INTERACTIONS OF POSE ESTIMATION AND ONLINE DYNAMIC MODELING FOR A SMALL INSPECTOR SPACECRAFT

Mishari Alarfaj[1] **and Forrest Rogers-Marcovitz**[2]

[1] *ECE Dept., Carnegie Mellon University, Pittsburgh, PA, USA, malarfaj@andrew.cmu.edu*
[2] *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, forrest@cmu.edu*

Abstract

Close-proximity operations between spacecraft allow for docking and inspection of a target vehicle. Very small spacecraft are well-suited for proximity activities and are of growing interest in the aerospace community. However, due to size and power constraints, small vehicles cannot carry traditional precision navigation systems and generally have noisy sensor and actuator options. This paper presents two techniques for improved autonomous, on-board navigation that account for these poorly observable states. First, an Unscented Kalman Filter is implemented for pose estimation which incorporates orbital dynamics and quaternion rotation. Second, online Bayes Linear Regression (BLR) models the time-varying thruster dynamics. The BLR is then used to complement an unreliable sensor in the UKF, improving pose prediction. These techniques have been demonstrated successfully on a simulated small inspector vehicle and are being integrated in the Bandit inspector spacecraft.

1. INTRODUCTION

Close-proximity operations between spacecraft allow for docking and inspection of a target vehicle. Extended applications include on-orbit assembly and transfer of fuel, electricity, or other resources. Very small spacecraft, under 20 kg, are well suited for these types of proximity activities, defined here to be less than 100 meters in separation. They are easy to maneuver, inexpensive to build and operate, and have a minimal mass cost for launch. However, they are constrained by the amount of electircal energy they can store, and they cannot carry traditional, large-footprint, precision navigation systems. These constraints pose a significant navigation challenge.

The Bandit mission demonstrates remote operation of a 3-kg service vehicle, including repeatable docking, station-keeping and blended autonomous control [2][15]. Bandit addresses the constraints of small spacecraft operation by decoupling orbital functions between the 3 kg Bandit inspector vehicle and a larger (30 kg) dedicated host vehicle, Figure 1. The host vehicle, Akoya, will be responsible for all ground communication and power generation, enabling Bandit to be stripped to its essentials: imaging, short-term power, short-range communications, and navigation. This functional decoupling, along with other rapid integration protocols, meets the needs of the responsive space community [11].

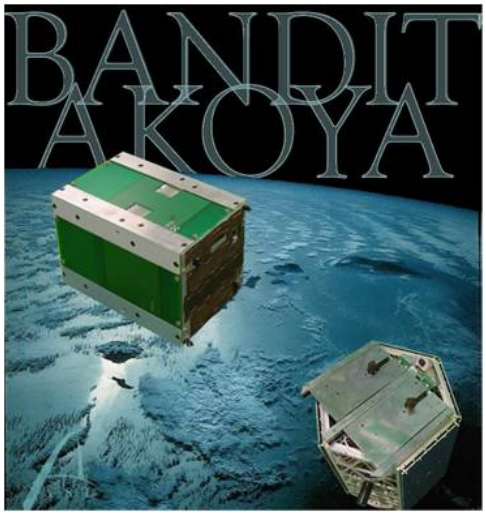

Figure 1: Bandit and Akoya spacecraft.

Bandit uses eight cold gas thrusters located around its exterior for actuation and uses low-cost acceleration and roll-rate sensors, along with LED based imaged tracking, for localization. The dynamics of Bandit change over time based on thruster mis-alignment, propellant temperature, propulsion performance, and changes in inertia moments as propellant is used.

We present two techniques for improved autonomous, on-board navigation of the Bandit vehicle. First we have implemented an Unscented Kalman Filter for pose estimation which incorporates orbital dynamics and quaternion rotation. Second, the thruster dynamics were learned via online Bayes Linear Regression. The online dynamic modeling takes the active thruster commands as inputs and uses them to predict the change in linear and angular velocities.

This paper looks at the integration between these two techniques, in which the output from the Bayes Linear Regression is used to complement the Unscented Kalman Filter, significantly increasing performance by reducing the dependence on the unreliable accelerometer. Results from our full orbital simulation

for each method are provided, showing good performance despite external disturbance and un-modeled variances in the actuators. In addition, improvements were seen in both docking time and reducing valve firings. These techniques are currently being integrated into the Bandit inspector spacecraft.

