AUTONOMOUS RENDEZVOUS WITH AN UNCERTAIN, UNCOOPERATIVE TUMBLING TARGET: THE TUMBLEDOCK FLIGHT EXPERIMENTS

Keenan Albee[1], **Caroline Specht**[2], **Hrishik Mishra**[2], **Charles Oestreich**[1], **Bernhard Brunner**[2], **Roberto Lampariello**[2], **and Richard Linares**[1]

Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA; {albee, linaresr} at mit.edu

German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Wessling, Germany; {firstname.lastname} at dlr.de

ABSTRACT

As the amount of orbital debris grows so too does the need for on-orbit repair and deorbit solutions to avoid cascading Kessler syndrome. While a number of options have been proposed for capturing defunct satellites and other high-value debris, methods for performing close-proximity rendezvous with these objects are also necessary. However, a significant portion of these objects are tumbling with unknown angular orientation and rate; the rendezvous procedure for these tumbling objects is complex and must be performed in real-time, precluding human teleoperation or offline, on-the-ground solutions. Therefore, autonomous rendezvous for these tumbling targets is highly desirable. A fully autonomous rendezvous pipeline was recently proposed by the authors, and has been extended in this paper to a working demonstration in microgravity on resource-constrained hardware. Utilizing factor graph-based SLAM to identify a target object's rotation, nonlinear programming-based motion planning, and robust control for safe online-updateable reference trajectory tracking, this work overviews the TRACE (**T**umbling **R**endezvous via **A**utonomous **C**haracterization and **E**xecution) algorithmic pipeline in its entirety. The pipeline is shown in practice on a sample rendezvous case with a tri-axially tumbling target. The centerpiece of this work is on-orbit results following a two-year hardware implementation and microgravity testing campaign using NASA's Astrobee robots. A number of implementation considerations are discussed, including augmentations to the Astrobees' localization system. This work represents the first autonomous on-orbit rendezvous with an uncharacterized, uncooperative tumbling target, to the authors' knowledge.

Key words: on-orbit servicing, active debris removal, space robotics, SLAM, model predictive control, trajectory optimization, state estimation.

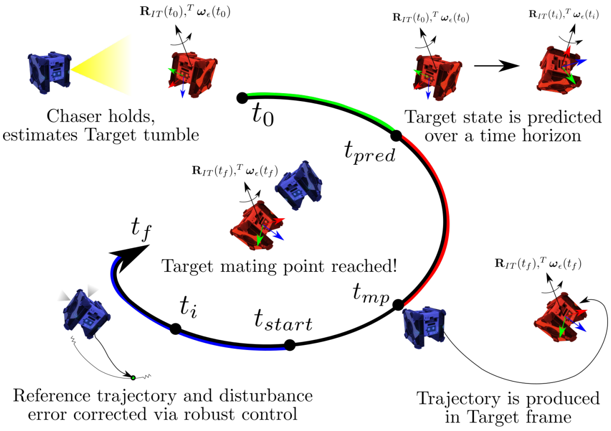
$\mathbf{R}_{IT}(t_0), {}^T\boldsymbol{\omega}_\epsilon(t_0)$
Chaser holds,
estimates Target tumble
t_0
$\mathbf{R}_{IT}(t_0), {}^T\boldsymbol{\omega}_\epsilon(t_0)$
$\mathbf{R}_{IT}(t_i), {}^T\boldsymbol{\omega}_\epsilon(t_i)$
Target state is predicted
over a time horizon
t_{pred}
$\mathbf{R}_{IT}(t_f), {}^T\boldsymbol{\omega}_\epsilon(t_f)$
t_f
Target mating point reached!
t_{mp}
$\mathbf{R}_{IT}(t_f), {}^T\boldsymbol{\omega}_\epsilon(t_f)$
t_i
t_{start}
Reference trajectory and disturbance
error corrected via robust control
Trajectory is produced
in Target frame

Figure 1: The TRACE pipeline, consisting of a SLAM observation period (green), motion planning period (red), and a robust tracking period (blue). The pipeline is run in sequence, with outputs from previous stages feeding subsequent stages of the rendezvous procedure.

1. INTRODUCTION

Autonomous close proximity rendezvous is desirable to repair or refuel damaged satellites, or to capture and safely deorbit debris. A number of authors have proposed solutions to rendezvous with a static object [1] [2], and even demonstrated a variety of methods to deorbit debris using a passive system like a net or harpoon [3]. A significant fraction of this debris is in free tumble [4], a more challenging case than non-rotating or simple rotation (i.e., flat spin). The unintuitive nature of the Newton-Euler dynamics precludes direct human teleoperation and significant uncertainty in how individual objects are tumbling demands real-time onboard identification of target properties [5][6]. A number of techniques propose deterministic motion planning solutions assuming knowledge of the Target's tumble [7][8] [9] [10] [11] [12]. However, rendezvous with an uncooperative, uncertain, tumbling target was previously undemonstrated; what's more, a complete autonomous estimation, planning, and control pipeline had not been previously shown

This work was completed under NASA Space Technology Research Fellowship support, grant number 80NSSC17K0077. This work was also supported by the Institute of Robotics and Mechatronics, DLR. The authors gratefully acknowledge these sponsors.

until [13]. This work introduces the expanded TRACE pipeline for autonomous robotic rendezvous with an uncertain, tumbling target and showcases the pipeline's on-orbit results from the International Space Station during an on-orbit test campaign. Practical implementation details are considered alongside the presentation of selected results from on-orbit experiments.

