(Preprint) AAS 24-494

SPACECRAFT THRUSTER PLUME MODELING FOR INTER-SPACECRAFT IMPACTS

Scott Piggott[*], Jennifer Wood[†], Thibaud Teil[‡] and Hanspeter Schaub[§]

Performing orbital servicing on a general spacecraft means occasional plume impingement is unavoidable. Developing the techniques for inter-spacecraft plume interaction requires investigation into the unique problems for spacecraft control when vehicles are flying in close formation. When an approach vehicle deposits spent propellant onto the client spacecraft in a classic passive/active vehicle control pair, it will perturb the client spacecraft's rotational as well as translational states. Modeling the effects of thruster plumes impinging on a spacecraft's structure often relies heavily on complex numerical computational fluid dynamics (CFD) methods in order to obtain sufficient accuracy. This computational complexity can make the inclusion of full CFD modeling in realtime and faster-than-realtime simulations of dynamics and algorithmic control strategies infeasible.

This paper presents a fast model for approximating inter-spacecraft thruster plume interactions and explores the dynamical perturbations that result from these effects. Spacecraft are modeled as a series of flat plates and the plume impingement applied is modeled as a cone with constant gas-density that emanates from the throat of the servicer thruster. The subset of the plume cone that intersects with the plate models determines the resulting net force and torque on the plumed spacecraft. This reduced fidelity plume deposition model is then compared to a more complex model that uses a shader-based intersection approach to compute the net force and torque for the same relative geometry. These techniques are compared and assessed using a series of plates based on a complex set of spacecraft geometry against the computed force and torque along with the overall computation time.

INTRODUCTION

The In-Space Servicing, Assembly, and Manufacturing (ISAM) of on-orbit spacecraft by servicer vehicles is of increasing interest. Unfortunately, successful servicing is complicated by the fact that the client spacecraft are frequently not in the desired attitude for docking or are tumbling quickly enough that docking with them becomes effectively impossible. Capturing a target spacecraft can be thought of as occurring in two phases. The first phase is the detumbling of the client object and the second is capture. In Reference[7], a comparative analysis of different methods of both detumbling and capturing a target spacecraft is performed. The detumbling of a client spacecraft via plume is not included in the options assessed, but the reviewed concepts include the use of robotic arms, harpoons, and nets. Conceptually one can see that successfully capturing a target

[*]Atomos Space, Broomfield CO, 80020.

[†]Research Associate, University of Colorado at Boulder, Boulder, CO, 80303.

[‡]Laboratory for Atmospheric and Space Physics, Boulder, CO, 80303

[§]Professor and Department Chair, Schaden Leadership Chair, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, 431 UCB, Colorado Center for Astrodynamics Research, Boulder, CO, 80309. AAS Fellow, AIAA Fellow.

object is enormously difficult if the target vehicle has significant relative momentum compared to the servicer.

This paper focuses on the implementation by one spacecraft of a desired state-change on a separate spacecraft using a thruster plume, but it is directly extensible to other cases where spacecraft plumes can impact structure. For example, when approaching a large object like the International Space Station (ISS), servicing spacecraft must first prove that their thruster systems do not represent a hazard to the ISS structure, particularly its solar arrays. Knowledge of relative plume dynamics providea a means of both what the risks are to remote spacecraft as well as what opportunities there are for orienting that structure to avoid damage from a servicer's plumes. Another direct extension is modeling what the expected plume self-impingement will be on a servicer's structure, particularly for elements that are dynamically moved such as rotating solar arrays or robotic elements. It will also be applicable for cases where the thruster itself can be moved relative to the servicer spacecraft rigid structure for all cases of expected plume impingement.

It may be possible to rectify the attitude of the client spacecraft by using the onboard thruster systems of the servicer to impart torque to the target spacecraft, detumbling it such that it can be captured. This operation requires a significant amount of propellant be deposited onto the client spacecraft, but the amount is small compared to the total mass of propellant required to meaningfully change the client spacecraft's orbit. In fact, if carefully selected, the implemented relative orbits can minimize the propellant cost for the servicer spacecraft to slow down the client spacecraft, with the understanding that increased time to complete the operation is traded for decreased use of propellant.

