IAA-AAS-DyCoSS1-09-04

HARDWARE-IN-THE-LOOP RENDEZVOUS SIMULATION INVOLVING AN AUTONOMOUS GUIDANCE, NAVIGATION AND CONTROL SYSTEM

Heike Benninghoff[*], Toralf Boge[†], and Tristan Tzschichholz[‡]

The rendezvous process is a key technology in multi-spacecraft missions like on-orbit servicing missions. An active spacecraft (chaser) approaches a passive spacecraft (target) in its orbit by performing controlled orbit and attitude maneuvers. The paper presents an autonomous guidance, navigation and control system for rendezvous using a monocular camera as vision-based sensor for relative navigation. Image processing algorithms and navigation filters are employed to get accurate information about the relative position and attitude between the two spacecrafts. The rendezvous sensor and the entire GNC system is tested and verified at DLR's robotic-based test bed European Proximity Operations Simulator 2.0.

INTRODUCTION

Rendezvous and docking/berthing (RVD/B) operations are key technologies in multi-spacecraft mission, i.e. in missions involving two or more spacecrafts. The rendezvous process consists of several controlled orbital maneuvers in which an active spacecraft (called *chaser*) approaches a passive spacecraft (called *target*) in its orbit. In the consecutive mating phase (docking or berthing) the target is captured and the two spacecrafts are structurally connected.[1]

RVD/B was performed in numerous US space shuttle missions, in missions involving the European Automated Transfer Vehicle (ATV) or the Japanese H-II Transfer Vehicle (HTV) to give a few examples. In the future, RVD/B technology is going to be employed in interplanetary missions, like the Mars Sample Return mission, involving a lander vehicle which is separated from its mother ship and returns to it after exploration of the planet. Further, servicing and inspection missions are in preparation making use of RVD/B. In so-called On-Orbit Servicing (OOS) missions, an active spacecraft (also called *servicer*) approaches a client spacecraft in its orbit. The client is usually affected by some damage, for example its orbit or attitude control can be lost, the spacecraft can be out of fuel, the solar arrays can be damaged and/or it is no longer operational. Consider a communication satellite where only the attitude control has been lost, whereas the payload and the other subsystems are still operational. In this case, a service satellite can approach and dock on the client satellite in order to overtake its attitude control. Apart from life time extension, OOS also aims at safe end-of-life de-orbiting. At the end of a space mission controlled de-orbiting of a satellite is not always possible, for example if the communication between satellite and ground station is broken. In this case a service satellite is inserted in the vicinity of the target, captures the target,

[*]German Aerospace Center (DLR), 82234 Wessling, Germany
[†]German Aerospace Center (DLR), 82234 Wessling, Germany
[‡]German Aerospace Center (DLR), 82234 Wessling, Germany, University of Wuerzburg, 97074 Wuerzburg, Germany

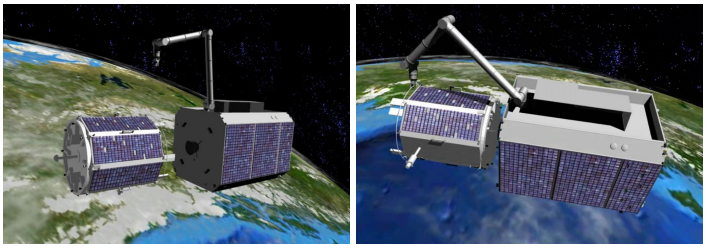

Figure 1. Capture of a client satellite in the DEOS mission

overtakes its orbit and attitude control and a de-orbiting takes place. In a coupled configuration a re-entry in the Earth's atmosphere is performed or the target spacecraft is brought to a graveyard orbit. With respect to decreasing space debris servicing missions of this kind will become more and more important in the future. Figure 1 shows one capturing scenario planned for a future OOS mission called DEOS (Deutsche Orbitale Servicing Mission). DEOS is a German mission where capturing of a tumbling, non-cooperative LEO client satellite with a service spacecraft should be demonstrated followed by several experiments and a final de-orbiting in a coupled configuration within a pre-defined orbit corridor.[2]

Whereas RVD/B has become state of the art in manned spaceflight, new challenges arise for RVD/B in servicing and inspection missions. First, a typical target satellite was not intended for rendezvous and docking tasks when it was built. Hence, one has to cope with a completely passive, uncooperative target. Secondly, most targets are no longer operative and control over their orbit and attitude could be lost. Therefore, one has to expect a freely tumbling target performing arbitrary rotational movements which impedes the relative navigation during the approach. In addition, it is desired that the complete approach makes use of an autonomous guidance, navigation and control (GNC) concept such that no human intervention on ground is needed. The autonomy constraint poses a further challenge to the on-board control system. Consequently, a complex GNC concept has to be developed for a safe rendezvous process.

Before a critical mission like an on-orbit servicing mission can be launched, the navigation sensors and the GNC system have to be tested and verified intensively. For this purpose, German Aerospace Center has established a robotic-based test bed called European Proximity Operations Simulator (EPOS 2.0) which can be used for hardware-in-the-loop (HIL) simulations of the last 25m of the rendezvous phase and for simulating of the final docking/berthing.[3]

This paper presents an autonomous GNC system using a monocular camera as vision based sensor for relative navigation in the rendezvous phase. Image segmentation methods based on edge detection are applied to track continuously the target in the given camera images. The image processing results are further processed in dynamic navigation filters to get an estimation of the current relative position and orientation between servicer and target. The accurate information about the relative state is significant for a safe approach to the target and is therefore used for the chaser's position and attitude control.

This work further describes the testing of the sensor and the GNC system in the framework of real-time HIL simulations at the EPOS facility. Several experiments are performed simulating different test scenarios like a continuous straight-line approach or the behavior at hold points. Further, the performance of the navigation system, the control system and the total performance is investigated.

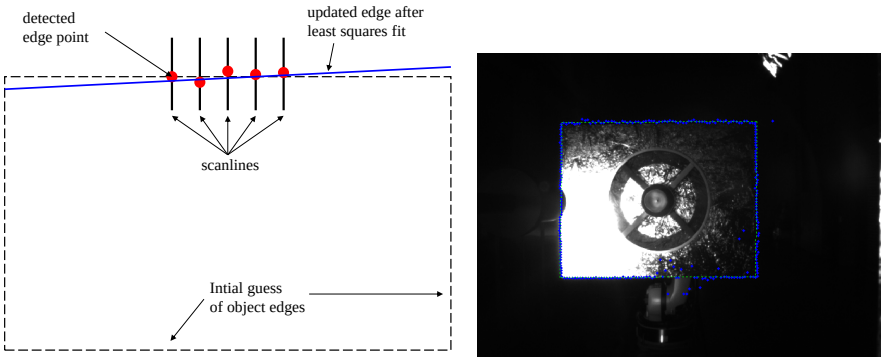

Figure 2. Illustration (left) and results (right) of the edge tracking algorithm

AN AUTONOMOUS GUIDANCE, NAVIGATION AND CONTROL SYSTEM

Image Segmentation and Object Tracking

Image segmentation aims at dividing a given camera image into several segments and especially in separating objects in the foreground from their background. Assume an 8-bit gray-scaled image which is given by its image intensity function, i.e. a function $I : \Omega \to [0, 255]$ where $\Omega \subset \mathbb{R}^2$ is a rectangular domain. $I(x) = 255$ corresponds to white, $I(x) = 0$ to black. To detect edges that separate objects in the image from its background one searches for points where ∇I is big.

The used image segmentation method is based on an algorithm presented in Reference 4. It assumes that a rough estimate of the four corner points is available. Typically, the result of the previous time step provided by the navigation filter is used as initial guess. Four edges are constructed by simply connecting the given corner points, see Figure 2 (left). Perpendicular to these edges a set of lines (so-called *scanlines*) is constructed. Let $\vec{\nu} \in \mathbb{R}^2$ denotes the direction of one scanline. Along each line one searches for the pixel at which the derivative of the image intensity function in the direction of $\vec{\nu}$, i.e. $\nabla I \cdot \vec{\nu}$ attains its maximum absolute value. Consequently, one obtains one edge point for each scan-line. This results in four sets of *detected edge points*, one set for each edge of the rectangular object. The random sample consensus (RANSAC)[5] algorithm is applied to remove outliers. Finally, four new edges are constructed by performing a least squares fit using the sets of detected edge points and the corner points are obtained by intersection of the edges. The image coordinates of the four corner points are further processed in dynamical navigation filters:

State Estimation using Dynamic Filtering

Introduction to Kalman Filtering The Kalman filter is a state estimation technique making use of prior knowledge of the state combined with measurements observed over time.[6] Therefore, consider a time-discrete system of the form

$$x_{k+1} = f(x_k, u_k, t_k, \nu_k), \tag{1}$$

$$z_k = h(x_k, t_k, \omega_k), \tag{2}$$

where $x_k \in \mathbb{R}^n$ denotes the state vector, $u_k \in \mathbb{R}^k$ the input or control vector, $\nu_k \in \mathbb{R}^n$ the system noise, $z_k \in \mathbb{R}^m$ the measurement and $\omega_k \in \mathbb{R}^m$ the measurement noise vector at time t_k. Equation (1) is called *system model*, Equation (2) is called *measurement model*.