2. ASTROBEE AND THE AUTONOMOUS RENDEZVOUS PROBLEM

This work focuses on initial results of an experimental campaign for autonomous rendezvous using two Astrobee robots on the International Space Station (ISS), conducted under the ROAM/TumbleDock test campaign, a collaboration between MIT, DLR, and NASA. In this series of experiments one robot serves as the autonomously controlled "Chaser," and another as the unknown "Target". The Target mimics the anticipated tumble of the Envisat satellite, a high-priority target for active debris removal.

NASA's Astrobee robots are a set of free-flying robots which operate aboard the ISS. The Astrobees enable microgravity autonomy research through a suite of sensors and three reconfigurable general-purpose processors. The Astrobees utilize impellers to provide full holonomic propulsion within the Japanese Experiment Module (JEM), with multiple sensors for navigation including cameras and an inertial measurement unit (IMU). The flight software is implemented on two general-purpose Snapdragon-based processors, which utilize Ubuntu 16.04 and ROS [14].

The autonomous rendezvous problem considers a close proximity rendezvous maneuver between two of the Astrobees, analogous to the last $\sim 20-40$ [m] of an on-orbit approach operation, with the goal of safely reaching a predefined offset point called the mating point (MP), fixed in the tumbling Target's body frame. To complete this analogue, an artificial hull similar to the shape of Envisat is superimposed on the Target in the motion planning computation, Fig. 2. From the point of view of the Chaser spacecraft the Target is non-cooperative, passive, and uncharacterized. A strategy is required to perform ingress to the uncertain tumbling Target and reach a predefined offset mating point.

The Chaser spacecraft which will perform the rendezvous begins at some known initial standoff distance, in this case 1.5 [m] along the ISS y-axis, and is equipped to perform visual estimation of the Target. Kinematic and velocity state constraints $\mathcal{X}$ on the Chaser motion are determined by the tumbling Target's artificial collision volume, and operational limits of the Chaser. The Chaser also has input constraints $\mathcal{U}$. The Chaser has approximate knowledge of the Target's inertial parameters, but not its rotational state. Finally, unstructured uncertainty on the Chaser motion may exist, i.e., additive uncertainties lying in a convex set $\mathbf{w} \in \mathcal{W}$.

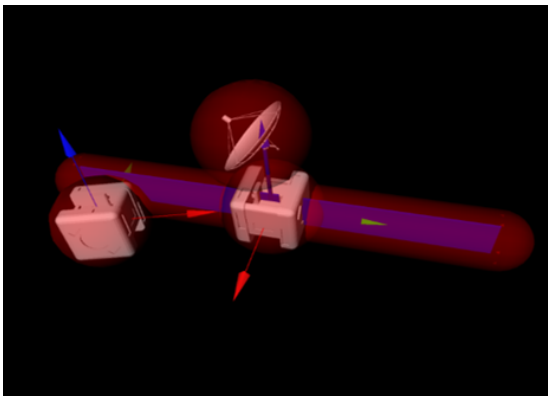

Figure 2: A visualization of the simulated virtual obstacle set superimposed on the Astrobee robots for motion planning purposes. The Target robot has the additional collision geometry.

Three main tasks must be successfully completed to accomplish the close proximity tumbling target rendezvous task:

1. Identify the Target's tumble, consisting of its attitude and angular velocity ($\mathbf{R}_{IT}$ and ${}^{T}\boldsymbol{\omega}_{T}$) and, optionally, its inertia tensor ($\mathbf{I}$).

2. Create a collision-avoiding, fuel-conserving, constraint-aware motion plan, $\mathcal{P}$.

3. Perform precise control, ideally with robustness guarantees against prevailing uncertainty levels, $\mathcal{W}$.

The interested reader may consult [13] for a full problem formulation.

3. METHODS

The three tasks satisfying the tumbling rendezvous task are accomplished by the TRACE pipeline, briefly outlined in Fig. 1. First, an estimation phase utilizing point cloud analysis of the tumbling Target feeds a factor graph-based approach which results in an estimate of the Target's rotational state, $\mathbf{T}_{att} := \{\mathbf{R}_{IT}, {}^{T}\boldsymbol{\omega}_{T}\}$. SLAM computations are additionally available online to provide real-time updates of $\mathbf{R}_{IT}$. Based on predicted $\hat{\mathbf{T}}_{att}(t)$, a nonlinear programming-based trajectory optimization approach computes a motion plan $\mathcal{P} := \{\mathbf{x}_k, \mathbf{u}_k\}$ with respect to a model-based propagation of the Target's rotational motion, $\hat{\mathbf{R}}_{IT}(t)$. Finally, an uncertainty determination module analyzes possible deviations in $\mathcal{P}$ if it is updated online, informing a robust controller that provides general purpose robustness against an estimated uncertainty bound, $\mathcal{W}$. Finally, the mating point is reached and the mission concludes. TRACE executes the above in sequence, proceeding clockwise in Fig. 1, paying close attention to hardware timing constraints of each component, noted as t_* in Table 1. These components are briefly summarized here; the reader is referred to [13] for a more detailed discussion.