A primary effect of this operation on the client spacecraft is to change the client's orbit. This is simply conservation of momentum where the exhaust gases from the servicer spacecraft impact the client spacecraft, and deposit their momentum onto its surface, changing the overall velocity of the client spacecraft. When the two spacecraft are flying in very close proximity, the client spacecraft velocity is changed in an equal and opposite manner (100% deposition), but more typically only a small percentage of the propellant from the servicer is deposited, resulting in a smaller change to the client's orbit than that of the servicer. The manner in which the propellant impacts the client spacecraft is complex and contains some fraction of deposited, reflected, and scattered propellant, but for this study, it is assumed that the propellant is entirely deposited on to the client spacecraft.

There are two references found that use a technique very similar to that examined in this work. The first paper used a database of force/torque data generated from computational fluid dynamics (CFD) to model the effect of thruster plumes on the client spacecraft in [10]. This used Direct Simulation Monte Carlo (DSMC) techniques to model the effect of the thruster plume on a fairly large rocket body. Interestingly, in this paper the minimum distance that was considered between the rocket body and the servicer spacecraft was 8 m. This is a considerable distance when thruster plumes are being considered. A thruster plume expands rapidly in a vacuum, so as distance increases the magnitude of the torque exerted on the target spacecraft rapidly decreases because the target is essentially painted in a uniform field of propellant. This factors in to the high cost, ($> 10kg$) , of detumbling calculated for the use of this expensive and time-consuming, albeit successful, technique. This paper is also interesting for cross-comparison with a paper from the last research where CFD was also used in [5].

The same authors then extend the work to comprehensive detumble strategies and rates in [8]. A full control law is developed that aims the plume at the client object and then maintains the desired relative state compared to the client object. Desired target points are defined, and then pinged with

the thruster plume in simulation. As one would anticipate, reducing the rotation rate about the long axis of the rocket body requires much less propellant than about the central axis. It is possible to reduce the rotation rate of the object about this axis, but it takes vastly more propellant than the other two axes because the aim point is much harder to hit, and the moment arm is much shorter.

In [13], a similar concept, that of an "ion shepherd", is discussed where a plume of charged ions is emitted by the shepherd (servicer) vehicle that creates a plasma on the client spacecraft, generating forces and torques on the exposed surface. This paper does not attempt to model the effects of a plasma or even cases where the flow is sufficiently dense to create atmospheric effects. It assumes free molecular flow of a gas in vacuum. Correlating this design against real flow effects, both for ion shepherd as well as cold-gas thrusters in vacuum is the subject of further current research.

This work seeks to begin investigating strategies for controlling these relative dynamics by implementing effective simulation techniques. The spacecraft thruster plume is first modeled as an even-density cone that emanates from the throat of the thruster oriented along a constant vector in the servicer spacecraft's body frame. The client spacecraft is modeled as a series of flat plates assembled into a cube and the relative geometry of each plate is assessed compared to the thruster location to determine the amount and location of gas that is deposited onto the client spacecraft. This information is then used to determine the total force and torque imparted on to the client spacecraft by the plume from the servicer spacecraft and the relative kinetics are assessed.

With this flat-plate model computed, a method for interacting with more complex vehicle geometry is assessed. Using a 3D visualization application, a simulated camera is positioned at the modeled location of the servicer thruster and oriented along the boresight of the thruster. The simulated camera is commanded to generate a depth map of the in-view vehicle geometry rather than a realistic image. The resulting depth map texture encodes the range to the location on the client spacecraft in the output color of each pixel. This is advantageous because spacecraft models of arbitrary complexity can be directly imported, theoretically giving a higher-fidelity way of representing the spacecraft-to-spacecraft plume interaction. It also directly computes the self-shadowing that the client's surfaces may experience so that plume rays cannot be double-counted if they would impinge on multiple plates.

PLUME IMPACT ON PLATES

Reference [1], models the self-impingement geometry as a series of infinite plates and show that if a thruster plume is modeled as a continuous cone with half-angle of α, the area of the plume that interacts with a given flat plate is the family of conic cross-sections similar to the shape of the classic Keplerian orbits. For thruster firings where the thrust vector (plume 180° reversed) is parallel to the client plate normal, the conic shape is a circle. For cases where the angle between the thruster and the plate is less than $90^\circ - \alpha$ away from the plate normal, but greater than 0, it is an ellipse of increasing eccentricity, reaching eccentricity of 1 (parabolic) at $90^\circ - \alpha$, and then hyperbolic between $90^\circ - \alpha$ and $90^\circ + \alpha$. When the incidence angle exceeds this $90^\circ + \alpha$, the plume no longer intersects the plate and so the plume deposit fraction is zero. Figure 1 shows a good representation of these conic sections as a function of incidence angle.