The Kalman filter method assumes that ν_k and ω_k are zero-mean, Gaussian noise with known covariances Q_k and R_k. The method states a recursive solution to the filtering problem (1)-(2). In detail: The filter consists of predictor and corrector steps to determine an estimation of the state $\hat{x}_k$ and an estimation of $\hat{P}_k$, the covariance of $x_k - \hat{x}_k$. The idea is to find an optimal state estimation by minimizing the covariance $\hat{P}_k$. Therefore, the Kalman filter can be interpreted as a *minimum variance estimator*.

The predictor step uses the system model and sets the predicted state estimate $\tilde{x}_{k+1}$ to the mean of $f(\hat{x}_k, u_k, t_k, \nu_k)$. If f is linear, the mean is $f(\hat{x}_k, u_k, t_k, 0)$, because ν_k has zero mean. In the corrector step the predicted estimate $\tilde{x}_{k+1}$ is updated by setting

$$\hat{x}_{k+1} = \tilde{x}_{k+1} + K_k(z_{k+1} - \tilde{z}_{k+1}),$$

where K_k is the so-called gain matrix which serves as a weighting factor and is dependent on the statistics of ν_k and ω_k and $\tilde{z}_{k+1}$ is the mean of $h(\tilde{x}_{k+1}, t_{k+1}, \omega_{k+1})$. In fact, K_k is chosen such that the state error covariance $\hat{P}_{k+1}$ is minimized. If h is a linear mapping, the mean is $h(\tilde{x}_{k+1}, t_{k+1}, 0)$.

In addition, the state error covariance $\hat{P}_{k+1} = \mathbb{E}((\hat{x}_{k+1} - x_{k+1})(\hat{x}_{k+1} - x_{k+1})^T)$ is determined by a predictor and corrector step. Details are omitted here and can be found in various textbooks dealing with the Kalman filter (see for example Reference 7).

Two main limitations and problems of the Kalman filter arise:

1. It assumes Gaussian, zero-mean noise and detailed knowledge about the statistics, i.e. knowledge of the covariances of ν_k and ω_k.
2. Statistical knowledge about $f(x_k, u_k, t_k, \nu_k)$ and $h(x_k, t_k, \omega_k)$ is not known if f and h are non-linear. Their mean and covariance can be calculated analytically in only rare cases. (Recall, that if y is a random variable and g a non-linear function, $\mathbb{E}(g(y)) \neq g(\mathbb{E}(y))$.)

In practice, the constraint to have zero-mean Gaussian noise is rarely fulfilled. Concerning camera measurements and noise analysis done at EPOS 2.0, we observe non-Gaussian noise, see Figure 3 which shows histograms of measurements of the four corner points (in pixel coordinates) of a rectangular target object visible in the images. For comparison, a Gaussian density with the corresponding mean and variance is plotted (red line). We clearly observe non-Gaussian probability densities. Further, the system dynamics and the measurement model used for relative navigation that we employ are both non-linear (see subsections below).

Facing these problems we applied advanced filter techniques: the Extended Kalman filter (EKF) and the Unscented Kalman filter (UKF). Both techniques approximate the involved non-linear functions. The following section introduces the basic ideas:

Approximation of a Non-Linear Function One main problem is the determination of the mean value of $f(x_k, u_k, t_k, \nu_k)$ and $h(x_k, t_k, \omega_k)$ and the calculation of their covariances. First, we focus on the more general task:

Given a probability space (Ω, Σ, P) where Σ is the Borel σ-algebra on $\Omega \subset \mathbb{R}^n$ and $P : \Sigma \to \mathbb{R}$ a probability measure on Σ. Let $f : \Omega \to \mathbb{R}^n$ be a (with respect to P) integrable, vector-valued function. Calculate the mean $\overline{y}$ and the covariance P_{yy} of $y = f(x)$, i.e. calculate

$$\overline{y} = \mathbb{E}(y) = \int_\Omega f(x)\, \mathrm{d}P(x) \quad \text{and} \quad P_{yy} = \mathbb{E}((y - \overline{y})(y - \overline{y})^T).$$

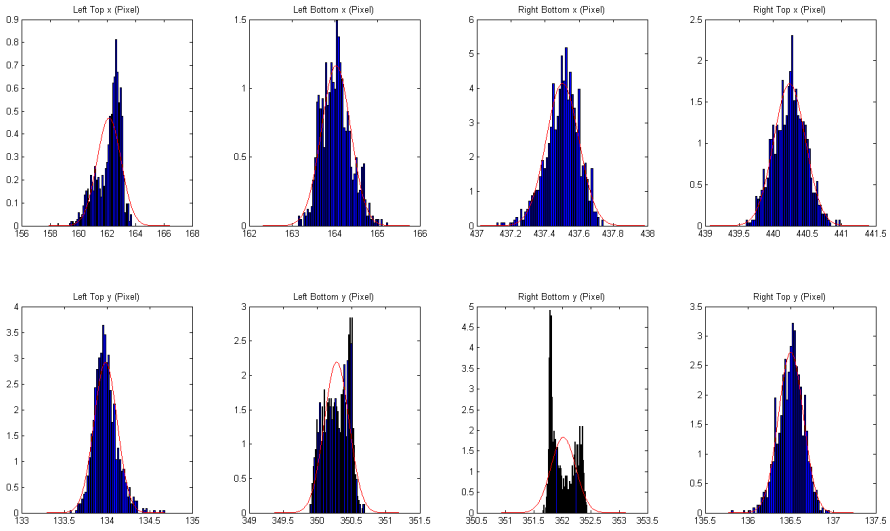

Figure 3. Camera measurement noise analysis

In the special case that f is linear, i.e. $f(x) = Ax + b$, $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, the mean is given by

$$\overline{y} = \int_\Omega f(x)\,\mathrm{d}P(x) = A\int_\Omega x\,\mathrm{d}P(x) + b = A\overline{x} + b.$$

Further, one can show by some calculus

$$P_{yy} = \mathbb{E}((y-\overline{y})(y-\overline{y})^T) = \mathbb{E}((Ax - A\overline{x})(Ax - A\overline{x})^T) = \ldots = AP_{xx}A^T,$$

where $P_{xx} := \mathbb{E}((x-\overline{x})(x-\overline{x})^T)$. Hence, in the linear case one can directly calculate the statistical quantities $\overline{y}$ and P_{yy} from the statistics $\overline{x}$ and P_{xx} of x.

In the general case that f is non-linear, the calculation of $\overline{y}$ and P_{yy} (if the integrals exist) is more complicated. Usually closed-form solutions do not exist. Therefore, one tries to find approximations of $\overline{y}$ and P_{yy}. We will consider two approaches:

Taylor series approximation Assume f is k times differentiable in a neighborhood D of $\overline{x}$. Set $\delta x := x - \overline{x}$. Taylor theorem states the existence of functions $h_\alpha : D \to \mathbb{R}^n$ with $\lim_{x\to\overline{x}} h_\alpha(x) = 0$ such that

$$f(x) = \sum_{i=0}^{k} \sum_{\alpha\in\mathbb{N}^n, |\alpha|=i} \frac{D^\alpha f(\overline{x})}{\alpha!}\delta x^\alpha + \sum_{\alpha\in\mathbb{N}^n,|\alpha|=k} h_\alpha(x)\delta x^\alpha,$$

where $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$ is a multi-index, $|\alpha| := \alpha_1 + \ldots + \alpha_n$, $\alpha! := \alpha_1! \ldots \alpha_n!$, and

$$D^\alpha f(\overline{x}) = \frac{\partial^{|\alpha|} f(\overline{x})}{\partial^{\alpha_1} x_1 \ldots \partial^{\alpha_n} x_n}$$

and $\delta x^\alpha = \delta x_1^{\alpha_1} \ldots \delta x_n^{\alpha_n}$.