Table 1: Approximate timing information for TRACE's execution.

Time Marker	Elapsed Time $[s]$	Duration $[s]$
Observation start (t_0)	0	-
Prediction complete (t_{pred})	120	120 (fixed)
Motion planning complete (t_{mp})	$\sim 130 - 180$	$\sim 10 - 60$ (safe upper bound)
Rendezvous start (t_{start})	180 (safe upper bound)	-
Rendezvous complete (t_f)	$\sim 210 - 270$	$\sim 30 - 90$

3.1. Visual Estimation

Simultaneous Localization and Mapping (SLAM) is the two-fold problem of determining a robot's pose, $\mathbf{T}_{IB} \in SE(3)$ while determining its surrounding environment. A SLAM problem emerges for the tumbling Target estimation case where the Chaser must determine its navigational state $\mathbf{x}_{IC} \triangleq \{\mathbf{T}_{IC}, \mathbf{v}_C, \boldsymbol{\omega}_C\}$, and the Target's attitude state $\mathbf{x}_{IT} \triangleq \{\mathbf{R}_{IT}, \boldsymbol{\omega}_T\}$, while disambiguating relative motion.

However, measurements obtained by the Chaser are never perfect, and so the values above are not known precisely. Moreover, the Target's initial angular state is entirely unknown. Determining the quantities above under uncertain measurements then becomes a nonlinear optimization with the goal of minimizing probabilistic fit error. A line of research [15][16][17] solves the on-orbit inspection problem by applying various SLAM techniques, ultimately settling on a real-time factor graph-based approach. The main variables become the Chaser's state history $\mathbf{x}_{IC}$, the Target's pose history $\mathcal{F}_G$, the Chaser's pose $\mathbf{T}_{GC}$ w.r.t. the arbitrary Target frame $\mathcal{F}_G$, the Target's principal axis frame offset from an initial guess of the frame $\mathbf{r}_{GT}$, and the Target's principal axis frame offset w.r.t. the inertial frame, $\mathbf{r}_{IT}$.

A number of factors introduced by Oestreich in [18] relate the variables, forming a graph structure of probabilistic constraints between variables. Front-end computation to perform pose estimation using point cloud keyframes, $\mathcal{K}$, obtained from a time of flight camera are accomplished using Teaser++ [19]. Angular velocity is approximated using successive estimates of the Target's pose followed by a conversion to the Target's body frame. $\mathbf{R}_{GT}$ is additionally determined, post-factor graph solution, using Setterfield's polhode analysis procedure [16]—this is the final key to creating a Target frame angular velocity estimate.

Finally, the above procedure results in a model of the Target's motion,

$$\mathcal{M}_T := \{J_1, J_2, \mathbf{R}_{IT}, {}^T\boldsymbol{\omega}_T\} \tag{1}$$

which can be employed for motion prediction. $J_1 \triangleq \frac{I_{xx}}{I_{zz}}$ and $J_2 \triangleq \frac{I_{yy}}{I_{zz}}$ are inertia ratios, defined from principal axis moments of inertia. Note that these values are estimated with accompanying statistics on their uncertainty, which are useful in feeding the robustness portions of the pipeline.

3.2. Motion Planning

The global motion planner's role is to produce a plan $\mathcal{P}$ appropriate for the nominal system model $\mathcal{M}_T$ produced by the visual estimation module. This is accomplished using a nonlinear optimization-based approach, where the mechanical energy is minimized as a function of the free parameters of three B-splines—one for each translational degree of freedom—and subject to position, velocity, actuation, plume impingement, and collision constraints. A warm start is provided to the nonlinear optimization problem through an offline-generated look-up table (LUT). The LUT is parameterized by initial Target state values, against which the estimated state of the Target is compared to determine which LUT entry should be used for the initial guess of the optimization parameters.

The generated motion plan $\mathcal{P}$ is defined and invariant in the Target's body frame for the purposes of efficient collision avoidance under trajectory tracking. To maintain feasibility, the plan as viewed in the Inertial frame ${}^I\mathcal{P}$ must preserve the trajectory as viewed in Target body frame, ${}^B\mathcal{P}$.