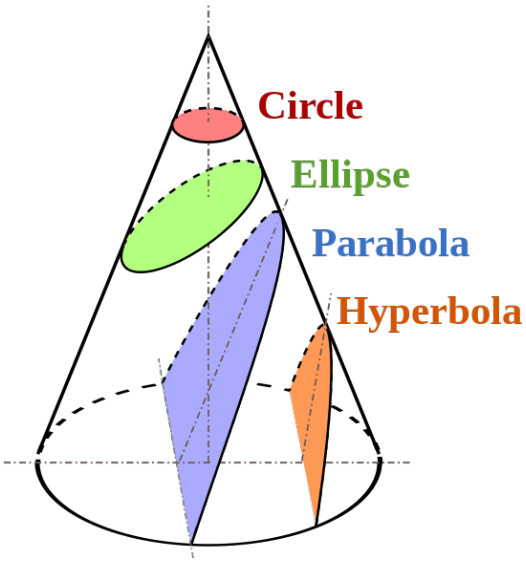

Figure 1. Plane/Cone intersection diagram

With the shape of the plume contact area as a function of incidence angle on an infinite plate defined, the question of how to extend these calculations to a finite plate arises. Finite plates do not lend themselves to analytical analysis because of the unique areas that are carved out when the thruster plume intersects with the plate and cannot be calculated with a smooth integral like that used in [1]. For example, a given circular plume can cover an entire plate or leave up to four sub-sections of the plate missed, which prevents directly integrating the plume force. With this limitation, the impact of the plume on finite plates can be computed with a finite element approach where the total plume area is subdivided into elements and the sum of the total plume elements is 100% of the force that is output at the thruster's throat. If it is assumed that any pressure distribution model will be applied on top of this mesh grid, the design goal for the grid is to evenly distribute the plume elements throughout as a series of rays spread throughout the plume at constant angular separation.

Cartesian model for thruster plume mesh

In order to model the impact of the thruster plume with a plate, a purely Cartesian mesh can be constructed that has constant offset angles between each mesh element. Rather than revolving around the center in polar coordinates, a 2D mesh of points can be constructed that treats the thruster like a camera sensor in that the thruster field-of-view (FOV) is square with the major dimensions reaching their maximum extent at the plume cone half-angle α. If we take the length dimension to be x, and the width dimension to be y, the plume mesh points can be expressed according to

Equation 1.

$$\begin{aligned} \psi_i &= \frac{2i}{N_x}\alpha \\ \beta_j &= \frac{2j}{N_y}\alpha \\ \phi_T &= \sqrt{\psi_i^2 + \beta_j^2} \\ \theta_T &= \arctan\frac{\beta_j}{\psi_i} \\ {}^P\hat{T}_{mesh} &= [\sin\theta_T \sin\phi_T, \cos\theta_T \sin\phi_T, \cos\phi_T]^T \end{aligned} \tag{1}$$

Using this, the total plume that impacts the target plate can be expressed according to.

$$\sum_{i=-\frac{N_x}{2}}^{\frac{N_x}{2}} \sum_{j=-\frac{N_y}{2}}^{\frac{N_y}{2}} \hat{T}_{i,j} \frac{|\vec{T}|}{N_x N_y} \tag{2}$$

This representation ensures even spacing between points, but issues arise at the extents of the plume. The goal for this paper is to model the spacecraft plume solely as a cone with half-angle α as described above. Because the Cartesian representation is totally square, there are elements of this plume that do not fall within the desired conic section. The length/width of the parameterization can be reduced in order to deal with this modeling issue, but then there are elements of the desired cone that will fall outside of the meshed region, reducing the total footprint of the plume cone. Figure 2 shows an example of this plume model along with the compromises that are made in the overall plume footprint.

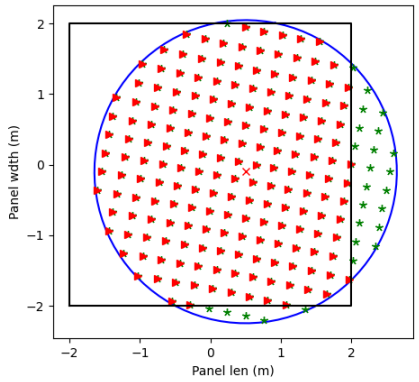

Figure 2. Intersection of Cartesian Plume Model and Finite Plate

As this plot illustrates, the Cartesian modeling approach gives totally even spacing between the different points in the mesh. However, the mesh is assembled as a rectangular volume and invalid points (outside the conic envelope) are simply dropped from the mesh. While this issue can easily be compensated for when constructing the mesh, it also results in some wider spacing along the edge

of the circle where plume elements are dropped in an uneven fashion. This is certainly an improved physical model for even spacing in the plume, but a technique that combines the constraints of a polar coordinate approach and the Cartesian approach would be ideal.