2. POSE ESTIMATION

Bandit navigates through a combination of inertial and vision sensors. MEMS accelerometers and roll-rate gyros provide high-rate information about Bandit's motion, although sensitivity limitations in the accelerometers may render Bandit's small translational accelerations indistinguishable from sensor noise. An on-board vision system converts images of the host to relative position and attitude estimation, which is used to periodically correct the accumulated drift from the inertial sensors. The exterior of the host vehicle is instrumented with color-coded LEDs in order to aid the image-based navigation solution.

Estimating the state of a dynamic system is a fundamental problem in spacecraft control. We will first explain the details of Bandit's motion model, followed by the Unscented Kalman Filter (UKF) used, which was modified in order to handle quaternion rotation. We will not cover the full details of the UKF as it has been described in other sources [7] [17].

2.1. Relative Orbital Dynamics

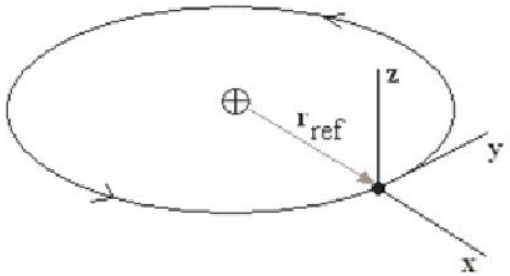

Figure 2: Clohessy-Wiltshire Relative Coordinate Frame.

Bandit's translational dynamics are written in a Clohessy-Wiltshire (CW) coordinate frame [3] which captures the relative orbital dynamics, shown in Figure 2. In the figure, the central sphere represents Earth, while the center of mass of the host satellite, Akoya, is designated as the origin. The x-axis is anti-nadir pointing, away from Earth. The y-axis is the velocity vector tangent to the orbit, and the z-axis is the cross product of the x and y axes. The relative orbital translation dynamics of Bandit, when in close proximity to Akoya, can be expressed in the closed form solution,

$$\vec{x}^p = \begin{bmatrix} \vec{x} \\ \dot{\vec{x}} \end{bmatrix}, \ \vec{x}^p_t = \Phi(t)\vec{x}^p_0 \tag{1}$$

where $\vec{x}$ and $\dot{\vec{x}}$ represents 3-dimensional position and velocity respectively. $\vec{x}^p_t$ denotes the augmented state at time t, with $t = 0$ denoting the initial state, and the linear orbit propagation matrix $\Phi(t)$ is defined as:

$$\Phi(t) = \tag{2}$$

$$\begin{bmatrix} 4-3C & 0 & 0 & \frac{S}{\Omega_r} & \frac{2}{\Omega_r}(1-C) & 0 \\ 6(S-\Omega_r t) & 1 & 0 & \frac{-2}{\Omega_r}(1-C) & \frac{4}{\Omega_r}S-3t & 0 \\ 0 & 0 & C & 0 & 0 & \frac{S}{\Omega_r} \\ 3\Omega_r S & 0 & 0 & C & 3S & 0 \\ -6\Omega_r(1-C) & 0 & 0 & -2S & 4S-3 & 0 \\ 0 & 0 & -\Omega_r S & 0 & 0 & C \end{bmatrix}$$

$$C = \cos(\Omega t), \ \ S = \sin(\Omega t), \ \ \Omega_r = \sqrt{\frac{G_{earth}}{R^3_{earth}}}$$

where Ω_r is the angular velocity of the reference orbit, G_{earth} is Earth's gravitational constant, and R_{earth} is the orbit's altitude (distance from Earth's center).

2.2. State Propagation

The state vector X and control vector u are defined as such,

$$\vec{X} = \begin{bmatrix} \vec{x} \\ \dot{\vec{x}} \\ \vec{q} \end{bmatrix} \quad \vec{u} = \begin{bmatrix} \vec{\alpha} \\ \vec{\omega} \end{bmatrix}$$