Now, consider $k = 2$. Set $H_2(x) := \sum_{\alpha\in\mathbb{N}^n,|\alpha|=2} h_\alpha(x)\delta x^\alpha$ and let $\nabla f(\overline{x}) \in \mathbb{R}^{n\times n}$ denote the gradient of f at $\overline{x}$ defined by $(\nabla f(\overline{x}))_{ij} = \frac{\partial}{\partial x_j} f_i(\overline{x})$ for $i, j = 1, \ldots n$ and $\mathrm{Hesse}_f(\overline{x}) \in (\mathbb{R}^n)^{n\times n}$

the Hesse-matrix of f defined by $(\mathrm{Hesse}_f(\overline{x}))_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} f(\overline{x}) \in \mathbb{R}^n$. In this subsection, the subscript i denotes the i-th component of a vector in $\mathbb{R}^n$, the subscript ij the matrix entry of the i-th row and j-th column. Then we have

$$y = f(x) = f(\overline{x}) + \nabla f(\overline{x})\delta x + \frac{1}{2}\delta x^T \mathrm{Hesse}_f(\overline{x})\delta x + H_2(x). \tag{3}$$

Taking expectations leads to

$$\begin{aligned}\overline{y} &= f(\overline{x}) + \mathbb{E}\left(\nabla f(\overline{x})\delta x\right) + \frac{1}{2}\mathbb{E}\left(\delta x^T \mathrm{Hesse}_f(\overline{x})\delta x\right) + \mathbb{E}\left(H_2(x)\right) \\ &= f(\overline{x}) + 0 + \frac{1}{2}\sum_{i,j=1}^{n} (\mathrm{Hesse}_f(\overline{x}))_{ij}(P_{xx})_{ij} + \mathbb{E}\left(H_2(x)\right)\end{aligned} \tag{4}$$

Here, we used the linearity of the mean and $\mathbb{E}(\delta x_i) = \mathbb{E}(x_i - \overline{x}_i) = 0$ for $i = 1, \ldots, n$ as well as $\mathbb{E}(\delta x_i \delta x_j) = \mathbb{E}\left((x - \overline{x})_i (x - \overline{x})_j\right) = (P_{xx})_{ij}$. A linearization of first order neglects all terms of order 2 (terms with $\delta x_i \delta x_j$) or higher in Equation (4). This leads to the approximation (on noting that $H_2(x)$ and P_{xx} contain terms of order 2)

$$\overline{y} \approx f(\overline{x}) =: \hat{y}. \tag{5}$$

The variance $P_{yy} = \mathbb{E}((y - \overline{y})(y - \overline{y})^T)$ is calculated using Equation (3) and (4). Subtracting the equations results in

$$y - \overline{y} = \nabla f(\overline{x})\delta x + \frac{1}{2}\sum_{i,j=1}^{n} (\mathrm{Hesse}_f(\overline{x}))_{ij} \left(\delta x_i \delta x_j - (P_{xx})_{ij}\right) + H_2(x) - \mathbb{E}(H_2(x)).$$

We obtain

$$P_{yy} = \mathbb{E}((y - \overline{y})(y - \overline{y})^T) = \nabla f(\overline{x}) P_{xx} (\nabla f(\overline{x}))^T + \text{ t.h.o.},$$

where t.h.o denotes *terms of higher order* which covers all additional terms which are of order 4 or higher. Neglecting all terms of order greater or equal 4 leads to the simple approximation

$$P_{yy} \approx \nabla f(\overline{x}) P_{xx} (\nabla f(\overline{x}))^T =: \hat{P}_{yy}. \tag{6}$$

In summing up, we found approximations $\hat{y}$ for the mean and $\hat{P}_{yy}$ for the covariance.

Unscented approximation Unscented approximation goes back to the work of S. Julier and J. Uhlmann.[8,9] The idea of unscented approximation is to chose a deterministic set of points in a neighborhood of $\overline{x}$ (so-called *sigma points*) such that their mean is $\overline{x}$ and their covariance is P_{xx}. The non-linear function f is applied to the sigma points and a set of transformed points is constructed, see Figure 4. From the set of transformed points $\overline{y}$ and P_{yy} are constructed.

In detail, given an $\overline{x} \in \Omega \subset \mathbb{R}^n$ a set of $2n + 1$ sigma points is constructed by setting

$$\begin{aligned} x^{(0)} &:= \overline{x}, \\ x^{(i)} &:= \overline{x} + \left(\sqrt{(n + \kappa)P_{xx}}\right)_i, \quad i = 1, \ldots n, \\ x^{(i+n)} &:= \overline{x} - \left(\sqrt{(n + \kappa)P_{xx}}\right)_i, \quad i = 1, \ldots n, \end{aligned} \tag{7}$$

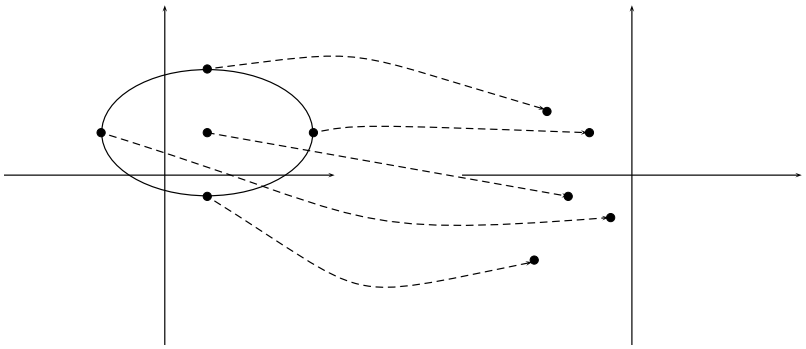

Figure 4. Transformation of sigma points under a non-linear function

where $\left(\sqrt{(n+\kappa)P_{xx}}\right)_i$ denotes the i-th column of the matrix $\sqrt{(n+\kappa)P_{xx}}$ and $\kappa \in \mathbb{R}$ is a tuning parameter. For example, if x is assumed to be Gaussian, Julier and Uhlmann recommend to set $\kappa + n = 3$. It can be easily proved that their mean is $\overline{x}$ and their covariance is P_{xx}.[9]

The unscented method approximates the mean of $y = f(x)$ by a weighted sum of $f(x^{(i)})$:

$$\hat{y} := \sum_{i=0}^{2n} W^{(i)}\, f(x^{(i)}), \tag{8}$$

where the weights $W^{(i)} \in \mathbb{R}$, $i = 0, \ldots, 2n$, are set to

$$W^{(0)} := \frac{\kappa}{n+\kappa}, \quad W^{(i)} := \frac{1}{2(n+\kappa)}, \quad i = 1, \ldots, 2n. \tag{9}$$

The covariance is approximated by the following weighted outer product

$$\hat{P}_{yy} := \sum_{i=0}^{2n} W^{(i)} (f(x^{(i)}) - \hat{y})(f(x^{(i)}) - \hat{y})^T \tag{10}$$

Comparison and Discussion Taylor approximation and unscented approximation follow a completely different approach. The linear Taylor approximation truncates the Taylor series and second and higher terms are neglected whereas the unscented approximation makes use of a weighted sum using $2n + 1$ deterministically chosen points. Both methods are simple with small computational effort and therefore suitable for real-time applications. The effort of linear Taylor approximation is one single function evaluation and one gradient computation and evaluation. The unscented approximation needs $2n + 1$ function evaluations.

One main drawback of the Taylor approximation is the gradient computation which can result in a high effort for some functions. In some cases, the gradient cannot be analytically calculated so the gradient has to be approximated numerically by finite differences derivatives or by automatic differentiation. In this case, further numerical errors will influence the accuracy of the estimation. In our application (see subsections below), non-linear system and measurement functions are involved. Further, the measurement function is given as a composition of functions. This would result

in multiple application of the chain rule to calculate the gradient which is quite expensive. The unscented approximation has a small additional effort due to the construction of sigma points and weights. Therefore the matrix square root $\sqrt{(n+\kappa)P_{xx}}$ has to be calculated. This is an easy task, if P_{xx} is diagonal. Otherwise, an efficient solver like Cholesky decomposition can be used.

The approximation order of linear Taylor approximation is 1 which means that the mean of $f(x)$ for polynomials f of order 1 is calculated exactly. (See example $f(x) = Ax + b$ above.) Julier and Uhlmann showed that the approximation order of unscented transformation is at least 2.[8,9] They further state that one can achieve even higher approximation orders by tuning κ. Thus, κ can be regarded as an extra degree of freedom.

Application on Filtering We now return to the problem (1)-(2) and state two filtering techniques for non-linear system and/or measurement models which distinguish in the way how to approximate mean and covariance. The Extended Kalman Filter (EKF)[6,7] approximates the statistics of f (system model) and h (measurement model) by linear Taylor approximation. The Unscented Kalman Filter (UKF)[8,9] approximates these quantities by unscented transformation, i.e. by a weighted sum and a set of sigma points.