Spiral model for thruster plume mesh

An improved mesh concept for modeling the interior points of the thruster cone would enforce even spacing between each point and be parameterized in polar coordinates such that the total angular distance from the origin can be directly controlled. An Archimedean Spiral, which can be found in [11], fulfills these two requirements. This geometric construct has constant spacing between each successive spiral and, with regular spacing between each subsequent point, even spacing can be accomplished while specifying both the maximum extent and the total number of points for the mesh. An example diagram of this type of spiral is shown in 3.

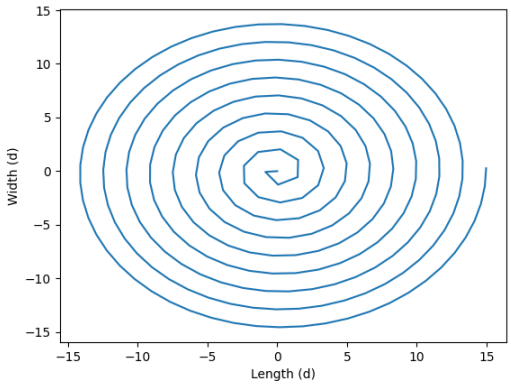

Figure 3. Example Archimedean Spiral

The fundamental equation for this spiral is given in Equation 4.

$$r = b\theta \tag{3}$$

In this equation, the angle θ represents the angle of a point on the spiral. When the angle of a point is 2π radians, the point is located along the x axis at a distance from the origin. With this spiral then, each turn that crosses the origin is separated from the previous turn by the relationship show in Equation 4.

$$h = 2\pi b \tag{4}$$

The parameter b is a free variable that essentially defines how ”tight” the spiral is as it unwinds from the origin. But in order to meet the desired spiral properties for the mesh parameterization, it can be fully defined. Because the separation between each point on the spiral and the next winding of the spiral is the h parameter from above, the distance between subsequent points on the spiral should also be set to h in order to obtain equal radial and tengential spacing along the spiral. So, if the number of desired points in the mesh is specified as N_s, that requires that the total arc-length of the spiral be given by Equation 5.

$$L_s = 2\pi b N_s \tag{5}$$

Additionally, we know that the maximum extent of the spiral, r from above should be at the half-angle of the cone when the maximum angle θ is reached and that the total length of the spiral should be as shown in Equation 5. These two constraints can be combined to determine what the appropriate setting for the parameter b should be when combined with the equation for the total arc-length of an Archimedean spiral shown in Equation 6.

$$L_s = \frac{b}{2}\left[\theta\sqrt{1+\theta^2} + ln\left(\theta + \sqrt{1+\theta^2}\right)\right] \tag{6}$$

As both of the equations for desired total arc length and computed total arc-length are linear in b, Equations 6 and 5 are set as equivalent and the resulting relationship can be iterated using a linearization to solve for the required θ angle. Then the relationship between max extent r and full angle θ is used to solve for b. The equations for the unit vector at each point are assembled similarly to Equation 1 and this relationship is shown in Equation 7.

$$\begin{aligned} \theta_i &= \frac{i}{N_s}\theta_M \\ r_i &= b\theta_i \\ \phi_i &= r_i \\ {}^P\hat{T}_{mesh} &= [sin(\theta_i)sin(\phi_i), cos(\theta_i)sin(\phi_i), cos(\phi_i)]^T \end{aligned} \tag{7}$$

This relationship combines the benefits of both of the previous approaches with minor downsides. Deriving the correct values for θ and b requires some iteration, but once that is defined for a given thruster cone, the Archimedean spiral provides an even distribution of points, with the specified number of points, along with an easy mesh computation and definition at runtime for the thrust field across the plate surface. Figure 4 shows a plot of this plume mesh configured identically to the above when impacting the same flat plate.