3.3. Robust Control

To enable control robustness guarantees, an uncertainty bound $\mathbf{w} \in \mathcal{W} \subset \mathbb{R}^n$ for the Chaser's state dynamics must be defined. One uncertainty source arises from the Target's estimated nominal model, $\mathcal{M}_T$, which affects the Chaser through online updates of the reference trajectory. In the tracking control problem this means the reference trajectory ${}^I\mathcal{P}$ may be adjusted from online visual estimation updates. In addition, general purpose uncertainty (noise) can be added into this uncertainty bound to account for other dominating uncertainty sources—in practice, Astrobee's localization system is the dominant uncertainty source.

Finally, the uncertainty bound $\mathcal{W}$, the inertial frame reference trajectory ${}^I\mathcal{P}$, and estimates of the Chaser's true state $\hat{\mathbf{x}}$ are fed to a robust tube MPC control strategy to track the reference trajectory with a tube robustness guarantee. This tube robustness guarantee holds as long as $\mathcal{W}$ is not too large. This ensures that nominal MPC inputs, $\mathbf{u}_{mpc}$, can be sufficiently tightened to counter disturbances with ancillary disturbance rejection controller input commands, $\mathbf{u}_{dr}$.

3.4. State Estimation for Control

To control the dynamics of each Astrobee robot for the considered rendezvous scenario, the local controller requires low-noise and smooth estimation of its motion state, i.e., $\mathbf{x}_{IC} \triangleq \{\mathbf{T}_{IC}, \mathbf{v}_C, \boldsymbol{\omega}_C\}$ for the Chaser, and $\mathbf{x}_{IT} \triangleq \{\mathbf{T}_{IT}, \mathbf{v}_T, \boldsymbol{\omega}_T\}$ for the Target. To this end, a two-layer approach was used for the state estimation to augment the default Localization Pipeline, shown in Fig. 3.

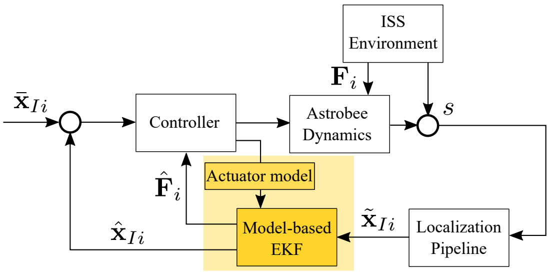

Figure 3: Controller block diagram for the i^{th} Astrobee, $i = C$ for the Chaser and $i = T$ for the Target. Extensions to the default Localization Pipeline are indicated in yellow.

First, the state ($\tilde{\mathbf{x}}_{Ii}$) is roughly estimated by the default localization pipeline using a kinematic sensor fusion of exteroceptive (cameras) and proprioceptive (IMU) measurements [13, §3.2]. However, the transitioning of optical features (s) from the field of view might result in estimation discontinuities, which negatively affects control performance. Furthermore, air circulation in the ISS can cause a disturbance wrench ($\mathbf{F}_i$) that perturbs the desired Astrobee motion. Hence, and secondly, a model-based EKF (indicated by the yellow blocks in Fig. 3) was added to provide not only a smooth and precise estimate of the motion state ($\hat{\mathbf{x}}_{Ii}$), but also an estimate of the disturbance wrenches ($\hat{\mathbf{F}}_i$). The EKF uses the governing Newton-Euler equations to predict $\hat{\mathbf{x}}_{Ii}$, while exploiting $\tilde{\mathbf{x}}_{Ii}$ as measurements.

To this end, the geometric EKF in [20, §IV] was extended in two ways. Firstly, the process model was augmented with a constant disturbance model, $\dot{\mathbf{F}}_i = 0$. Secondly, the measurement model was extended to include velocity feedback ($\tilde{\mathbf{v}}_i, \tilde{\boldsymbol{\omega}}_i$).

4. RESULTS AND PRACTICAL CONSIDERATIONS

The TRACE pipeline was successfully run on-orbit in a series of experiments showcasing multiple rendezvous with a tumbling Astrobee robot, as in Fig. 4. SLAM, motion planning, and robust control/uncertainty determination modules were run in real-time on the Astrobee hardware to determine the tumbling Target's motion, create a safe motion plan, and track the plan despite preset uncertainty levels. Each of the modules' on-orbit outcomes are now outlined, and the unique challenges of moving to a hardware implementation and how they were overcome are considered. TRACE's results of the first autonomous on-orbit rendezvous with an unknown tumbling target are shown.

4.1. Practical Considerations

Several practical considerations emerged from integration and hardware testing of the TRACE pipeline on the Astrobee robots. Some of the most important observations are briefly discussed before elaborating on experimental results.

The Limits of Onboard Localization On-orbit testing faced a significant challenge with Astrobee's default localization module, which was prone to infeasible jumps and general high root mean square error (RMSE), well beyond the typical uncertainty set $\mathcal{W}$ computed to account for online tumbling Target updates. Ultimately, the two-layer state estimation approach to the localization presented above was developed to overcome these discontinuities. This is useful practical experience for implementation of TRACE: real uncertainty sources might often differ from those expected, and $\mathcal{W}_0$—the base uncertainty set before accounting for additional uncertainties—should be designed with buffer in mind to account for these "unknown unknowns." However, as noted in Section 3.3, sometimes it is impossible to provide sufficient robustness; in these cases, the only options are to proceed with lower robustness guarantees, reduce uncertainty, or relax constraints.