For initialization choose estimates $\hat{x}_0 \approx x(t_0)$ and $\hat{P}_0 \approx P(t_0)$ and set $k = 0$.

Step 1 (EKF): Propagation / Prediction:

Predicted State:	$\tilde{x}_{k+1}$	$= f(\hat{x}_k, u_k, t_k, 0)$
Predicted Covariance:	$\tilde{P}_{k+1}$	$= \nabla_x f(\hat{x}_k, u_k, t_k, 0)\hat{P}_k \nabla_x f(\hat{x}_k, u_k, t_k, 0)^T + Q_k$
Predicted Measurement:	$\tilde{z}_{k+1}$	$= h(\tilde{x}_{k+1}, t_{k+1}, 0)$
Predicted Innovation Cov.:	$\tilde{P}_{\nu\nu,k+1}$	$= \nabla_x h(\tilde{x}_{k+1}, t_{k+1}, 0)\tilde{P}_{k+1}\nabla_x h(\tilde{x}_{k+1}, t_{k+1}, 0)^T + R_{k+1}$
Predicted Cross-Correlation:	$\tilde{P}_{xz,k+1}$	$= \tilde{P}_{k+1}\nabla_x h(\tilde{x}_{k+1}, t_{k+1}, 0)^T$

Step 1 (UKF): Propagation / Prediction:

Sigma Points, Weights	Calculate $\hat{x}_k^{(i)}$ and $W^{(i)}$, $i = 0, \ldots, 2n$ using Equation (7),(9) applied on $\hat{x}_k$ and $\hat{P}_k$	
Predicted State:	$\tilde{x}_{k+1}$	$= \sum_{i=0}^{2n} W^{(i)} f(\hat{x}_k^{(i)}, u_k, t_k, 0)$
Predicted Sigma Points:	$\tilde{x}_{k+1}^{(i)}$	$= f(\hat{x}_k^{(i)}, u_k, t_k, 0)$
Predicted Covariance:	$\tilde{P}_{k+1}$	$= \sum_{i=0}^{2n} W^{(i)}(\tilde{x}_{k+1}^{(i)} - \tilde{x}_{k+1})(\tilde{x}_{k+1}^{(i)} - \tilde{x}_{k+1})^T$
Predicted Measurement:	$\tilde{z}_{k+1}$	$= \sum_{i=0}^{2n} W^{(i)} h(\tilde{x}_{k+1}^{(i)}, t_{k+1}, 0)$
Predicted Innovation Cov.:	$\tilde{P}_{\nu\nu,k+1}$	$= \sum_{i=0}^{2n} W^{(i)}(h(\tilde{x}_{k+1}^{(i)}, t_{k+1}, 0) - \tilde{z}_{k+1})(h(\tilde{x}_{k+1}^{(i)}, t_{k+1}, 0) - \tilde{z}_{k+1})^T + R_{k+1}$
Predicted Cross-Correlation:	$\tilde{P}_{xz,k+1}$	$= \sum_{i=0}^{2n} W^{(i)}(\tilde{x}_{k+1}^{(i)} - \tilde{x}_{k+1})(h(\tilde{x}_{k+1}^{(i)}, t_{k+1}, 0) - \tilde{z}_{k+1})^T$

Step 2 (EKF and UKF): Measurement Update / Correction:

Filter Gain:	K_{k+1}	$= \tilde{P}_{xz,k+1}\left(\tilde{P}_{\nu\nu,k+1}\right)^{-1}$
Updated State:	$\hat{x}_{k+1}$	$= \tilde{x}_{k+1} + K_{k+1}(z_{k+1} - \tilde{z}_{k+1})$
Updated Covariance:	$\hat{P}_{k+1}$	$= \tilde{P}_{k+1} - K_{k+1}\tilde{P}_{\nu\nu,k+1}K_{k+1}^T$

Set $k = k + 1$ and go to the step 1.

System Model for Relative Navigation The filter should give an estimate of the relative position of the service spacecraft w.r.t. the target and an estimation of the client's attitude. The attitude of the servicer is assumed to be known; provided by an accurate sensor like star cameras, gyroscope, etc. The relative position of the service spacecraft is described in the Clohessy Wiltshire (CLW) coordinate framework whose origin is aligned with the center of mass of the target spacecraft.[1] Thus, the chaser's position is seen in the local orbital frame of the target. Table 1 explains the axes of the CLW coordinate system.

Table 1. Clohessy Wiltshire coordinate framework

Axis	Axis name	Description
x	V-Bar	Tangential direction, i.e. direction of the orbital velocity vector
y	H-Bar	Opposite direction of the angular momentum vector of the orbit, i.e. parallel to the normal vector of the orbit plane
z	R-Bar	Direction to Earth, i.e. radial from the spacecraft's center of mass to the center of the Earth

The Hill equations are a system of linear, ordinary differential equations describing the position $p = (p_x, p_y, p_z)$ and the velocity $v = (\dot{p}_x, \dot{p}_y, \dot{p}_z) = (v_x, v_y, v_z)$ of the service satellite in the CLW framework:[1]

$$\ddot{p}_x = 2\omega_0 \dot{p}_z + \frac{1}{m} f_x, \tag{11}$$

$$\ddot{p}_y = -\omega_0^2 p_y + \frac{1}{m} f_y, \tag{12}$$

$$\ddot{p}_z = -2\omega_0 \dot{p}_x + 3\omega_0^2 p_z + \frac{1}{m} f_z, \tag{13}$$

where ω_0 denotes the angular rate of the target orbit, m the mass of the servicer, $f = (f_x, f_y, f_z)$ the sum of external forces acting on the satellite. $\dot{u}$ denotes the time derivative of a time-dependent function u, $\ddot{u}$ the second derivative of u with respect to time.

The client spacecraft's attitude is described by Euler angles and quaternions. The attitude kinematics are given by the quaternion differential equation:[10]

$$\dot{q} = \frac{1}{2} \Omega(\omega) q, \tag{14}$$

where $q = (q_x, q_y, q_z, q_s)$ is the quaternion, $\omega = (\omega_x, \omega_y, \omega_z)$ is the attitude rate and

$$\Omega(\omega) = \begin{pmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{pmatrix}.$$

The attitude dynamics are described by the Euler equation:[10]

$$I\dot{\omega} = T - \omega \times (I\omega). \tag{15}$$

Here, I denotes the moment of inertia of the satellite and $T = (T_x, T_y, T_z)$ is the sum of all external torques acting on the satellite.

The state vector x and the input vector u are set to

$$\begin{aligned} x &:= (p_x, p_y, p_z, v_x, v_y, v_z, q_x, q_y, q_z, q_s, \omega_x, \omega_y, \omega_z)^T && \in \mathbb{R}^{13}, \\ u &:= (f_x, f_y, f_z, T_x, T_y, T_z) && \in \mathbb{R}^6. \end{aligned}$$

Equation (13), (14), (15) can be summarized to

$$\dot{x} = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \\ 2\omega_0 x_6 + \frac{1}{m}u_1 \\ -\omega_0^2 x_2 + \frac{1}{m}u_2 \\ -2\omega_0 x_4 + 3\omega_0^2 x_3 + \frac{1}{m}u_3 \\ 0.5(x_{13}x_8 - x_{12}x_9 + x_{11}x_{10}) \\ 0.5(-x_{13}x_7 + x_{11}x_9 + x_{12}x_{10}) \\ 0.5(x_{12}x_7 - x_{11}x_8 + x_{13}x_{10}) \\ 0.5(-x_{11}x_7 - x_{12}x_8 - x_{13}x_9) \\ (I_2 - I_3)x_{12}x_{13} + u_4 \\ (I_3 - I_1)x_{11}x_{13} + u_5 \\ (I_1 - I_2)x_{11}x_{12} + u_6 \end{pmatrix} =: g(x, u, t, 0),$$

where x_i and u_i denotes the i-th coordinate of $x \in \mathbb{R}^{13}$ or $u \in \mathbb{R}^6$, respectively. Here, we assume I to be diagonal, i.e. $I = \mathrm{diag}(I_1, I_2, I_3)$. I_i, $i = 1, 2, 3$ are the principal moments of inertia.

Now let $x_k = x(t_k)$, $u_k = u(t_k)$, $\Delta t = t_{k+1} - t_k$. Solving the ordinary differential equation approximately with the Euler method leads to the state transition equation (system model)

$$x_{k+1} = x_k + \Delta t g(x_k, u_k, t_k, 0) =: f(x_k, u_k, t_k, 0). \tag{16}$$

The real system can be expressed as $x_{k+1} = f(x_k, u_k, t_k, \nu_k)$ with ν_k being the system noise describing the uncertainty of our model.