The state vector is in the local Clohessy-Wiltshire coordinate frame and includes position, $\vec{x}$, linear velocity, $\dot{\vec{x}}$, and quaternion rotation, $\vec{q}$. The control vector is in Bandit's body frame and includes the accelerometer, $\vec{\alpha}$, and roll-rate gyroscope, $\vec{\omega}$, measurements. When no thrusters are firing, linear acceleration is assumed to be zero as all outside forces are negligible until contact with the dock. Bandit's rotation quaternion is updated via the instantaneous Euler rotation rates reported by the gyroscope,

$$\vec{q}_{t+1} = e^{(\frac{1}{2}\Omega\delta_t)}\vec{q}_t \tag{3}$$

where δ_t is the time step, and

$$\Omega = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \tag{4}$$

Euler integration is performed on the acceleration measurements after the CW orbital propagation in order to estimate translational velocities,

$$\dot{\vec{x}}^+_{t+1} = \dot{\vec{x}}^-_{t+1} + \delta_t \mathbf{C}(\vec{q}_{t+1}) \begin{bmatrix} \alpha_{x(t+1)} \\ \alpha_{y(t+1)} \\ \alpha_{z(t+1)} \end{bmatrix} \tag{5}$$

Where $\mathbf{C}(\cdot)$ is the corresponding rotation matrix for the given quaternion and the superscripts $+$ and $-$ denote pre- and post-estimation states.

2.3. Unscented Kalman Filter

The Unscented Kalman Filter (UKF) estimates the state of a dynamic system based on a sequence of observations and control information. Let $\vec{x}_t$, $\vec{u}_t$, and $\vec{z}_t$, be the discrete-time states of the system, control inputs, and observations at time t, respectively. We assume that the system evolves according to the state transition function g,

$$\vec{x}_t = g(\vec{u}_{t-1}, \vec{x}_{t-1}) + \vec{\epsilon}_t \tag{6}$$

where $\vec{\epsilon}_t$ is additive zero-mean Gaussian noise with covariance Q_t, representing process noise. Similarly the observation $\vec{z}_t$ is a function of the current state corrupted by observation noise $\vec{\gamma}_t$, which is assumed to be Gaussian with covariance R_t,

$$\vec{z}_t = h(\vec{x}_t) + \vec{\gamma}_t \tag{7}$$

We assume the functions g and h are non-linear. In order to estimate posteriors over the state space using efficient Kalman filtering, we have to linearize the functions g and h. Extended Kalman filters (EKF) perform this linearization using Taylor series expansion, with explicit Jacobian matrices, around the most recent estimate. UKFs apply a more accurate, stochastic approximation called the unscented transform [4]. The unscented transform performs this by extracting key 'sigma points' from the Gaussian estimate and passing them through the given non-linear function. The complete UKF algorithm is summarized in Table 1. In this implementation, the unscented transform is performed on an augmented state vector and covariance matrix which integrates the control inputs and observations. This transform utilizes the Cholesky decomposition, which is denoted in the algorithm by a square-root operator.

The update step occurs whenever a LED patten is observed on Akoya via Bandit's camera. Given the square pattern, with a unique identifier, we can calculate the relative position and orientation of the two spacecraft [1].

The UKF algorithm was modified to preserve the nonlinear nature of the unit quaternion [5] [8]. While determining the sigma-points, the quaternion sections of each sigma points were calculated by constructing an error quaternion,

$$\vec{\delta q} = \begin{bmatrix} \frac{\sin(|\vec{\phi}|)/2}{|\vec{\phi}|}\vec{\phi} \\ \cos(|\vec{\phi}|/2) \end{bmatrix} \tag{8}$$

The rotational error vector $\vec{\phi}$ represents the x, y, and z rotational differences from the current state quaternion and is used in the Cholesky decomposition during sigma-point finding. The error quaternion is then combined with the the pre-estimation quaternion via quaternion multiplication prior to being passed into the transition function :