Software Integration and Development for an Evolving Platform Some major practical takeaways from the development and hardware implementation of TRACE include the need for early standardization and the practical difficulties of moving to hardware. Hardware implementation is vital, giving a direct look at the actual sensors, noise, computational power, and environments that will be seen by the algorithms developed for autonomous systems. However, hardware implementation leads to many complications, particularly in moving from desktop-based computational tools to embedded programming that may be lacking important libraries or computational power, for example. Additionally, because of the number of algorithmic components, standardization and message-passing procedures must be settled early in development; luckily, ROS takes care of some of this complexity on the Astrobee platform. Some practical lessons learned are further documented in [21].

Experimental Environment Two considerations of the environment in which the experiment occurs must be noted. Almost immediately in early hardware tests, it became apparent that Astrobee's impellers produce a suction effect which can pull the robots forcefully towards the exterior walls of the test space. An additional minimum distance from the published Japanese Experiment Module (JEM) dimensions was therefore accounted for in the definition of the position constraints. Secondly, the JEM is in active and daily use by the crew aboard ISS.

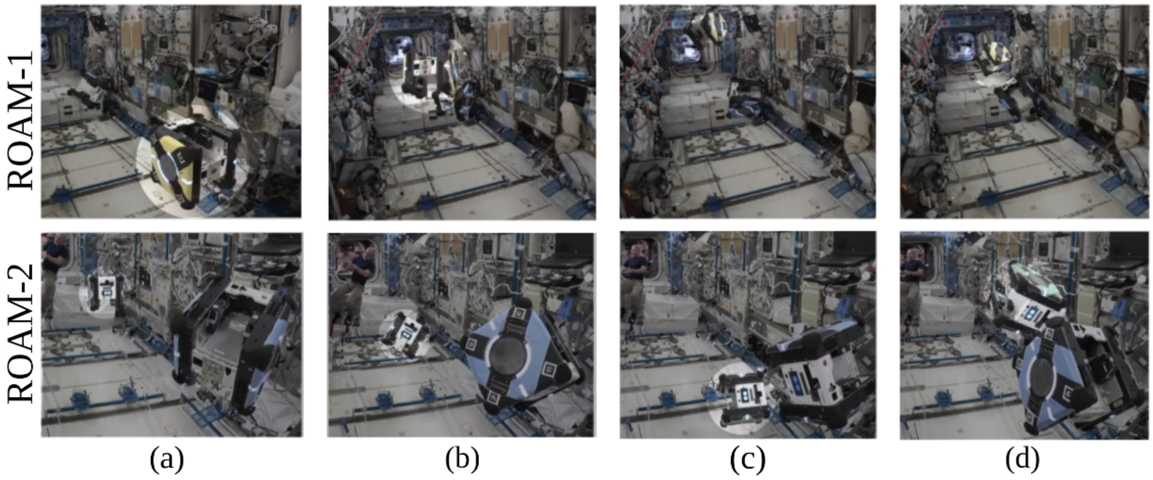

Figure 4: Successful rendezvous maneuvers from the ROAM-1 and ROAM-2 on-orbit tests. The Chaser, highlighted, approaches the Target from (a) - (d) after estimating the Target's attitude state, ultimately reaching the offset mating point. Note that Target and Chaser configuration is reversed between each test.

Sometimes cargo and other objects may be moved, resulting in environmentally-influenced constraint changes.

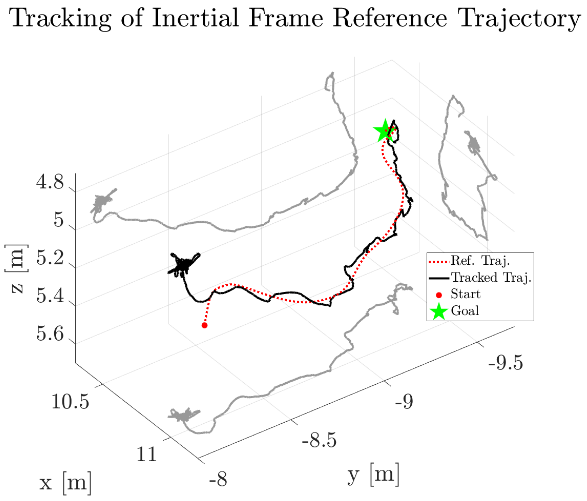

Figure 5: On-orbit data of a successful rendezvous maneuver, which was tracked using robust tube MPC.

