_x, \omega_y, \omega_z]^T$$

where the first three states are target Euler angles and the last three are components of angular velocity represented in the inertial frame.

The target is assumed to be a rigid body for which a CAD model and associated inertias are available. For many objects of interest, this information is readily obtainable. Also, this work assumes the target is undergoing torque-free tumbling motion and is not translating with respect to the observer. If the target's orbit is known with high confidence, approaching along a V-bar trajectory and holding station in distance to the target should be a realizable objective. Finally, knowledge of the observer's state (inertial position, orientation, and rates) is assumed to the precision of standard instrumentation.

The target is not equipped with any reflective targets or fiducials. Sensing information is gathered simply as range returns from the intrinsic geometry of the spacecraft. Also, the target is considered uncooperative. This does not indicate the target is actively trying to avoid the observer but rather that the target is not transmitting any helpful state information.

The previous work by Blais et al. accounted for motion during data collection by solving a batch optimization over an entire scan. The solution yielded additional corrections to each individual measurement. The cost function minimized was distance between the point and the reconstructed model. In their case, an a priori model was not assumed. In fact, the focus of that work was primarily target reconstruction as opposed to accurate motion estimation.

In contrast, having an estimate of target motion becomes paramount if the goal is relative position control. This is accomplished here by explicitly including states $(\omega_x, \omega_y, \omega_z)$ to capture target motion. With an estimate of motion and assuming a target model, a hypothesized range measurement can be generated for comparison with an actual sensor return at each time step. This enables recursive state estimation as opposed to storing measurements for batch ICP registration. This recursion, which eliminates batch data storage, intrinsically solves the data smearing problem by propagating the motion estimate between each individual beam scan. The implementation of this technique is discussed in detail throughout sections 2 and 3.

The remainder of this paper will present the following: Section 2 introduces the particle filter and demonstrates why this technique lends itself well to this problem. Section 3 presents the algorithm for incorporating online range measurements into a filtering framework to produce an estimate of relative pose and target motion. Simulation results are given in Section 4 before some concluding remarks in Section 5.

2 Estimator Selection

A particle filter is a probabilistic estimation technique that has distinct advantages over other methods for estimating the motion of a tumbling target. However, use of a particle filter comes at the price of computational complexity. A particle filter is just one variant of sequential Bayesian estimators; others include the Kalman filter and discrete, grid-based estimators. This section discusses the advantages that make using a particle filter worth the increased computational price.

A Bayes filter is a technique for estimating a state x by recursively generating probability distributions over the state space conditioned on previous measurements and inputs. The particle filter is a specific variant that represents the probability density using a discrete set of weighted samples distributed according to the belief state. Realization of recursive Bayesian filtering is enacted through a process of resampling the particles based on their importance weights. Many detailed references exist on the specifics of particle filters, e.g. [17] [18] [19] .

There are several other variants of the discrete Bayes filter as well. Many utilize grid-based or topological based approaches. This can be advantageous for data lookup in correlating versus known maps and other application specific reasons. However, these methods generally require propagating more states than a particle filter. For this application, a particle filter seems best suited due to its nature of focusing attention on high probability regions of the state space.

The primary distinction of this filtering approach is incorporating each single measurement as it is taken as opposed to storing batches of measurements as point clouds and then registering these batches to each other. This effectively eliminates the warping problem with the ICP technique noted in the previous section. This is in addition to several other advantages of using a particle filter for this particular scenario.

A particle filter can be initialized with a very poor idea of the actual state and still converge to the truth, so long as it is provided with particles initialized throughout the state space. The ability to populate the state space thoroughly and propagate these particles is limited only by computational complexity. In contrast, a Kalman filter can be sensitive to accuracy of the initial conditions. This issue of initialization has practical implications as well. If a target satellite is spinning quickly, it would be difficult if not impossible to initialize the state estimate well. Allowing the filter to initialize the estimate over a wide range of possibilities is more general and is especially beneficial from the perspective of enhanced autonomy.

A particle filter can easily model a multi-modal belief state. This can be very advantageous for pose estimation, especially for a target with geometric symmetry. If there are strong likelihoods at different viewpoints from which the observer would see similarly shaped parts of a spacecraft, it is desirable to have the filter model the appropriate distribution. This is as opposed to a Kalman filter, which assumes a unimodal posterior and is better suited to tracking a well localized estimate through the state space.