$$X^{x,q}_{t-1} = \vec{\delta q}_{t-1} * \vec{\mu}^{q}_{t-1} \tag{9}$$

Algorithm *Unscented Kalman Filter*
Input: $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$
1.
(∗ Generate augmented mean and covariance ∗)
$\mu^a_{t-1} = \begin{pmatrix} \mu^\top_{t-1} & 0^\top_{m\times 1} & 0^\top_{p\times 1} \end{pmatrix}^\top$
2. $\Sigma^a_{t-1} = \begin{pmatrix} \Sigma_{t-1} & 0 & 0 \\ 0 & Q_t & 0 \\ 0 & 0 & R_t \end{pmatrix}$
(∗ Generate sigma points ∗)
3. $\chi^a_{t-1} = (\mu^a_{t-1},\ \mu^a_{t-1} + \gamma\sqrt{\Sigma^a_{t-1}},\ \mu^a_{t-1} - \gamma\sqrt{\Sigma^a_{t-1}})$
(∗ Pass sigma points through motion model ∗)
(∗ and compute Gaussian statistics ∗)
4. $\bar{\chi}^x_t = g(u_t + \chi^u_{t-1}, \chi^x_{t-1})$
5. $\bar{\mu}_t = \sum_{i=0}^{2L} w_i^{(m)} \bar{\chi}^x_{i,t}$
6. $\bar{\Sigma}_t = \sum_{i=0}^{2L} w_i^{(c)} (\bar{\chi}^x_{i,t} - \bar{\mu}_t)(\bar{\chi}^x_{i,t} - \bar{\mu}_t)^\top$
(∗ Predict observations at sigma points ∗)
(∗ and compute Gaussian statistics ∗)
7. $\bar{Z}_t = h(\bar{\chi}^x_t) + \chi^z_{t-1}$
8. $\hat{z}_t = \sum_{i=0}^{2L} w_i^{(m)} \bar{Z}_{i,t}$
9. $S_t = \sum_{i=0}^{2L} w_i^{(c)} (\bar{Z}_{i,t} - \hat{z}_t)(\bar{Z}_{i,t} - \hat{z}_t)^\top$
10. $\Sigma^{x,z}_t = \sum_{i=0}^{2L} w_i^{(c)} (\bar{\chi}^x_{i,t} - \bar{\mu}_t)(\bar{Z}_{i,t} - \hat{z}_t)^\top$
11. $K_t = \Sigma^{x,z}_t S_t^{-1}$
(∗ Update mean and covariance ∗)
12. $\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)$
13. $\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^\top$
14. $p_{z_t} = det(2\pi S_t)^{-\frac{1}{2}} exp\{-\frac{1}{2}(z_t - \hat{z}_t)^\top S_t^{-1}(z_t - \hat{z}_t)\}$
15. **return** μ_t, Σ_t, p_{z_t}

Table 1: Unscented Kalman Filter

where $X^{x,q}_{t-1}$ is the quaternion elements of the augmented matrix X^x_{t-1}, and $\vec{\mu}^q_{t-1}$ denotes the quaternion element of the pre-estimation mean. This operator corresponds to the summations that occur in step 3. After the summation of sigma points, the quaternion part are normalized to eliminate negative weights. A similar rotational error vector was used during the sensor update step.

3. ONLINE DYNAMIC MODELING

Bandit's physical dynamics change over time. Propellant usage effects Bandit's center of mass and moments of inertia, not to mention degradation of thruster performance. There are also many un-modeled or hard to measure parameters such as: thruster misalignment, fluid dynamic losses, and the actual thrust per valve. Dynamic online modeling attempts to capture these time-varying phenomena with no prior knowledge of the thruster dynamics or physical parameters of the spacecraft.

An accurate thruster model is needed for the potential-based controller which selects the thruster firing closest to the desired change in linear and angular velocity [9]. It can also be used to complement the

inaccurate accelerometer input used in the UKF. A similar system in literature had combined a Gaussian Process Regression with an UKF on an autonomous blimp [6] [7]. Online dynamic modeling is also necessary for paramters of unmanned ground vehicles wheel slippage on rough terrains [13].

3.1. Thruster Dynamics

Each of the eight thrusters have a torque, distance and force vector related as such,

$$\vec{t_i} = \vec{d_i} \times \vec{f_i} \tag{10}$$

where $\vec{t_i}$, $\vec{d_i}$ and $\vec{f_i}$ denote torque, distance and force of the ith thruster. The distance vectors originate from the center of mass. The total force and torque is vector additive, where λ_i indicates which thrusters are on.

$$\vec{T}_{thrusters} = \sum_{i=1}^{8} \lambda_i \vec{t_i} \tag{11}$$

$$\vec{F}_{thrusters} = \sum_{i=1}^{8} \lambda_i \vec{f_i} \tag{12}$$

$$\lambda_i \in \{0, 1\}$$

Angular velocity, $\vec{\omega}$ in the fixed-body frame, is modeled as,

$$J\dot{\vec{\omega}} = \vec{T}_{thrusters} - \vec{\omega} \times (J\vec{\omega}) \tag{13}$$

where J is the inertia matrix. We can linearize about the current angular velocity, to find the change in angular velocity given a small change in time, δ_t.

$$\Delta\vec{\omega} = J^{-1}\vec{T}_{thrusters}\delta_t - J^{-1}\vec{\omega} \times (J\vec{\omega})\delta_t \tag{14}$$

For small angular velocities, the angular acceleration due to the current angular velocity interacting with the inertia matrix is orders of magnitude smaller than the acceleration due to the thrusters, and we can therefore safely ignore the second term. Thus, the change in velocity can be linearly approximated in terms of the thruster magnitude.