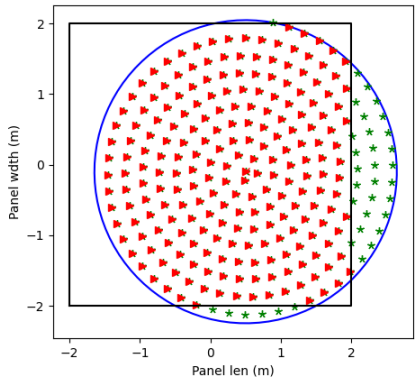

Figure 4. Intersection of Archimedean Spiral Plume Model and Finite Plate

This plot provides good visual confirmation that the desired spacing is provided via the spiral implementation and that all of the computed mesh points fall within the area of the cone. It also matches up closely with the previous meshing approach in the count of points falling within the finite plate area.

PLATE MODEL RELATIVE PLUME DYNAMICS

The improved element spacing provided by the Archimedean spiral mesh is used in a six-degree-of-freedom (6DOF) simulation in order to explore the impacts of these vehicle-to-vehicle thruster interactions on the overall relative dynamics of the two spacecraft. A Basilisk (REFERENCE HERE) simulation was instantiated with two spacecraft operating in close proximity to each other. The spacecraft that will be plumed is designated as the ”client” spacecraft and the spacecraft that will be pluming the client is designated as the ”servicer.” The servicer spacecraft is configured with a single thruster set up to plume the client spacecraft and the client is modeled as a cubic shape with length of 1 m, such that the servicer's thruster is checked for intersection with the 6 cube sides simultaneously. The two spacecraft are then placed into orbit with some relative geometry and the servicer's thruster is fired periodically to create some desired plume effects on the client. Figure 5 shows an example set of geometry between these two spacecraft.

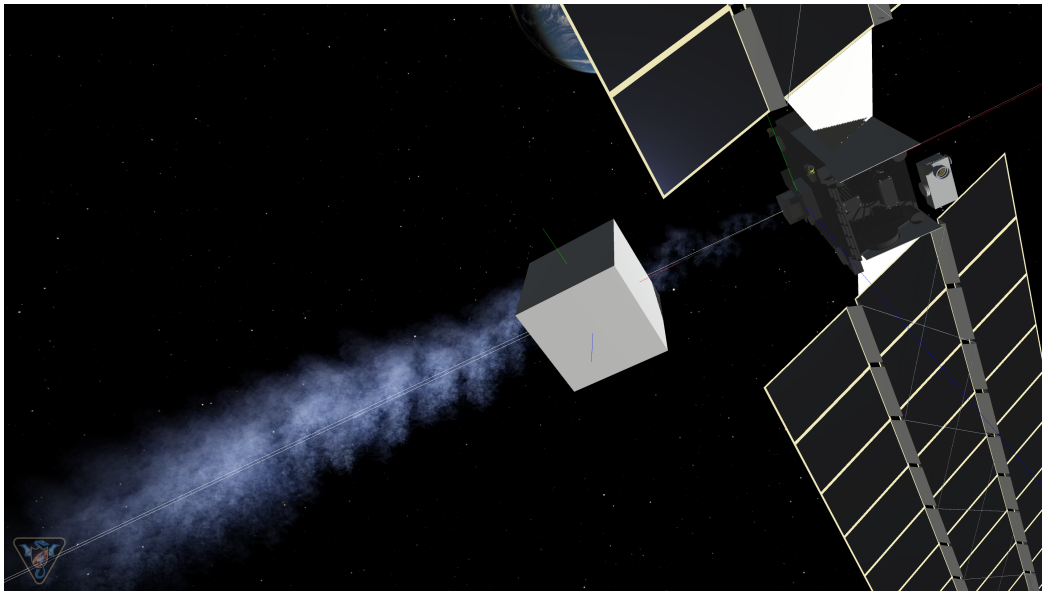

Figure 5. Example geometry of the servicer pluming the client spacecraft

While the initial conditions are somewhat arbitrary, the two vehicles in the simulation are configured to have masses of 500 kg, the thruster of the servicer is placed 1.5 m from the +x panel of the client spacecraft, and they are oriented in the same Hill-frame-relative attitude such that the thruster on the servicer, pointed in the -x direction for pluming, is directly orthogonal to the +x panel of the client spacecraft. It is further aligned with the y-vector in the Hill frame which is collinear with the center-of-mass of both spacecraft. The thruster is given a half angle of 15°, for a total thruster cone plume width of 30° The simulation is initialized and run for a period of 20 seconds to give a period of quiescent flight prior to igniting the servicer's thruster. The servicer's thruster is then fired for 10 seconds, after which the simulation is left to drift for a second period of 20 seconds. With these conditions neither spacecraft should experience significant attitude perturbations. The full thruster plume should impact in the center of the client's plate, and some relative translational dynamics should be induced. Figure 6 shows the surface of the modeled plate of the client along with the mesh points flagged as impacting the spacecraft during thrusting.