Measurement Model for a Monocular Camera Sensor We consider a cuboid target with one rectangular face pointing in camera direction. From the camera images we extract the pixel coordinates

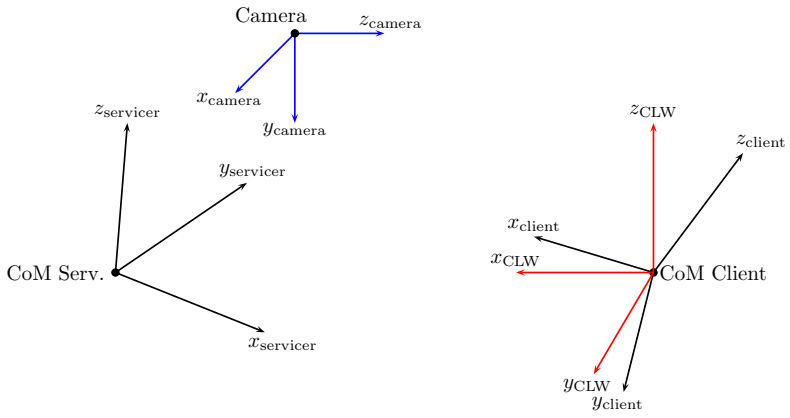

Figure 5. Coordinate systems

of the four corner points $c_1, \ldots, c_4 \in \mathbb{R}^2$ of that face. Consequently, the measurement vector is defined as

$$Z = (c_{1x}, c_{1y}, \ldots, c_{4x}, c_{4y}) \in \mathbb{R}^8. \tag{17}$$

We now derive the measurement model $Z = h(x, t, 0)$. Figure 5 shows the CLW coordinate system as well as the servicer's and the client's body frame and the camera frame. We assume that the following information is available:

$r^{\text{clw}}_{\text{serv}}$	position of servicer in CLW frame
$T^{\text{clw}}_{\text{serv}}$	orientation of servicer in CLW frame
$T^{\text{clw}}_{\text{client}}$	orientation of client in CLW frame
$r^{\text{serv}}_{\text{cam}}$	position of the camera in servicer body frame
$T^{\text{serv}}_{\text{cam}}$	orientation of the camera in servicer body frame
l, w	length and width of the rectangular target face

Note that, $r^{\text{clw}}_{\text{serv}} = (x_1, x_2, x_3)$ and $T^{\text{clw}}_{\text{client}}$ is the rotation matrix which associated with the quaternion (x_7, x_8, x_9, x_{10}). q^{clw}_{serv} and thus $T^{\text{clw}}_{\text{servicer}}$ are given (for example provided by some other attitude sensor) and further $r^{\text{serv}}_{\text{cam}}$ and $T^{\text{serv}}_{\text{cam}}$ are known (e.g. previously determined by calibration of the sensor).

Let $c_i^{\text{client}} \in \mathbb{R}^3$ be the four corner points of the target given in the client's reference system:

$$c_1^{\text{client}} = \begin{pmatrix} 0 \\ -l/2 \\ w/2 \end{pmatrix}, \quad c_2^{\text{client}} = \begin{pmatrix} 0 \\ -l/2 \\ -w/2 \end{pmatrix}, \quad c_3^{\text{client}} = \begin{pmatrix} 0 \\ l/2 \\ -w/2 \end{pmatrix}, \quad c_4^{\text{client}} = \begin{pmatrix} 0 \\ l/2 \\ w/2 \end{pmatrix},$$

For $i = 1, \ldots, 4$ the corresponding position in CLW system are given by $c_i^{\text{clw}} = T^{\text{clw}}_{\text{client}} c_i^{\text{client}}$. The position of the camera in CLW is $r^{\text{clw}}_{\text{cam}} = T^{\text{clw}}_{\text{serv}} r^{\text{serv}}_{\text{cam}}$. Let $r^{\text{clw}}_{\text{cam}\to c_i}$ denote the vector in the CLW system connecting the camera and the i-th corner. In detail:

$$r^{\text{clw}}_{\text{cam}\to c_i} = -r^{\text{clw}}_{\text{cam}} - r^{\text{clw}}_{\text{serv}} + c_i^{\text{clw}} = -T^{\text{clw}}_{\text{serv}} r^{\text{serv}}_{\text{cam}} - r^{\text{clw}}_{\text{serv}} + T^{\text{clw}}_{\text{client}} c_i^{\text{client}}$$

Transforming this vector into the camera 3D coordinate system results in

$$r^{\text{cam}}_{\text{cam}\to c_i} = T^{\text{cam}}_{\text{serv}} T^{\text{serv}}_{\text{clw}} r^{\text{clw}}_{\text{cam}\to c_i} = -T^{\text{cam}}_{\text{serv}} r^{\text{serv}}_{\text{cam}} + T^{\text{cam}}_{\text{serv}} T^{\text{serv}}_{\text{clw}} \left(-r^{\text{clw}}_{\text{serv}} + T^{\text{clw}}_{\text{client}} c_i^{\text{client}} \right).$$

Note that $T^{\text{frame2}}_{\text{frame1}} = \left(T^{\text{frame1}}_{\text{frame2}}\right)^{-1} = \left(T^{\text{frame1}}_{\text{frame2}}\right)^{T}$ for two coordinate frames frame 1 and frame 2. In summing up, the entire transformation can be expressed by

$$(r^{\text{cam}}_{\text{cam}\to c_1}, r^{\text{cam}}_{\text{cam}\to c_2}, r^{\text{cam}}_{\text{cam}\to c_3}, r^{\text{cam}}_{\text{cam}\to c_4}) = h_1(x), \tag{18}$$

where $h_1 : \mathbb{R}^{13} \to \mathbb{R}^{12}$ is a non-linear function. The non-linearity is caused by the transformation from quaternions to the rotation matrix $T^{\text{clw}}_{\text{client}}$, where products of $x_7, \ldots, x_{10}$ appear.

Now, the corner points in 3D-camera frame are transformed to 2D-pixel coordinates.[11] Table 2 gives a short description of the involved camera parameters. Given a point $p = (x, y, z)$ in the 3D

Table 2. Camera parameters

Parameter	Description	Unit
$f = (f_x, f_y)$	focal length	Pixels
$c_0 = (c_{0,x}, c_{0,y})$	center point/ principal point	Pixels
$k = (k_1, k_2, k_3)$	radial distortion coefficients	1
$t = (t_1, t_2)$	tangential distortion coefficients	1

camera frame with $z \neq 0$. We first define ideal (undistorted) image coordinates (x_u, y_u) by setting $x_u = f_x \frac{x}{z}$ and $y_u = f_y \frac{y}{z}$. We set $r := \sqrt{x_u^2 + y_u^2}$. The tangential distortion is given by

$$\begin{aligned} \Delta x &= 2t_1 x_u y_u &+t_2(r^2 + 2x_u^2), \\ \Delta y &= t_1(r^2 + 2y_u^2) &+2t_2 x_u y_u, \end{aligned}$$

and the radial distortion by a multiplicative factor

$$\lambda(k, r) = 1 + k_1 r^2 + k_2 r^4 + k_3 r^6.$$

The distorted coordinates are set to

$$\begin{aligned} x_d &= \lambda(k, r) x_u &+\Delta x, \\ y_d &= \lambda(k, r) y_u &+\Delta y. \end{aligned}$$

Finally, the 2D pixel coordinates are given by

$$\begin{aligned} c_x &= x_d &+c_{0,x}, \\ c_y &= y_d &+c_{0,y}. \end{aligned}$$

In summing up, the conversion from four 3D corner points to the measurement $Z = (c_{1x}, c_{1y}, \ldots, c_{4x}, c_{4y})$ can be written as

$$Z = h_2((r^{\text{cam}}_{\text{cam}\to c_1}, r^{\text{cam}}_{\text{cam}\to c_2}, r^{\text{cam}}_{\text{cam}\to c_3}, r^{\text{cam}}_{\text{cam}\to c_4})), \tag{19}$$

with $h_2 : \mathbb{R}^{12} \to \mathbb{R}^8$. The non-linearity is caused by the products in the tangential and radial distortion terms. Finally the measurement model function can be written as a composition of two non-linear functions $h = h_2 \circ h_1$.

Guidance and Control

The guidance system provides reference values to generate a position and attitude profile.[1] The objective of guidance is to define and force a state that the spacecraft should finally reach. Several guidance modes can be generated for different scenarios: fly-around, wait a hold points, continuous approach, departure.