Given the spacecraft's mass, m, the change in linear velocity can similarly be derived.

$$\Delta\vec{v} = \frac{\delta_t}{m}\vec{F}_{thrusters} + \Delta\vec{v}_{orbital} \tag{15}$$

The change in orbital velocity, $\Delta\vec{v}_{orbital}$ is calculated as described in Section 2.1. By first propagating the state using CW orbital dynamics, we can then isolate the translational thruster acceleration, which is also linear.

3.2. Bayes Linear Regression

Bayes Linear Regression (BLR) is a method of online linear regression which makes the assumption that there is a normal Gaussian matching between input data $\vec{x}$, and output data $\vec{y}$,

$$\vec{y}_t = \vec{\theta}^\top \vec{x}_t + \vec{\epsilon}_t \tag{16}$$

for a given set of weights, $\vec{\theta}$, and Gaussian noise, $\vec{\epsilon}_t \sim N(0, \sigma^2)$. In the the natural or canonical Gaussian parameterization, we can find the updated probability distribution of the weights given a new data point, $(\vec{x}, \vec{y})$:

$$\begin{aligned} p(\vec{y}|\vec{x}, \vec{\theta})p(\vec{\theta}) &\propto e^{\frac{-(y-\theta^\top x)^2}{2\sigma^2}} e^{\frac{-1}{2}\theta^\top P\theta + J^\top\theta} \\ &= e^{\frac{-1}{2}\theta^\top (P - \frac{xx^\top}{\sigma^2})\theta + (J + y\frac{yx^\top}{\sigma^2})^\top\theta} \end{aligned} \tag{17}$$

For each training set, we can derive the following update rule for the information matrix, P, and information vector, J,

$$P \leftarrow P + \frac{\vec{x}_t\vec{x}_t^\top}{\sigma^2} \tag{18}$$

$$J \leftarrow J + \frac{\vec{y}_t\vec{x}_t^\top}{\sigma^2} \tag{19}$$

We can use the same equations to remove a training set from our regression via subtraction; this can create a sliding window of training examples to track time dependent changes. We can transform back to the moment parameterization to find the predicted value, $\vec{y}$ as well as the weight vector, $\vec{\mu}_\theta$ and its variance Σ_θ

$$\vec{\mu}_\theta = P^{-1}J \tag{20}$$

$$\Sigma_\theta = P^{-1} \tag{21}$$

$$\vec{y} = (P^{-1}J)^\top\vec{x} \tag{22}$$

The output, $\vec{y}$, is linear in $\vec{x}$, so the predictive variance is just

$$\Sigma_y = \vec{x}^\top\Sigma_\theta\vec{x} \tag{23}$$

which is a scalar. This only captures uncertainty due to the parameters. The total uncertainty includes the noise variance.

$$\Sigma_y = \vec{x}^\top\Sigma_\theta\vec{x} + \sigma^2 \tag{24}$$

BLR requires the inversion of the information matrix, P during prediction with a time complexity of $O(d^3)$, where d is the number of input features, d = 8 for our example. This small time complexity makes a big difference when run on Bandit's embedded processor. Gaussian Process Regression was also applied to the problem, but the results were less accurate for this application and had a larger time complexity [12].

4. Integration

Prior work has considered the UKF and BLR in isolation, but not working together, [12]. The two techniques can be combined in order to produce enhanced performance. The accelerometer readings used in the

UKF were fused with the BLR output as a weighted sum. This acts to smooth the sensor input, allowing for lower variances. The BLR is not used until it is initialized with a set number of thruster examples. After initialization, the accelerometer readings and BLR acceleration are fused using the following equations.

$$P_{new} = ((P_{acc})^{-1} + (P_{blr})^{-1})^{-1} \quad (25)$$

$$\vec{u}_{new,t} = (\frac{P_{blr}}{P_{blr} + P_{acc}})\vec{u}_{acc} + (\frac{P_{acc}}{P_{blr} + P_{acc}})\vec{y}_{blr} \quad (26)$$

where P_{acc} represents the accelerometer variance, P_{blr}, represents the BLR total output variance found in Equation (25), $\vec{u}_{acc}$ represents the accelerometer readings, and $\vec{y}_{blr}$ represents the BLR output. P_{new} replaces the sub-matrix associated with accelerometer covariance in the Q matrix of the UKF, and u_{new} is used as the accelerometer input to the UKF.