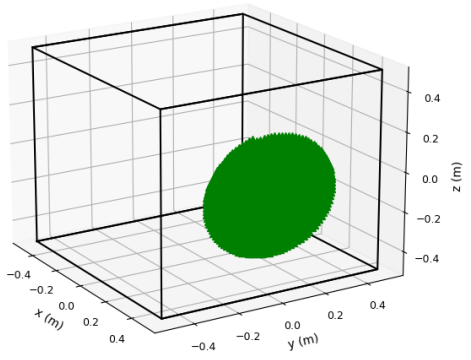

Figure 6. Plume mesh intersection with client's plate surface during thrusting orthogonally

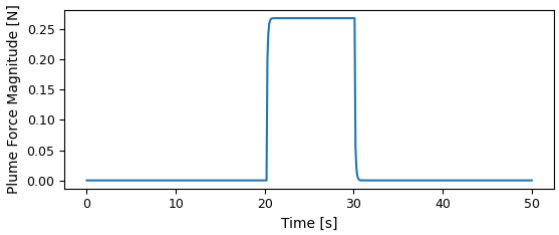

Figure 7. Plume force imparted to client spacecraft from servicer with orthogonal thrust

As this plot shows, the impact of the thruster spiral mesh falls directly in the center of the client's plate, with a spread of 0.75 m edge-to-edge in both the horizontal and vertical directions on the client's plate. So, with this plume impact, the thruster force is transferred to the client spacecraft 100%. The thruster is modeled as having 0.27 N of thrust for the purposes of this simulation so the client spacecraft should experience 0.27 N of thrust for the duration of thrusting. The total force imparted on to the client spacecraft from the plume model is shown in Figure 7.

Interesting relative kinematics are introduced once this thrust is imparted onto both spacecraft: the servicer spacecraft thrusts away from the client, and when the plume particles impact the client spacecraft, the client is pushed away with an equal force in the same direction as the thrust imparted to the servicer. So, in a relative sense, the total ΔV imparted to the system is effectively doubled compared to the thrust applied (if the spacecraft masses are equivalent). Figure 8 shows a plot of the relative velocity observed between the client and servicer spacecraft.

For two 500 kg spacecraft, with a thrust of 0.27 N applied for 10 seconds, accrual of 0.011 m/s of relative velocity is expected in the y-axis of the Hill frame (x radial, z angular momentum, y downrange). This matches the observed simulation results precisely with multiple orders of magnitude lower velocity in the other two axes of the Hill frame. For this simple test case, the simulation performance matches expectations precisely and more complex motion can be examined and compared with the other simulation tools.

The previous result shows the overall correctness of the model for a simple case where the thrust

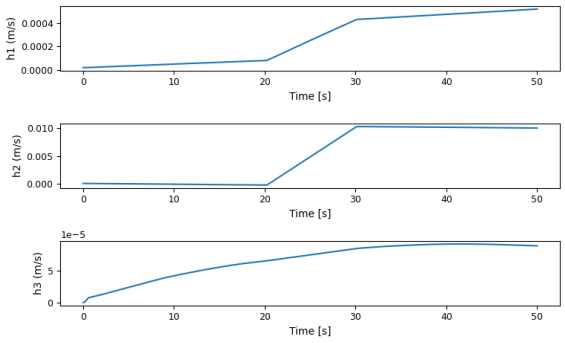

Figure 8. Relative velocity between the client and servicer spacecraft with orthogonal thrust

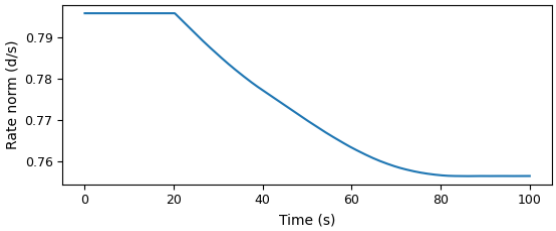

Figure 9. client spacecraft angular velocity during detumble

is placed along the centerline of the target object, collinear with the center-of-mass (CoM). As such, very little torque is imparted to the target object, and it is not rotating initially compared to the orbit so it does not rotate substantially while being plumed. To demonstrate the model working with an initial rotation applied to the client spacecraft, a new configuration demonstrating a detumbling scenario was constructed in the simulation.