Another benefit of this particular approach is the straight forward measurement update. Each particle has a believed sensor position and can generate a hypothesized range return. The error measurement at each time step is generated by comparing this hypothesis with the actual sensor return. The only challenge is constructing a function that assimilates these errors and distinguishes good particle estimates from bad. The specific measurement update for this work can be found in Section 3 below. In contrast, since a single range measurement is not a direct measurement of the target's pose nor is it a measurement of motion rate (unless Doppler sensors are considered), which are the states of interest, the motion update for a Kalman filter would be complicated and highly non-linear.

One final advantage of this filtering approach is the ability to employ different, simpler sensor patterns. ICP needs spatial variation of points for the registration to converge well. This is what necessitated the sophisticated 2D Lissajous scanner used by Blais and his colleagues. Using a particle filter though, one could hypothetically use a fixed direction single range scan. As long as particles have motion estimates that generate successive ranges closely matching the sensor returns, they will distinguish themselves over particles having large discrepancies relative to the sensor measurements. Empirically, it does appear that sweeping the scan direction and providing some variation is beneficial for convergence. However, sensor patterns that are simpler than the Lissajous and would likely cause ICP to fail (e.g. a crosshair pattern or a single line scan) have been tested and shown to converge when used with a particle filter. The ability to employ a simpler sensor could have dramatic practical implications, especially for realization on smaller spacecraft such as a CubeSat.

3 Algorithm

The following is an algorithm that utilizes range data in a particle filter framework to continuously incorporate measurements and extract an estimate of the relative pose and motion of a target.

3.1 Initialization

To initialize the filter, the Euler angles are spread to represent the space of rotations uniformly. This is done using the method presented by Arvo in [20]. A very nice property of the Euler angle initialization is the fixed scale of this portion of the state space. Since each Euler angle is bounded and cyclic, it is necessary only to cover a fixed interval (i.e. $[-\pi, \pi)$ for roll and yaw, $[-\pi/2, \pi/2]$ for pitch) to achieve coverage in these three dimensions. It is only necessary to increase the number of particles in the angular velocity dimensions if more of the state space must be explored.

The angular velocity states are initialized uniformly over a user defined interval, distributed as $U[-\omega_{max}, \omega_{max}]$. The particle filter should be able to capture very fast motion as long as this interval is wide enough and particles are initialized with sufficient density. Note that the smearing problem in the ICP method is exacerbated by faster target motion. Thus, this approach is more scalable provided sufficient computation is available.

3.2 Motion Update

The motion update propagates the Euler angles of the target and the angular velocity vector. The motion model leverages the torque-free motion assumption. These computations employ a linearized model for speed but, since the time scale is so short, this is a good approximation. Each update is the time between successive laser scans which, for many commercial sensors, is commonly less than 0.001 seconds (1 kHz scan rate).

Euler angle propagation:

Convert the Euler angles to a rotation matrix [21]

$$R_b^I(t-1) = f(\phi(t-1), \theta(t-1), \psi(t-1)) \quad (1)$$

Solve for the angular velocity components in the body frame from the previous time step

$$\omega_b(t-1) = R_b^I(t-1) w_I(t-1) \quad (2)$$

Note that subscript b and I refer to the body frame and inertial frame, respectively. R_b^I is a rotation matrix that takes a vector in inertial coordinates and represents it in the body frame.

Assuming constant angular velocity for time Δt, a rotation matrix can be propagated using the matrix exponential [22]

$$R_b^I(t) = e^{\hat{\omega}\Delta t} R_b^I(t-1)$$

where $\hat{\omega}$ is a skew-symmetric matrix containing components of ω_b from equation (2).

$$\hat{\omega} = \begin{bmatrix} 0 & \omega_b(3) & -\omega_b(2) \\ -\omega_b(3) & 0 & \omega_b(1) \\ \omega_b(2) & -\omega_b(1) & 0 \end{bmatrix}$$

For the actual propagation, the matrix exponential is linearized and only terms up through the quadratic are used

$$e^{\hat{\omega}\Delta t} \approx I + \hat{\omega}\Delta t + \frac{(\hat{\omega}\Delta t)^2}{2!}$$

Substituting,

$$R_b^I(t) = \left(I + \hat{\omega}\Delta t + \frac{(\hat{\omega}\Delta t)^2}{2!} \right) R_b^I(t-1) \quad (3)$$

The Euler angles at time t can be extracted from this updated rotation matrix.

Angular velocity propagation:

Once the target position at time t is determined, it is possible to propagate the angular velocity vector. This is done using the knowledge that under torque-free motion, the angular momentum vector is fixed in inertial space. The particles' state information can be used to form estimates of the angular momentum vector at time $t-1$. Then, using the updated Euler angles and the known inertia matrix, it is possible to compute what ω should be at time t.