5. RESULTS

For the BLR modeling, testing data was generated from a random walk autopilot consisting of the 256 possible valve configurations. The training set consisted of about 100 seconds of flight with around 500 samples. The BLR training input was a vector of eight binary (0,1) values representing the state of the valve for each valve firing. The training output was generated from the simulated accelerometer and angular gyroscope.

Using Bayes Linear Regression, the linear velocity had a correlation of 0.98 between predicted and actual values, Figure 3, and the angular velocity had a correlation of 0.87, Figure 4. A sliding window of 50 prior examples was used.

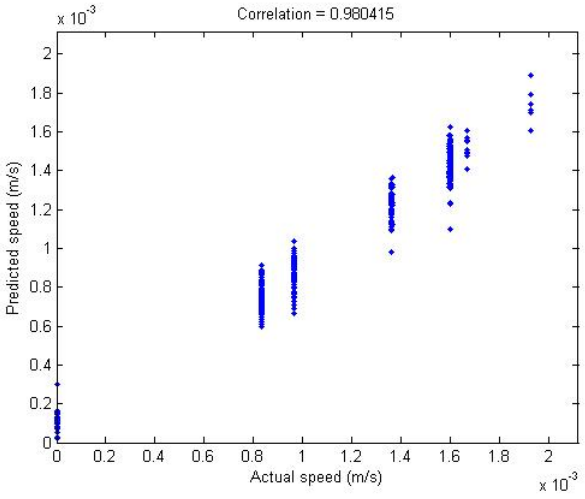

Figure 3: BLR Linear Velocity Correlation.

The combined UKF-BLR was tested using a behavior-based velocity potential-function docking algorithm which starts Bandit at a distance of 0.75 meters from Akoya, and proceeds to autonomously dock

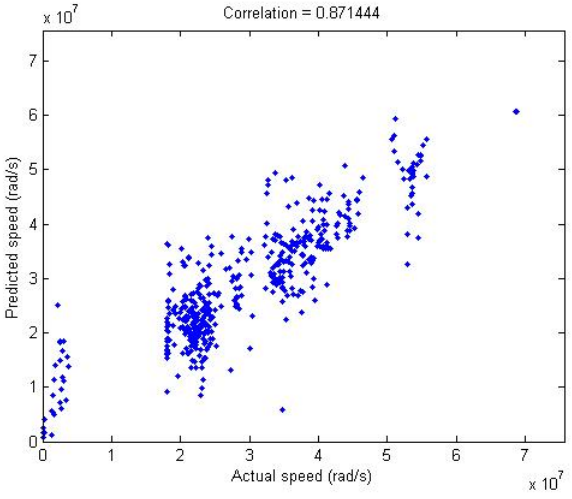

Figure 4: BLR Angular Velocity Correlation.

[14]. The sensor noise added to the simulation is as follows: accelerometer noise = 25%, gyroscope noise = 5%, ImageNav (position) = 5%, ImageNav (attitude) = 1%. A small initial position error was also added to the state estimation.

The UKF error states converge both in position, Figure 5, and velocity, Figure 6. The filter variance quickly converges to a steady state error as only the acceleration sensor output is used. After an initial period, of around six seconds, the BLR estimation is fused with the acceleration sensor output, and the variance further converges.

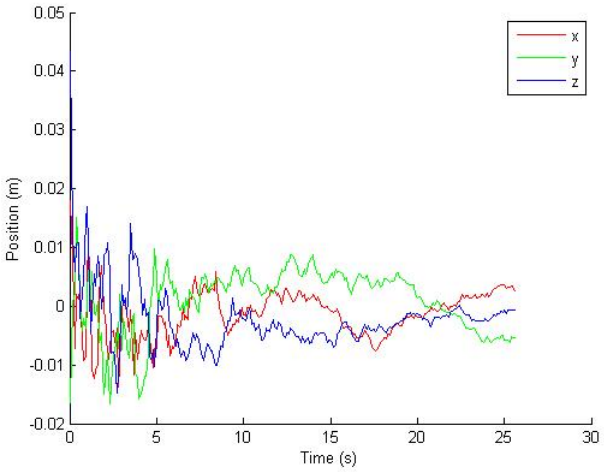

Figure 5: UKF Position Error with BLR.