4.2. The ROAM/TumbleDock Flight Experiments: On-Orbit Results

A number of experiments were performed on-orbit, evaluating different portions of the algorithmic pipeline; full pipeline results were obtained on a subset of tests, following the procedure of Fig. 1. A single test is highlighted here, which produced the motion plan $\mathcal{P}$ shown in Fig. 5, with the actual tracked plan shown against the inertial frame reference trajectory. Despite localization troubles, multiple successful rendezvous maneuvers were achieved. On-orbit results demonstrated that TRACE's modules can be run in real-time on resource constrained hardware.

Visual Estimation Visual estimation of the Target's tumble using the procedure outlined in Section 3.1 was performed for the period t_{pred}, after which principal axes were estimated resulting in the finalized estimate of the Target's rotational state. Attitude estimation results produced using the procedure for a sample experimental run are shown in Fig. 6. The estimated Target model, $\mathcal{M}_T$, and its associated state covariance $\mathbf{\Sigma}_T$ were then made available to the motion planning and uncertainty determination modules.

Computational times for on-orbit results are shown in Fig. 7. The SLAM module maintains a projected 0.5 [Hz] rate on hardware, with a majority of computational effort devoted to feature identification and matching of Target features.

Motion Planner The motion planner was able to produce a plan every time it was called on-orbit, in a mean time of 9.16 [s]. The trajectories produced are highly contingent on the values provided to the planner by the visual estimation step. As such, the initial positions of the Target and Chaser were not always as expected by the motion planner. The motion planner has been shown to tolerate a Chaser initial position up to 60 [cm] displaced from the expected position.

The trajectory produced for the test highlighted in this section can be viewed in Figs. 5 and 8, indicated by the dotted curve in each plot. This trajectory was planned with information indicating that the Chaser was located at $[0.25,\ 0.08,\ 0.15]$ [m] relative to the centroid of the JEM, or about 0.3 [m] from its expected position, and was generated in 8.45 [s].

Robust Tube MPC Fig. 8 shows the robust tube MPC tracking and ancillary/nominal controller activations for the successful rendezvous in the lower figure of Fig. 4 and also shown in Fig. 5. Disturbance rejection control worked as expected, tackling general disturbances from

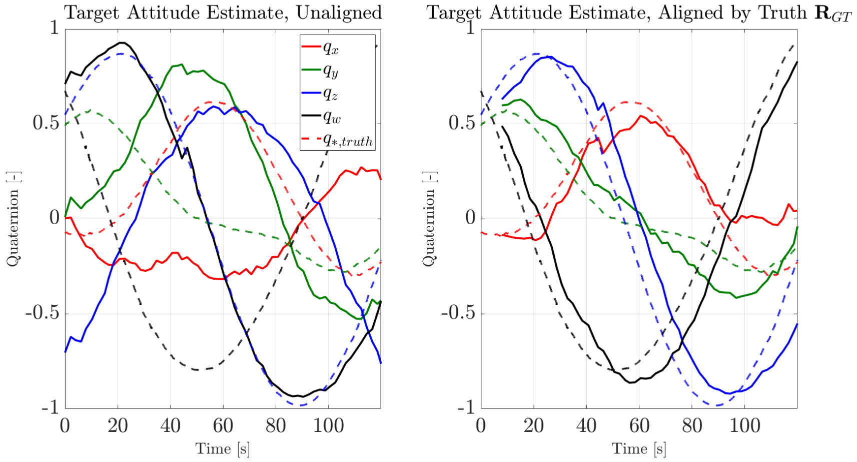

Figure 6: Attitude estimates produced online by the visual estimation component of the pipeline up to time t_{pred}. The values at right show attitude estimate quality after alignment with the true Target principal axes. "Truth" values shown are the Target's own attitude estimate.

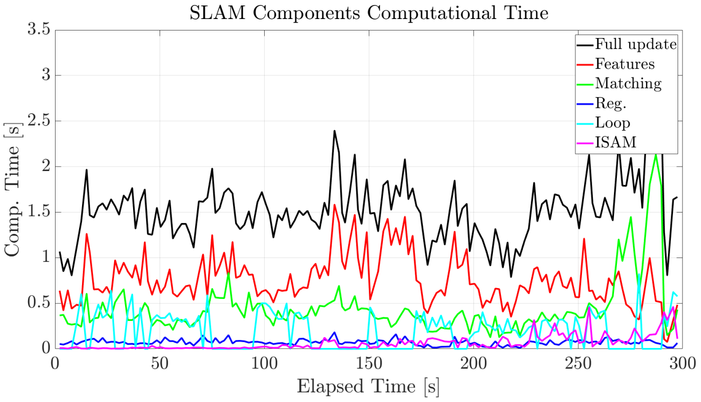

Figure 7: Timing information of various components of the SLAM system. Overall updates maintain a 0.5 [Hz] update rate on hardware.

the reference trajectory while nominal MPC provided longer-horizon guidance. For on-orbit results tube robustness still provides a general purpose robustness guarantee against set uncertainty levels, a desirable feature. Robust translational tracking was achieved despite disturbance sources such as the localization system mentioned in Section 4.1.