From the definition of angular momentum, solve for angular velocity in the inertial frame

$$\begin{aligned} H_I &= I_I \omega_I \\ I_I^{-1} H_I &= \omega_I \end{aligned} \quad (4)$$

Knowledge of the target inertia matrix is assumed but is likely represented in a body fixed frame. This must be transformed to the inertial frame to fit into the equation above, which is straight forward with a rotation matrix representing the orientation of the body.

$$I_I = (R_b^I)^T I_b R_b^I$$

Since the target is undergoing torque-free motion, its angular momentum H is constant in the inertial frame. The particles' estimates of pose and angular velocity are used to compute estimates of H

$$H_I(t) = H_I(t-1) = R_b^I(t-1)^T I_b R_b^I(t-1)\omega_I(t-1) \quad (5)$$

Substituting into equation (4) and solving for ω_I at time t,

$$R_b^I(t)^T I_b^{-1} R_b^I(t) \left[R_b^I(t-1)^T I_b R_b^I(t-1)\omega_I(t-1) \right] = \omega_I(t) \quad (6)$$

3.3 Measurement Update

As stated, incorporating new measurements is simply a matter of computing error between the hypothesized particle measurement and the actual sensor return. These errors are then squared, accumulated, and averaged since the last resampling. The square root of this average is taken to yield the root mean squared (rms) error per beam. The rms beam error for each particle is then translated to a weight. The weights are normalized to more closely model a discrete PDF and for computational stability.

The measurement error for particle i at time t is

$$\epsilon_i(t) = (\hat{z}_i(t) - z(t))^2$$

Accumulating and computing rms error per beam,

$$E_i = \sqrt{\frac{1}{n} \sum_t \epsilon_i(t)}$$

Finally, weights are computed as a Gaussian likelihood of measuring the rms error seen for particle i

$$\overline{W_i} = e^{-E_i^2/(2\sigma^2)}$$

and then normalized

$$W_i = \frac{1}{\sum_i \overline{W_i}} \overline{W_i}$$

Note that the variance σ is a parameter that can be adjusted to tune the filter. This shapes the weighting function and determines how much error a particle can accumulate before being in danger of elimination. This will vary depending on the sensor noise, the shape variation of the target, etc.

Resampling occurs after a specified minimum time interval using a low-variance sampling algorithm [19]. Allowing measurements to accumulate allows the angular velocity states of the particles to influence the errors/weights. That is, a particle needs to continue getting plausible measurements over a period of time for the filter to believe it is truly a good estimate.

4 Simulation Results

To test this algorithm, a simulation was developed in MATLAB. The target was a simplified virtual model based on the Hubble space telescope geometry [23]. It consisted of 268 triangular faces and is shown in Figure 2. This target is challenging due to strong symmetry when seen from either the front or back, the only difference being the 30° cant of the solar panels.

Measurements were generated using a virtual range sensor with a maximum range of 80 meters and a statistical beam error of 0.005 meters. These parameters were based on the SICK LMS-200 [16]. The sample rate and scanning

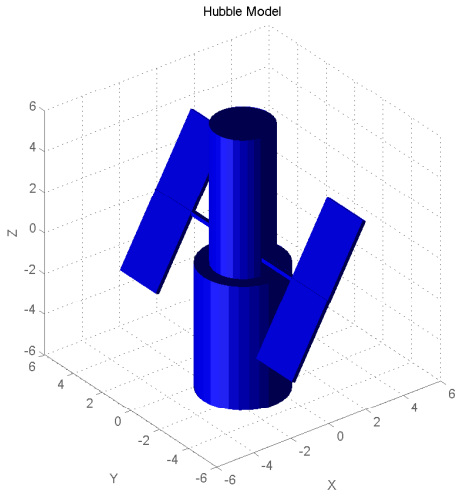

Figure 2. Virtual Hubble Model

pattern were varied to explore convergence properties for different sensors. The particle filter was generally populated with 10000 particles, though convergence has been shown for smaller particle set sizes. Initialization of the particles was uniform over the space of rotations for the Euler angles, as discussed in Section 3, and uniform over the interval $-\pi/2$ to $\pi/2$ radians per second (90° degrees per second) in each of the angular velocity components. These parameters are summarized in Table 1.

Figures 3 and 4 are plots of the particle set converging during a simulated scan. The results show the weighted mean (+) approaching and ultimately tracking the true state trajectory ($\circ$). These are results for just one example trial. Several Monte Carlo simulations were conducted with randomly generated angular momenta to verify repeatability. For this particular run, the Euler angles converge with approximately 0.02 radians (1.1°) of error and the angular velocity converges to within about 0.04 radians per second (2.3° per second) of the truth. These results were commensurate with other trials conducted.