For this scenario, the servicer spacecraft was placed at the same relative offset forward of the client spacecraft (1.5 m downrange). The mass, CoM, plate, and thruster properties are also consistent with previous simulation. The client spacecraft is rotated by 70 degrees to place the -z face of the client in the servicer plume. An angular velocity of $1.0\frac{^\circ}{s}$ in -y such that the -x face of the client spacecraft is rotating into view of the thruster. The servicer spacecraft is rotated 10 degrees about the angular momentum vector such that it is pluming the face of the client spacecraft against the direction of the client's rotation. This is a simple approach to detumbling a spinning cube, but it provides great insight into some constraints of the detumbling problem. For this simulation, the thruster is fired for 60 seconds instead of 10 seconds so that the separation between the two vehicles increases significantly while the burn is ongoing.

As Figure 9 shows a plot of the angular velocity of the client spacecraft over time for the duration of the simulation.

As this plot shows, the angular acceleration of the client spacecraft drops dramatically as the simulation progresses. This comes from two primary causes. The first is that as the separation distance increases, the percentage of the plume that impacts the client spacecraft drops because the client spacecraft no longer absorbs 100% of the plume emanating from the servicer's thruster. Figure 10 shows the two primary plates and their associated plume deposition after the thrusters

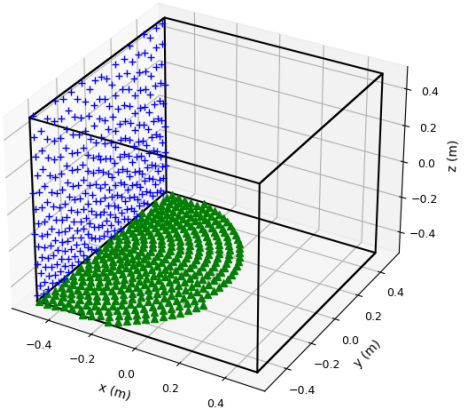

Figure 10. client spacecraft -z and -x plates for 50 seconds after thrust initialization

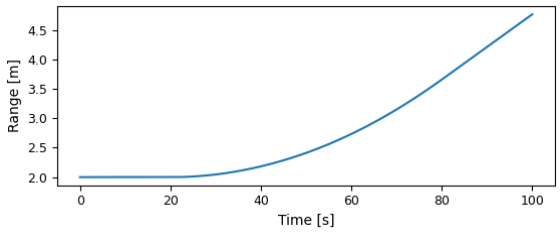

Figure 11. servicer/client relative range for detumbling case

have been firing for 50 seconds.

In this plot, the -z plate plume deposition is shown with the green triangles, and the -x plate plume deposition is shown with the blue + symbol. This case shows what also begins to become a problem for further rate reductions on the client spacecraft. As the separation distance increases between the servicer and client spacecraft, the effective area of the plume cone widens linearly. Then as the cone widens, the ability for the servicer spacecraft to target a certain part of the client structure becomes less effective. Then once a critical distance is reached, the reduction in plume density along with the saturation of plume across the surface of the visible plates only results in pushing the client spacecraft rather than applying a torque to it. Figure 11 shows a plot of the relative spacecraft

At initialization, using the baseline thruster plume cone model (30° full-angle), the two spacecraft are separated by 1.5 meters and the total width of the plume cone is 0.75 meters. This allows the center of the thruster plume to be targeted solidly within a specific part of the client spacecraft, with effective tumble rate reduction over time. Then after firing for 40 seconds, the relative range has grown to 2.2 meters, with an effective cone width at the client of 1.1 meters, completely covering the surface of the client with the cone in most cases. After firing for an additional 20 seconds the plume cone width has grown to 1.45 meters, almost double its initial width, and the rate of momentum reduction flattens out. This appears to be largely driven by the loss of targeting to a specific region of the client spacecraft because the relative velocity between the two vehicles continues to increase nearly linearly until thrust is stopped after 60 seconds. Figure 12 shows a plot of the relative velocity

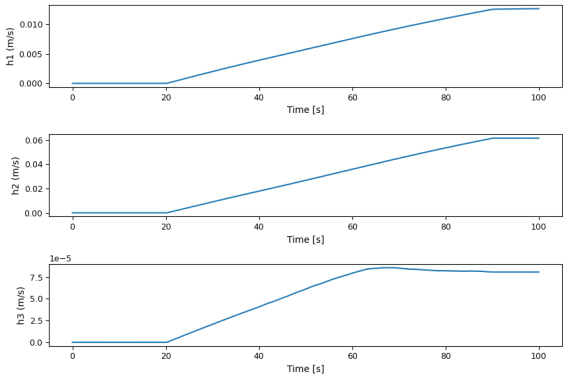

Figure 12. servicer/client relative velocity for the detumbling simulation

between the two vehicles over the simulation duration.