The servicer's guidance function additionally concerns the client's rotational movement observed from the measurements in the navigation part of the GNC system. The servicer has to react to changes in the client's attitude to keep the desired relative orientation and position with respect to the tumbling target spacecraft, see Figure 6. In detail, let $p^{\text{guidance}}_{0,\text{servicer}}$ and $q^{\text{guidance}}_{0,\text{servicer}}$ be guidance values for the servicer which are outputs of some trajectory generation system and $q^{\text{filter}}_{\text{client}}$ be the filter estimate

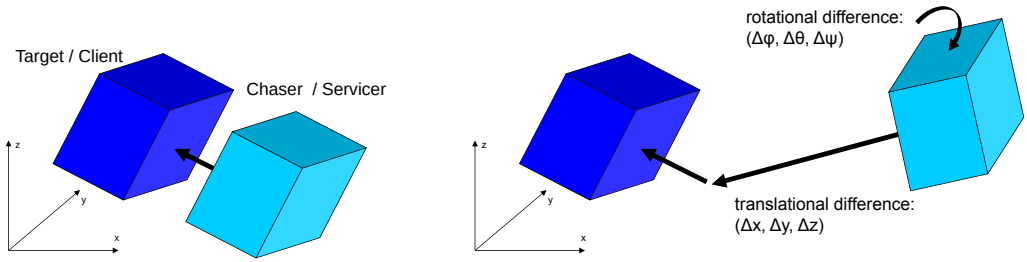

Figure 6. Left: desired relative state, right: current state and rotational and translational difference

of the client's attitude and $T^{\text{filter}}_{\text{client}}$ the corresponding attitude matrix. The final guidance values for the servicer are

$$p^{\text{guidance}}_{\text{servicer}} = T^{\text{filter}}_{\text{client}}\, p^{\text{guidance}}_{0,\text{servicer}}, \quad q^{\text{guidance}}_{\text{servicer}} = q^{\text{filter}}_{\text{client}}\, q^{\text{guidance}}_{0,\text{servicer}}.$$

In the first equation a matrix vector product appears, whereas in the second equation a quaternion multiplication appears.[10]

A PID-controller compares the reference values given by the guidance system with the actual values. An estimation of the actual position of the servicer is provided by the navigation filter. An estimation / measurement of the actual attitude is given by an accurate attitude sensor. The outputs of the controller are commands for the actuators (thrusters, reaction wheels). Let $e(t)$ be the position or attitude error, i.e. the difference between actual and guidance value. The controller output $u(t)$ is a weighted sum of $e(t)$, its time derivative and its integral:

$$u(t) = k_P\, e(t) + k_I \int_0^t e(\tau)\mathrm{d}\tau + k_D\, \frac{\mathrm{d}}{\mathrm{d}t} e(t),$$

where k_P, k_I, k_D are called proportional gain, integral gain and derivative gain and are tuned based on steady state performance requirements and the desired damping of the entire system.

HARDWARE-IN-THE-LOOP RENDEZVOUS SIMULATION

EPOS 2.0 Robotic Testbed for Rendezvous and Docking Simulation

German Aerospace Center (DLR) has experience in the field of simulating rendezvous and docking maneuvers for more than two decades. In the 1980s, DLR / GSOC (German Space Operations Center) developed the European Proximity Operations Simulator in its first version (EPOS 1.0) in cooperation with the European Space Agency. It was a robotic-based testbed where the simulation of rendezvous maneuvers over the last few critical meters prior to physical docking were performed. The platform could provide 6-DOF translational and rotational motion up to a distance of 12 m to the RVD interface. The last intensive utilization of the facility was the test and verification of the ATV RVD sensors and systems which are used for the approach to ISS. It was also used for testing RVD sensors of the Japanese HTV.

Future space applications like on-orbit servicing missions pose new challenges on the spacecraft's rendezvous and docking sensor systems and thus require the EPOS facility to provide advanced test and verification capabilities for complete RVD maneuvers of OOS missions.[3] This encloses

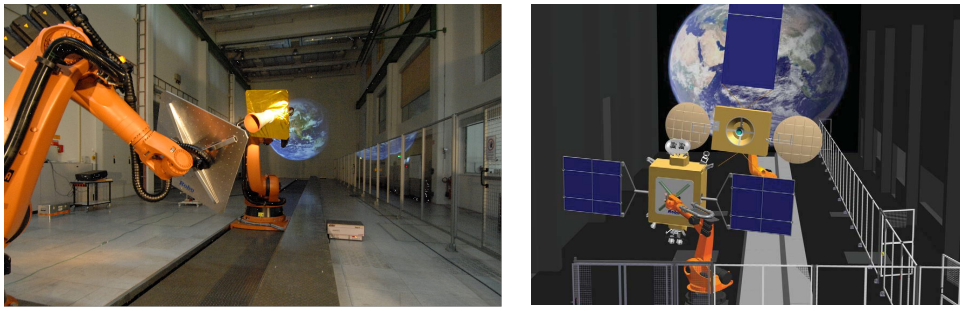

Figure 7. EPOS 2.0 - Testbed for real-time rendezvous and docking simulations

- 6-DOF relative dynamic motion of two satellites in the final approaching phase from 25 meters to 0 meters,
- 6-DOF contact dynamic behavior during the entire docking process including the initial impact, soft docking and hard docking,
- space-representative lighting and background conditions.

Since the old EPOS facility apparently could not provide all of these capabilities, it was replaced by a new EPOS system in 2008/2009. The facility comprises a hardware-in-the-loop simulator based on two industrial robots (of which one is mounted on a 25m rail system) for real-time simulations of rendezvous and docking maneuvers, see Figure 7. In detail, the test bed consists of

- a rail system mounted on the laboratory floor to move an industrial robot up to a distance of 25m,
- a KUKA KR100HA robot (robot 1) mounted on the rail system for simulating the 6 degrees of freedom of one spacecraft,
- a KUKA KR240 robot (robot 2) mounted at the end of the rail system for simulating the 6 degrees of freedom of the second spacecraft and
- a PC-based monitoring and control system to monitor and control the facility in real-time.

In a possible simulation scenario one robot simulates the client spacecraft and carries a typical mock-up of a geostationary satellite. The other robot simulates the service spacecraft and carries several sensors like cameras, force-torque sensor, etc.

The EPOS test bed allows simulation of the last critical phase (separation ranging from 25m to 0m) of the approach process. It is a highly accurate test bed, whereas the measurement and positioning performance has been increased by a factor 10 compared to the former EPOS facility.[3] Further, also contact dynamic simulations can be performed as opposed to EPOS 1.0. The dynamic capabilities of the new EPOS facility allow high commanding rates (250 Hz) which is a precondition for performing contact dynamic simulations. Finally, the simulation of sunlight illumination is performed by a 12kW spotlight such that the performance of an optical sensor under quite realistic conditions can be tested. Table 3 summarizes the motion capabilities of the facility.[3]

Table 3. EPOS motions capabilities

Parameter (Unit)	Robot 1	Robot 2
Position:		
X (m)	-2.5 to $+24.5$	-2.5 to $+2.5$
Y (m)	-2.5 to $+2.5$	-1.0 to $+4.0$
Z (m)	-0.5 to $+1.2$	-0.5 to $+1.5$
Attitude:		
Roll (deg)	-300 to $+300$	-300 to $+300$
Pitch (deg)	-90 to $+90$	-90 to $+90$
Yaw (deg)	-90 to $+90$	-90 to $+90$
Maximum velocity:		
Translational (m/s)	2	2
Rotational (deg/s)	180	180

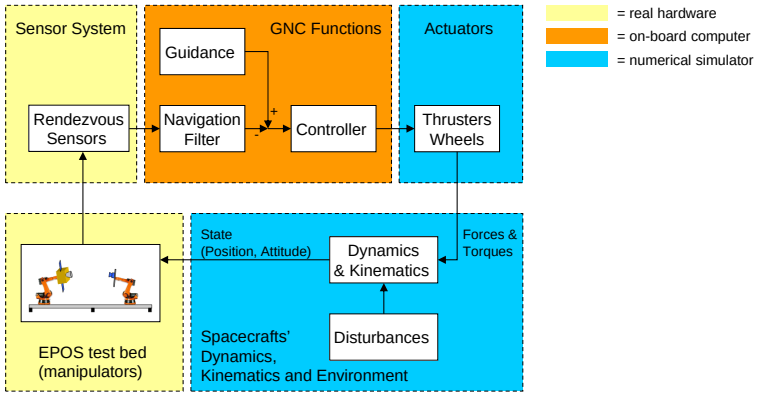

Figure 8. Rendezvous control loop

Overview on rendezvous HIL-simulation

Hardware-in-the-loop simulation is a very effective way to perform verification and testing of complex real-time embedded systems like rendezvous sensors. Inputs and outputs of an embedded system are connected to a correspondent counterpart - the so-called HIL-simulator - that simulates the real environment of the system.