Note that in a small percentage of trials the algorithm converged to the symmetric position mentioned in the description of the Hubble model, i.e. 180° away in yaw. Commonly in this case, the angular velocity is still correct indicating the algorithm has solved for the true angular momentum. However, the solution has converged to an incorrect estimate of the Euler angle initial conditions, thinking the observer started on the other side of the target. To add robustness, one could adapt the algorithm specifically for a given scenario. In this case, the algorithm could resample some particles into the symmetric region of the state space to see if those particles distinguish themselves as more likely solutions. This is akin to the technique of randomly injecting particles into the state space to encourage particle diversity.

Table 1. Simulation Parameters

Parameter		Value	Units
Sensor	Maximum range	80	m
	1σ beam error	0.005	m
Initialization	Number of particles	10000	
	(ϕ, θ, ψ)	See [20]	rad
	$(\omega_x, \omega_y, \omega_z)$	$U[-\frac{\pi}{2}, \frac{\pi}{2}]$	rad/s

5 Conclusion

This work has demonstrated a method for relative pose and motion estimation using range data from a scanning sensor. Measurements are incorporated continuously, which naturally handles the data smearing problem encountered when compiling batches of range returns for use in ICP registration. The advantages of a particle filter make it well suited for solving this problem, especially the ability to widely initialize the filter without a good estimate and still localize correctly in the state space. Thus, this method does not require solving for data warping, which worsens with rapid tumbling, nor does it require tight initialization of the filter, which is difficult for a quickly rotating target. This indicates the proposed method is better equipped to solve previously prohibitive estimation problems for rapid tumbling. In fact, the results presented show convergence for a target spinning at approximately 0.85 radians per second (approximately 50° per second). Through simulation, this algorithm has been shown to converge to within ± 0.05 radians or radians per second of the true target states. Finally, this method does not demand the spatial dispersion of data required by ICP, thus allowing the use of simpler sensor patterns.

One limitation of this algorithm is the computational complexity involved. The algorithm is currently unable to execute in real-time. Though, real-time capability is not necessary if the ultimate objective is only to estimate the target pose and motion; estimation could be done offline in this case. Real-time would be required if the algorithm were to be used as part of a closed-loop controller. Note that [24] offers encouraging results for parallel computing applied to real-time particle filter tracking.

This algorithm has been proven in simulation, but it would be beneficial to verify the performance via hardware trials. Future research on this topic will include laboratory demonstrations where the simulation scenario presented can be replicated using actual sensor data. Additionally, it would be advantageous to further streamline the estimator and reduce the required runtime. This could include the aforementioned parallel computing techniques and adapting the particle set size based on the convergence of the estimate. Finally, this technique has potentially broader application in fields outside of orbital rendezvous and it would be useful to expand its capability. This would primarily entail generalizing the motion model and relaxing the assumption of torque-free motion.

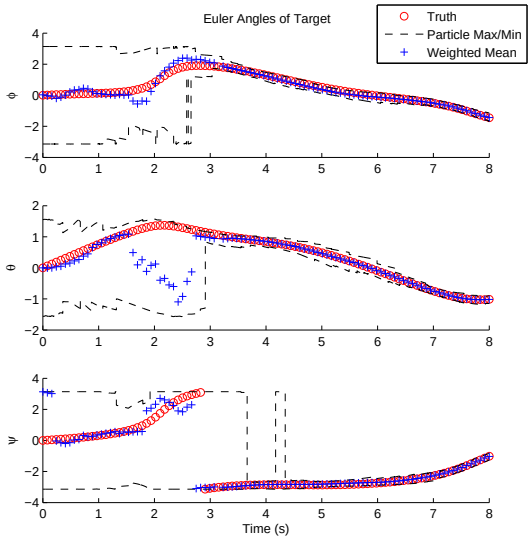

Figure 3. Particle filter convergence for Euler Angles

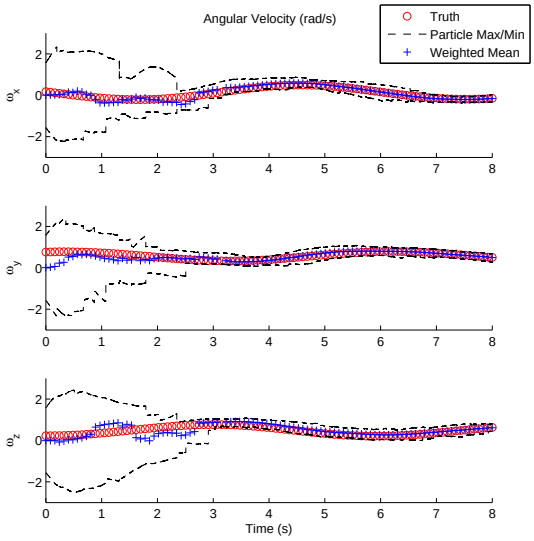

Figure 4. Particle filter convergence for angular velocity components

Acknowledgments

This research is funded by NASA through the Institute for Dexterous Space Robotics (IDSR) under subcontract #Z641501 to the University of Maryland at College Park.