Model-Based EKF The experimental results for the model-based EKF on the Target Astrobee are provided in Figs. 9-10; the EKF was also used on a subset of runs for Chaser localization improvement. The position and velocity estimates provided by the localization pipeline (blue) and the EKF (red) from Fig. 3 are shown in Fig. 9. It can be seen that the localization estimates tend to suffer sporadic discontinuities (dashed lines). The model-based EKF is not affected by these discontinuities as they are not in agreement with the Astrobee model. Note that both position and velocity estimates are required for the motion stabilization of the Astrobee, and the removal of discontinuities was a major benefit. To validate the estimation of disturbance forces, $f_{T\bullet}$, the Astrobee actuators were turned off so that the commanded forces of the controller serve as a disturbance. In Fig. 10, the actual (red) and the estimated (blue) disturbance forces are shown, which demonstrates the estimation convergence.

5. CONCLUSION

The on-orbit demonstration of TRACE is a significant step toward autonomous rendezvous with tumbling targets, uniting multiple key algorithmic components of the autonomy pipeline. Without *a priori* target tumble knowledge, TRACE can determine how to best reach a safe offset mating point, from which docking procedures could be initiated. Robustness and constraint satisfaction despite significant uncertainties are also incorporated into the pipeline logic.

The estimation, motion planning, uncertainty propagation, and robust control components of the TRACE pipeline have been discussed and their application in multiple on-orbit demonstrations were shown. This is, to the authors' knowledge, the first on-orbit demonstration of autonomous rendezvous with an uncertain tumbling tar-

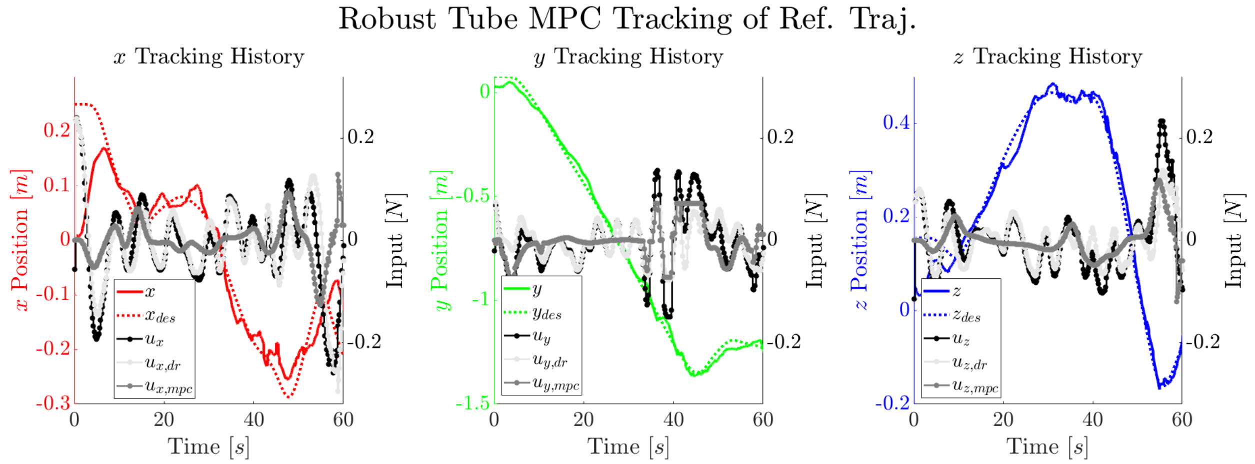

Figure 8: Robust tube MPC tracking of a successful rendezvous maneuver trajectory during ROAM-2. Note that u_* is the sum of constraint-tightened nominal MPC ($u_{*,mpc}$) and disturbance rejection control ($u_{*,dr}$), defined in Sec. 3.3.

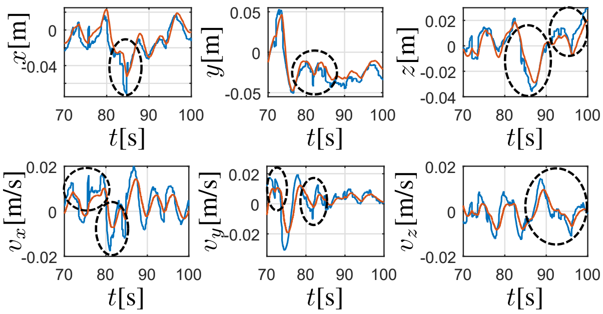

Figure 9: Position (top row) and velocity (bottom row) state estimation results. The localization pipeline is shown in blue, and the Model-based EKF in red. Dashed circles indicate localization discontinuities

.

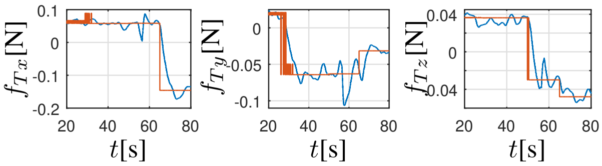

Figure 10: $\mathbf{F}_T$, actual (red) and estimated (blue) values.

get. Future work aims to address the localization concerns discussed in Section 4.1, additional speed improvements to the body-frame motion planner, and expanding robustness constraints to better handle large disturbance sets and the nonlinear attitude dynamics.