For comparison, when BLR is not used the position error is larger, Figure 7, and the velocity error variance stays at the initial steady state error, Figure 8. Improvements can also be seen in docking time and number of valves fired, Table 2, which greatly affects system performance.

Even with poor accelerometer performance, the fil-

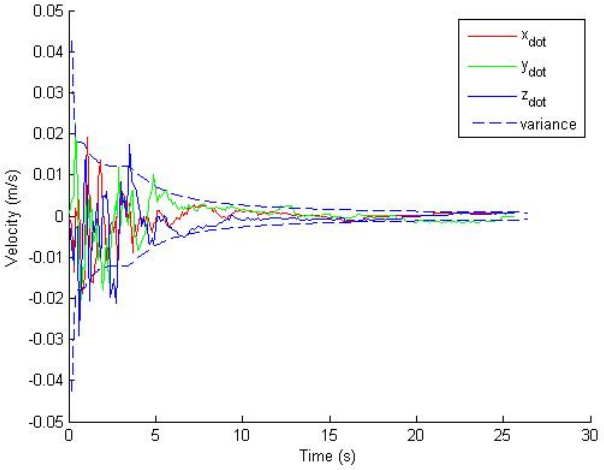

Figure 6: UKF Velocity Error with BLR.

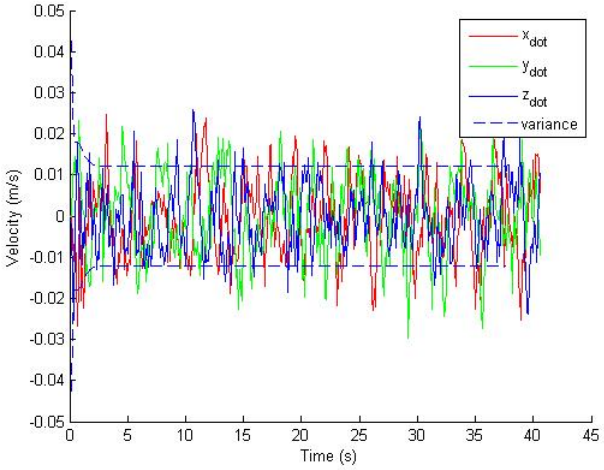

Figure 8: UKF Velocity Error without BLR.

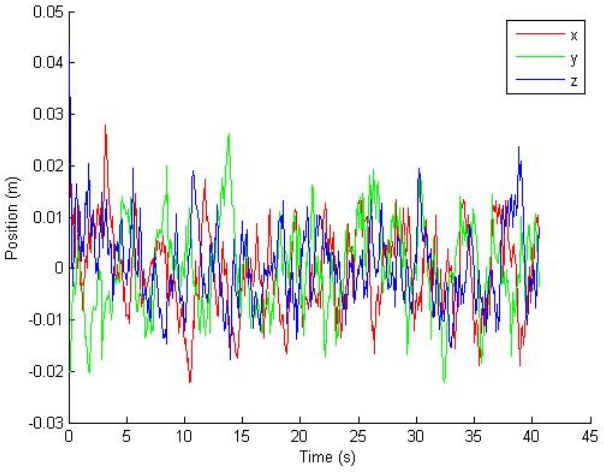

Figure 7: UKF Position Error without BLR.

ter has a maximal estimation error of only 2 cm and 3 cm/s. The maximum errors are made towards the beginning of the initialization of the filter, before the Kalman gain had reached a relatively steady state. The filter, with the help of BLR, does a good job at minimizing the linear velocity error even though the none of the sensors can directly observe it.

6. CONCLUSION

We have successfully shown the Unscented Kalman Filter for pose estimation and Bayes Linear Regression for online thruster dynamic learning for a small inspector spacecraft. The UKF bounded errors in both observed (position and attitude) and un-observed states (linear velocities) while capturing the spacecraft's nonlinear dynamics. Quaternion rotation was handled using rotational error vectors when finding the sigma-points. The image based navigation system help bound the position and attitude errors.

Table 2: Improvement using BLR

	Docking Time	Valve Fired
With BLR	21.1 s	760
Without BLR	41.0 s	3,032

Bayes Linear Regression gave good results, in good time complexity, even with noisy thrusters and inputs. The sliding window of training data allows the online model to capture dynamic changes such as mass loss and thruster performance degradation, which cannot be otherwise detected in the UKF. The current control laws use potential fields to determine a desired linear and angular velocity at a given position around Akoya. The BLR dynamic model will determine the correct thruster firing to drive the velocities to the desired values.