References

[1] T. Kasai, M. Oda, and T. Suzuki. Results of the ETS-7 Mission-Rendezvous docking and space robotics experiments. *Artificial Intelligence, Robotics and Automation in Space*, 5:299–306, 1999.

[2] NASA. Overview of the DART Mishap Investigation Results. Public Release Version, May 15 2006.

[3] D.A. Whelan, EA Adler, S.B. Wilson III, and G.M. Roesler Jr. DARPA Orbital Express program: Effecting a revolution in space-based systems. In *Proceedings of SPIE*, volume 4136, page 48, 2000.

[4] S. Van Winkle. Advanced Video Guidance Sensor (AVGS) project summary. In *Proceedings of SPIE*, volume 5418, page 10, 2004.

[5] J. Obermark, G. Creamer, B.E. Kelm, W. Wagner, and C.G. Henshaw. SUMO/FREND: vision system for autonomous satellite grapple. In *Proceedings of SPIE*, volume 6555, page 65550Y, 2007.

[6] P. Jasiobedzki, M. Greenspan, and G. Roth. Pose determination and tracking for autonomous satellite capture. In *Proceedings of the sixth International Symposium on Artificial Intelligence and Robotics & Automation in Space (i-SAIRAS 01), Montreal, Canada*, 2001.

[7] MD Lichter and S. Dubowsky. State, shape, and parameter estimation of space objects from range images. In *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04*, volume 3, 2004.

[8] S. Ruel, C. English, M. Anctil, J. Daly, C. Smith, and S. Zhu. Real-time 3D vision solution for on-orbit autonomous rendezvous and docking. In *Proceedings of SPIE*, volume 6220, page 622009, 2006.

[9] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge Univ Pr, 2003.

[10] M. Pollefeys, M. Vergauwen, K. Cornelis, J. Tops, F. Verbiest, and L. Van Gool. Structure and motion from image sequences. In *Proc. Conf. on Optical 3-D Measurement Techniques*, pages 251–258, 2001.

[11] Sean Augenstein and Stephen Rock. Simultaneous Estimation of Target Pose and 3-D Shape using the FastSLAM Algorithm. In *Proceedings of AIAA GNC 2009*, 2009.

[12] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-time 3d model acquisition. *ACM Trans. Graph.*, 21(3):438–446, 2002.

[13] PJ Besl and HD McKay. A method for registration of 3-D shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992.

[14] F. Blais, M. Picard, and G. Godin. Recursive model optimization using ICP and free moving 3D data acquisition. In *Proc. 3DIM*, pages 251–258, 2003.

[15] F. Blais, M. Picard, and G. Godin. Accurate 3D acquisition of freely moving objects. In *Second International Symposium on 3D Data Processing, Visualization and Transmission.*, pages 6–9, Thessaloniki, Greece, 2004.

[16] SICK. LMS200-30106 Online Data Sheet. https://www.mysick.com/PDF/Create.aspx?ProductID=33755&Culture=en-US.

[17] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[18] A. Doucet and N. De Freitas. *Sequential Monte Carlo methods in practice*. Springer, 2001.

[19] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[20] J. Arvo. *Fast random rotation matrices*, pages 117–120. Academic Press Professional, Inc. San Diego, CA, USA, 1992.

[21] A.E. Bryson. *Control of spacecraft and aircraft*. Princeton university press, 1994.

[22] Y. Ma, S. Soatto, J. Kosecka, Y. Ma, S. Soatta, J. Kosecka, and S. Sastry. *An invitation to 3-D vision*. Springer, 2004.

[23] Space Telescope Science Institute (SCScI). Hand-held hubble: The real telescope compared. http://hubblesite.org/the_telescope/hand-held_hubble/the_real_thing.php.

[24] A.S. Montemayor, J.J. Pantrigo, Á. Sánchez, and F. Fernández. Particle filter on GPUs for real-time tracking. In *International Conference on Computer Graphics and Interactive Techniques*. ACM New York, NY, USA, 2004.