REFERENCES

[1] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, 2014.

[2] E. Papadopoulos, F. Aghili, O. Ma, and R. Lampariello, "Robotic Manipulation and Capture in Space: A Survey," *Frontiers in Robotics and AI*, vol. 8, p. 686723, July 2021.

[3] J. L. Forshaw, G. S. Aglietti, N. Navarathinam, H. Kadhem, T. Salmon, A. Pisseloup, E. Joffre, T. Chabot, I. Retat, R. Axthelm, S. Barraclough, A. Ratcliffe, C. Bernal, F. Chaumette, A. Pollini, and W. H. Steyn, "RemoveDEBRIS: An in-orbit active debris removal demonstration mission," *Acta Astronautica*, vol. 127, pp. 448–463, Oct. 2016.

[4] J. Šilha, J.-N. Pittet, M. Hamara, and T. Schildknecht, "Apparent rotation properties of space debris extracted from photometric measurements," *Advances in Space Research*, vol. 61, pp. 844–861, Feb. 2018.

[5] T. P. Setterfield, D. W. Miller, J. J. Leonard, and A. Saenz-Otero, "Mapping and determining the center of mass of a rotating object using a moving observer," *Intl. J. of Robotics Research (IJRR)*, vol. 37, no. 1, pp. 83–103, 2018.

[6] R. Lampariello, H. Mishra, N. W. Oumer, and J. Peters, "Robust motion prediction of a free-tumbling satellite with on-ground experimental validation," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 10, pp. 1777–1793, 2021.

[7] S. Stoneman and R. Lampariello, "A nonlinear optimization method to provide real-time feasible reference trajectories to approach a tumbling target satellite," in *i-SAIRAS*, (Beijing, China), June 2016.

[8] R. Lampariello, "Motion Planning for the On-orbit Grasping of a Non-cooperative Target Satellite with Collision Avoidance," *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, vol. 1, pp. 636–643, 2010.

[9] A. Fejzi and D. W. Miller, *Development of Control and Autonomy Algorithms for Autonomous Docking to Complex Tumbling Satellites*. PhD thesis, Massachusetts Institute of Technology, 2008.

[10] F. Aghili, "A prediction and motion-planning scheme for visually guided robotic capturing of free-floating tumbling objects with uncertain dynamics," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 634–649, 2012.

[11] D. C. Sternberg and D. Miller, "Parameterization of fuel-optimal synchronous approach trajectories to tumbling targets," *Frontiers Robotics AI*, vol. 5, no. APR, pp. 1–11, 2018.

[12] J. Virgili-Llop, C. Zagaris, R. Zappulla, A. Bradstreet, and M. Romano, "A convex-programming-based guidance algorithm to capture a tumbling object on orbit using a spacecraft equipped with a robotic manipulator," *International Journal of Robotics Research*, vol. 38, no. 1, 2019.

[13] K. Albee, C. Oestreich, C. Specht, A. Terán Espinoza, J. Todd, I. Hokaj, R. Lampariello, and R. Linares, "A Robust Observation, Planning, and Control Pipeline for Autonomous Rendezvous with Tumbling Targets," *Frontiers in Robotics and AI*, vol. 8, p. 234, 2021.

[14] T. Smith, J. Barlow, M. Bualat, T. Fong, C. Provencher, H. Sanchez, and E. Smith, "Astrobee: A New Platform for Free-Flying Robotics on the ISS," tech. rep., 2016.

[15] B. E. Tweddle, T. P. Setterfield, A. Saenz-Otero, and D. W. Miller, "An open research facility for vision-based navigation onboard the international space station," *Journal of Field Robotics*, vol. 33, no. 2, pp. 157–186, 2016.

[16] T. P. Setterfield, D. W. Miller, J. J. Leonard, and A. Saenz-Otero, "Mapping and determining the center of mass of a rotating object using a moving observer," *The International Journal of Robotics Research*, vol. 37, pp. 83–103, 2018.

[17] A. Terán Espinoza, *Versatile inference algorithms using the Bayes tree for robot navigation*. PhD thesis, Massachusetts Institute of Technology, 2020.

[18] C. Oestreich, A. Teran, J. Todd, K. Albee, and R. Linares, "On-Orbit Inspection of an Unknown, Tumbling Target using NASA's Astrobee Robotic Free-Flyers," in *Conference on Computer Vision and Pattern Recognition*, (Virtual), 2021.

[19] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Trans. Robotics*, 2020.

[20] H. Mishra *et al.*, "A geometric controller for fully-actuated robotic capture of a tumbling target," in *2020 American Control Conference (ACC)*, pp. 2150–2157, 2020.

[21] K. Albee, M. Ekal, and C. Oestreich, "A Brief Guide to Astrobee's Flight Software," 2020.