The combined UKF-BLR techniques significantly increased performance by reducing the dependence on the unreliable accelerometer. The senor fusion of the BLR estimate with the accelerometer checked the potential development of a diverging feedback loop and allowed for redundancies in the event that the BLR digresses. In addition, improvements where seen in both docking time and valve firing reduction. This is crucial when dealing with the inaccurate sensors used on very small low-cost spacecraft.

While these techniques are being integrated on the Bandit inspector spacecraft, they can be applied to many other spacecraft and robotic applications with restricted sensors and actuator options. The increased use of small spacecraft for proximity operations will allow for many new interesting missions.

7. ACKNOWLEDGMENTS

The authors are gratefully for the hard work of the Bandit and Akoya teams at Washington University in St. Louis and St. Louis University.

References

[1] M.A. Abidi, and T. Chandar. "Pose estimation for camera calibration and landmark tracking", *IEEE International Conference on Robotics and Automation (ICAR)*, vol.1, pp 420-426, Cincinnati, Ohio, May, 1990.

[2] E. Beck. "Optimizing the Small Satellite Platform for Compelling Technology Demonstrations: Bandit/Akoya Proximity Operations and Rapid Integration", *AIAA/USU Conference on Small Satellites*. Logan, UT, August 2007.

[3] W.H. Clohessy and R.S. Wiltshire. "Terminal Guidance for Satellite Rendezvous", *J. Aerospace Sciences*, Vol 27, p 653, 1960.

[4] S.J. Julier and J.K. Uhlmann. "A new extension of the Kalman filter to nonlinear systems", *International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, Orlando, Florida, April 1997.

[5] W. Khoder, B. Fassinut-Mombot, and M. Benjelloun. "Quaternion unscented Kalman filtering for integrated inertial navigation and GPS", *International Conference on Information Fusion*, Cologne, Germany, June 2008.

[6] J. Ko, D. Klien, D. Fox, and D. Haehnel. "Gaussian Processes and Reinforcement Learning for Identification and Control of an Autonomous Blimp", *International Conference on Robotics and Automation (ICRA)*, Rome, Italy, April 2007.

[7] J. Ko, D. Klien, D. Fox, and D. Haehnel. "GP-UKF: Unscented Kalman Filters with Gaussian Process Prediction and Observation Models", *International Conference on Intelligent Robots and Systems (IROS)*, San Diego, CA, October 2007.

[8] E. Kraft. "A Quaternion-Based Unscented Kalman Filter for Orientation Tracking". *International Conference on Information Fusion*, Cairns, Australia, July 2003.

[9] J. Neubauer. "Controlling Swarms of Bandit Inspector Spacecraft". *AIAA/USU Conference on Small Satellites*. Logan, UT, August 2006.

[10] C.E. Rasmussen, and C.K.I. Willimans. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[11] F. Rogers-Marcovitz, and P. Williams. "DEEP: Dallas EEProm Equipment Profile for Rapid Integration and Automatic System Modeling". *AIAA/USU Conference on Small Satellites*. Logan, UT, August 2007.

[12] F. Rogers-Marcovitz. "Online Dynamic Modeling and Localization for Small-Spacecraft Proximity Operations". *AIAA/USU Conference on Small Satellites*. Logan, UT, August 2009.

[13] F. Rogers-Marcovitz, "On-line Mobile Robotic Dynamic Modeling using Integrated Perturbative Dynamics," tech. report CMU-RI-TR-10-15, Robotics Institute, Carnegie Mellon University, April 2010.

[14] S. Scarritt, and M. Swartwout. "Proximity Navigation of Highly-Constrained Spacecraft", *International Symposium on Space Flight Dynamics*, Annapolis, MD, September 2007.

[15] M. Swartwout. "Bandit: A Platform for Responsive Educational and Research Activities". *Responsive Space Conference*, AIAA-RS4-2006-3004, Los Angeles, CA, April 2006.

[16] M. Swartwout., S. Scarritt, and J. Neubauer. "Potential Function Controllers for Proximity Navigation of Underactuated Spacecraft". *AAS Guidance & Control Conference*, Breckenridge, CO, February 2007.

[17] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2006.