A typical setting for rendezvous simulation is as follows, see also Fig. 8 which shows a typical control loop for a rendezvous: A rendezvous sensor for relative navigation measures relative position and attitude of the servicing satellite with respect to the target satellite. This measurement is processed by the GNC system and thruster commands are computed. Actuators like thrusters or reaction wheels cannot be simulated with real hardware and need to be replaced by actuator models. Similarly, mathematical models that describe the orbit and attitude dynamics as well as its environment are part of the numerical simulator. In detail, the position and attitude of the satellites are calculated by solving equations of motion for the satellites' orbit and attitude. In the next sample, the positions and attitudes are commanded to the facility and simulated by the manipulators of the EPOS facility. The manipulators can be regarded as connection of the numerical HIL-simulator with the embedded system, i.e. with the rendezvous sensor.

The GNC system, including navigation (image processing and navigation filters), guidance and

Figure 9. Prosilica GC-655 used as optical sensor in rendezvous simulations

control, has been presented in the previous section. In the following a short description of the sensor system, the numerical simulator and the development of real-time simulation software is given.

Sensor System As vision based sensor a Prosilica Gigabit Ethernet mono camera of type GC-655 (see Figure 9) is used. It is a charge coupled device (CCD) sensor with a high dynamic range to cover various lighting situations.

Numerical Simulator The objective is to develop a realistic simulation of the rendezvous process. Actuator models transform the thruster commands given by the controller to force and torque quantities by taking actuator limits (e.g. maximum thruster forces) into account. The spacecrafts' physical environment is realistically simulated by including the real orbit mechanics. Hill equations, Euler equation and quaternion differential equation are applied to model the spacecrafts' translational and rotational kinematics and dynamics (see Equations (13), (14) and (15)).

Development of Real-Time Simulation Software The software products MATLAB and MATLAB Simulink are used as devolopment environment for the rendezvous software. Simulink is a model-based simulation tool which is integrated in the Matlab environment and is widely used in control theory domain.

An interface to the camera (via Ethernet) has been established. All software components like image segmentation, navigation filters, control algorithms and the dynamic satellite simulator are integrated in the Simulink model. The model can be executed in multi-tasking mode. The satellites dynamics run with a frequency of 250 Hz which is the commanding frequency the facility requires. Thus, every 4ms positioning commands are sent to the facility and the robots move according to the prescribed values computed by the numerical simulator. Other components like image processing, navigation filter and controller are executed with a lower rate. Task with lower frequency are assigned with a lower priority. They can be preempted by tasks with higher priority such that the provision of positioning commands every 4ms is guaranteed even if the computation time of some other subsystems is greater than 4ms.

By making use of the Matlab Real-time Workshop C code for real-time execution of the model is generated automatically. The final executable runs on the real-time operational system VxWorks.

RESULTS

In the following results of hardware-in-the-loop tests at EPOS 2.0 are presented. To investigate the performance of the GNC system simulations with a constant guidance trajectory are performed that forces the service satellite to keep a given relative position and attitude w.r.t. the target. The main task is to show stability of the closed loop simulation. Table 4 gives an overview of the simulation parameters.

Table 4. Simulation Parameters

Parameter name	Value	Unit
Camera parameters:		
focal length	$[604, 604]$	pixels
pixel grid size	$9.9\,10^{-6}$	m
resolution	640×480	pixel
Satellite parameters:		
mass (servicer, client)	1000	kg
moment of inertia (servicer, client)	$(100, 200, 80)$	kg m^2
orbit rate (for LEO orbit)	0.001	rad s^{-1}
Filter parameters:		
standard deviation of process noise		
position, velocity	0.0001	m, m s^{-1}
attitude, attitude rate	0.25	deg, deg s^{-1}
standard deviation of measurement noise	2	pixels
Multi-tasking sample rates:		
satellite kinematics and dynamics	250	Hz
camera, image processing, filter corrector step	5	Hz
filter predictor step, controller	10	Hz

In a first experiment, an open loop simulation was performed for comparison of the Extended and the Unscented Kalman filter, see Figure 10 which shows the error of the filter estimates, i.e. the difference between filter value and real value. Open loop means that neither the EKF estimate nor the UKF estimate are fed back to the Rv control loop. The distance between servicer and target is approximately 20 m. Both filters show good results. The errors are in an acceptable range.

Considering the translational coordinates, the biggest error appears at the x-coordinate / V-Bar which points approximately in the same direction as the optical axes of the camera for small angles (attitude servicer in CLW, attitude camera w.r.t servicer). However the error is less than 1% of the distance. A change in V-Bar direction is hard to detect if the distance is big. The length of the rectangular target face is 2.30 m. At a distance of 20m this results in a length of 69.46 pixels in the image plane (using a focal length of 604 pixels). If the distance decreases to 19.80 m, the length in the image plane is 70.16 pixels. A difference in the distance of 0.2 m results in a difference of less than 1 pixel. Thus, no better results can be expected for the distance estimation. Similar, small changes in the pitch and yaw component are difficult to detect. That is why the error in this coordinates are relative big (up to 10 deg).

Throughout all translational and rotational components the Unscented Kalman filter provided better estimates than the Extended Kalman filter. Errors are less noisy and the absolute error is smaller. Consequently, we applied the Unscented Kalman filter for closed loop rendezvous simulations:

Figure 11 and 12 shows the error of the UKF estimates, the controller performance and the total performance of the system during a closed loop simulation at approximately 5m and 20m distance between servicer and target. The performance of the controller is the difference between guidance and real values. The performance of the entire system is the difference between reference and real values. In the guidance subsystem we make use of the filter estimate of the client's attitude. The reference state is calculated the same way replacing the filter estimate by the real value of the client's attitude.

Note that, in a closed loop simulation an error in the client's attitude influences the performance

of the position of the servicer. Errors in the pitch and yaw component result in errors in the guidance values for the servicer (y- and z-component) which leads to a worse total performance. This is why we have a relative high performance error (see subfigure bottom, left in Figure 11 and 12) in the y- and z-coordinate, although the pure filter error (difference between filter and real value) is small (compare subfigure top, left).

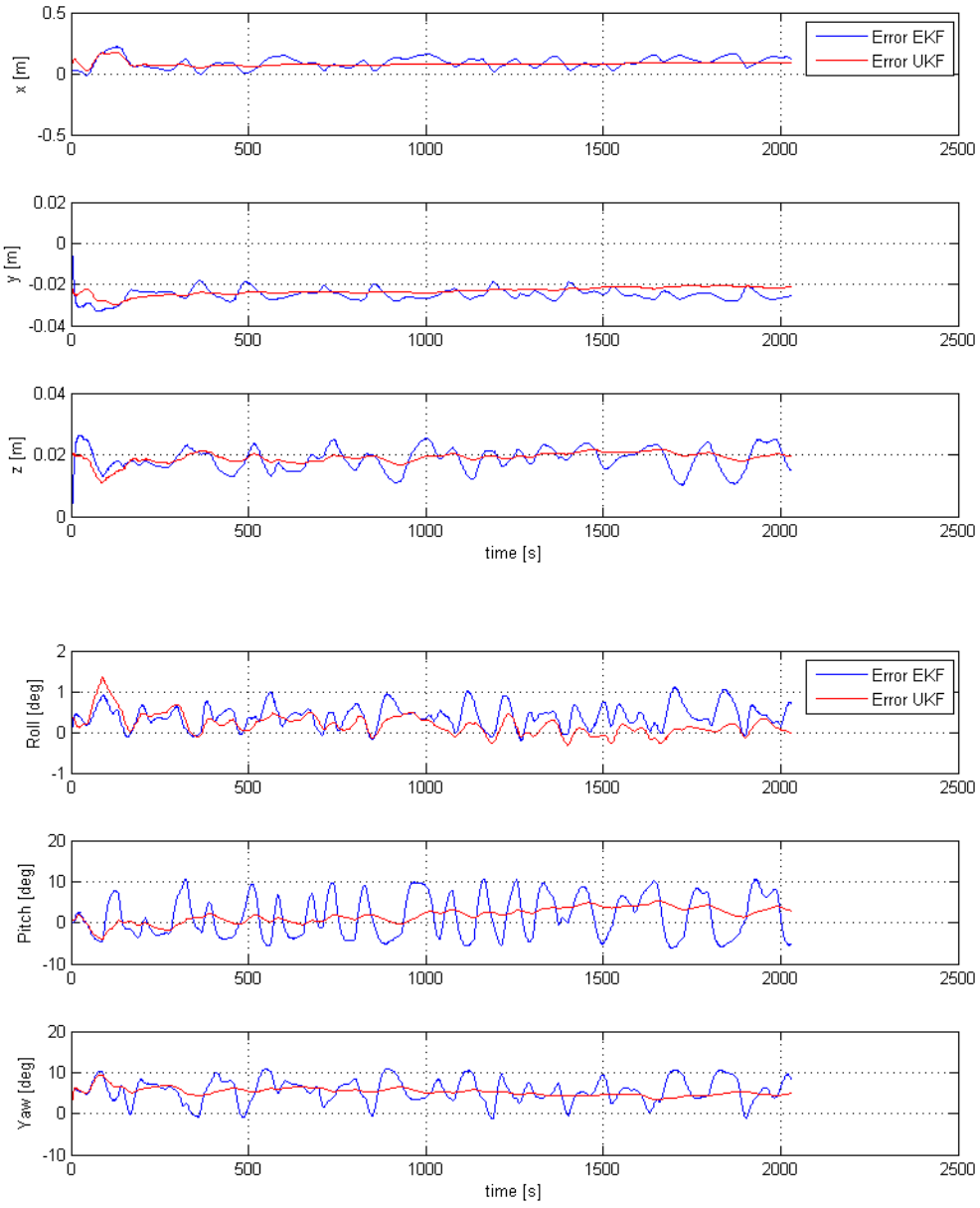

Figure 10. Simulation at 20m Distance, Top: Position Error of EKF and UKF, Bottom: Attitude Error of EKF and UKF

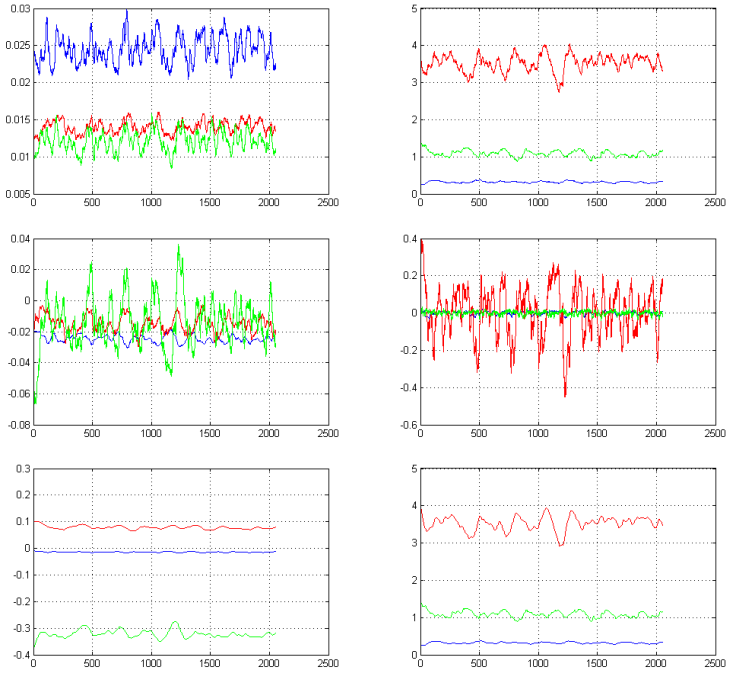

Figure 11. Simulation at 5m Distance, Top: Filter Error, Middle: Controller Performance, Bottom: Total Performance, Left: Position [m] (blue: x, red: y, green: z), Right: Attitude [deg] (blue: roll, red: pitch, green: yaw), abscissa: time [s]

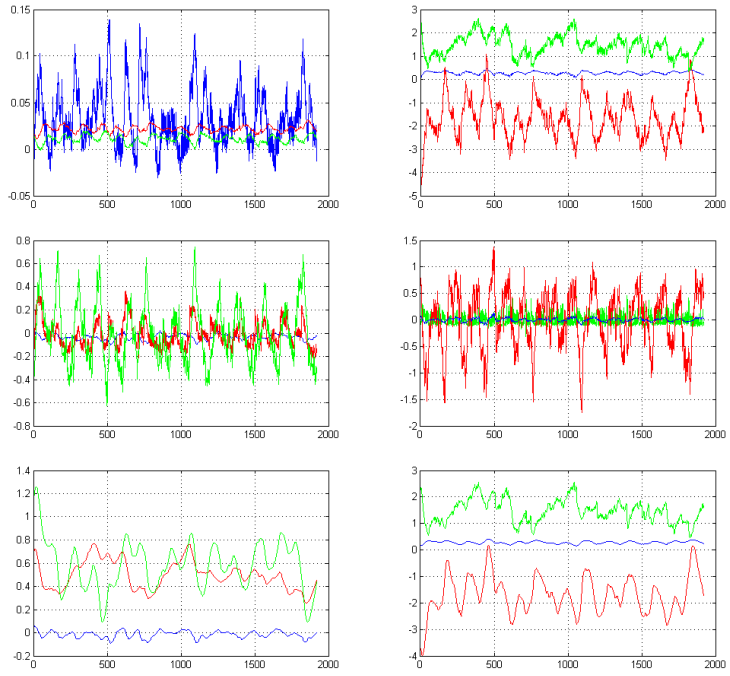

Figure 12. Simulation at 20m Distance, Top: Filter Error, Middle: Controller Performance, Bottom: Total Performance, Left: Position [m] (blue: x, red: y, green: z), Right: Attitude [deg] (blue: roll, red: pitch, green: yaw), abscissa: time [s]

CONCLUSION AND OUTLOOK

This paper described the development of a guidance, navigation and control system for autonomous rendezvous processes and the testing of the GNC system in the framework of a hardware-in-the-loop simulation at DLR's RvD simulation facility EPOS 2.0. As pointed out in the introduction, the GNC system has to cope with an uncooperative target without any attitude control. Further the target detection, navigation and control of the rendezvous process should be performed in an autonomous way.

The proposed solution mainly meets these requirements. The target is completely passive and the approach is purely based on the camera data evaluation. To keep a prescribed relative position and orientation with respect to the target, the servicer's guidance is adapted such that the service satellite can react to sudden rotational movements of the client like tumbling. The autonomy constraint is mainly fulfilled. Our system needs an initial guess for the image segmentation which could be provided by other sensors used in previous rendezvous phases. If there is no such information available our tracker cannot be initialized. In the simulations done at EPOS we always provided the image processing system with an initial value for the position of the target in the camera image (pixel coordinates of the four corner points). Depending on the length of the scanlines, the initial guess has to be quite accurate.

The development of target acquisition and/or improved image segmentations methods which need no initial knowledge of the edges to be detected are currently under development. Furthermore, tests under different illumination conditions are planned to test the robustness of the vision-based navigation system.

REFERENCES

[1] W. Fehse, *Automated Rendezvous and Docking of Spacecraft*. Washington, D.C: Cambridge Aerospace Series, 2003.

[2] T. Rupp, T. Boge, R. Kiehling, and F. Sellmaier, "Flight Dynamics Challenges of the German On-Orbit Servicing Mission DEOS," *21st International Symposium on Space Flight Dynamics*, Toulouse, France, 2009.

[3] T. Boge, T. Rupp, K. Landzettel, T. Wimmer, C. Mietner, J. Bosse, and B. Thaler, "Hardware in the Loop Simulator for Rendezvous and Docking Maneuvers (Hardware in the Loop Simulator von Rendezvous und Docking Manoevern)," *German Aerospace Congress of DGLR*, Aachen, Germany, 2009.

[4] T. Tzschichholz, T. Boge, and H. Benninghoff, "A Flexible Image Processing Framework for Vision-Based Navigation Using Monocular Imaging Sensors," *8th International ESA Conference on Guidance, Navigation & Control Systems*, Karlovy Vary, Czech Republic, 2011.

[5] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, Vol. 24, No. 6, 1981.

[6] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transaction of the ASME, Journal of Basic Engineering*, Vol. 83, 1960, pp. 35–45.

[7] P. Zarchan and H. Musoff, *Fundamental of Kalman Filtering: A Practical Approach*, Vol. 190. Cambridge, Massachusettts: Progress in Astronautics and Aeronautics, 2000.

[8] S. Julier and J. Uhlmann, "A New Approach for Filtering Nonlinear Systems," *Proceedings of 1995 American Control Conference*, Vol. 3, 1995.

[9] S. Julier and J. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," *Proceedings of 1995 American Control Conference*, Vol. 3, 1997.

[10] J. R. Wertz, *Attitude Determination and Control*. Dordrecht, Boston, London: Kluwer Academic Publishers, 2002.

[11] J. Heikkila and O. Silven, "A Four-step Camera Calibration Procedure with Implicit Image Correction," *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, Washington, DC, USA, IEEE Computer Society, 1997.