As this plot shows, the primary axis of acceleration (y-Hill) continues to show steady acceleration until the thrust is stopped. The momentum from the thruster plume continues to change the state of the client spacecraft, but in a very even way across the spacecraft body, rather than in a specific axis. The effectiveness of this detumbling strategy is determined by the relative range between client and servicer, the narrowness of the the thruster plume, and the ability to aim the plume at a specific region of the client spacecraft.

COMPARISON OF RANGE IMAGE PROCESSING APPROACH TO PLATE MODEL

There are some limitations to the above approach. The first is that for each flat surface on the spacecraft, a unique plate must be defined to represent that surface. For a spacecraft with complex geometry, this would likely require that thousands of plates be computed and specified to build up the spacecraft model. This model is further limited in that it does not take any self-shadowing of plates into account when computing the total plume imparted to the client spacecraft. The first limitation is difficult to entirely resolve, although a CAD geometry output like STL or OBJ could be used to derive the plates programmatically to reduce the manual effort required to compute the plate shapes. Increasing the complexity of the modeled geometry by adding surfaces will increase the amount of computation required to calculate the total plume. The second limitation is not accounted for here, but could be overcome by using a hierarchy of plates instead of a single plate model assessed atomically. To overcome these limitations, a second approach is considered.

Vizard, the companion visualization to Basilisk, is built upon the Unity 3D Development Platform. The appearance of objects rendered in Vizard is determined by the graphics shaders applied to the objects. Unity internally calculates many global variables and texture maps for the shaders in use, including a non-linear depth map. The depth for each pixel in the camera view is recorded into the Unity depth map non-linearly to allow for finer distinction of rendering order for objects closer to the camera. The depth map is used to cull hidden objects from being rendered to the screen, to map shadows, and to include semi-transparent effects in the scene, amongst other uses. In Vizard, a secondary camera can be instantiated and configured to desired values for field of view, position, orientation, and near and far clipping planes. It can also be configured to render the depth map of the objects within its field of view instead of the visual camera view.

When the depth map mode of an Instrument camera in enabled, a custom DepthMap shader is

applied as a "lens" before the camera. The shader samples the internal Unity non-linear depth map and utilizes the Unity Linear01Depth shader helper macro to linearize the internal depth texture. A value between 0 and 1 is returned for each (u,v) coordinate where 1 is the maximum depth (the far clipping plane of the Instrument camera). The depth value at each (u,v) coordinate is encoded in the output color of the fragment texture in the red-channel. The Instrument camera's rendered texture reveals the depth of objects in the scene relative to the camera in shades of red. The Basilisk spacecraft simulation can request images be saved at regular intervals, using the "renderRate" setting, or by clicking the Image button on the Instrument camera's display panel within Vizard. To decode the depth of a specific pixel in an image, sample the RGB color values of the pixel and calculate the depth as:8. In this equation r_c is the distance to the far clipping plane and R_v is the red-channel color value measured for the pixel point.

$$r_d = r_c * \frac{R_v}{256} \tag{8}$$

It should be noted that the depth texture values are more accurate for objects closer to the camera and error in the calculated depth increases with distance from the camera.

Using these range-shader output images, the spacecraft plume can be computed very similarly to the spiral or Cartesian meshes described above. Each pixel in a rendered image is taken as a mesh point, with a corresponding azimuth angle (θ) and elevation angle (ϕ) defined per pixel. For each pixel location (i_{pix}, j_{pix}), the ray is defined by Equation 9.

$$\begin{aligned} l_{pix} &= \sqrt{i_{pix}^2 + j_{pix}^2} \\ \phi_i &= \frac{l_{pix}}{l_{cone}}\alpha \\ \theta_i &= \arctan\left(\frac{j_{pix}}{i_{pix}}\right) \\ {}^P\hat{T}_{mesh} &= [sin(\theta_i)sin(\phi_i), cos(\theta_i)sin(\phi_i), cos(\phi_i)]^T \end{aligned} \tag{9}$$

Similarly to the Cartesian plume approach, at each mesh point the angular spread (ϕ) must be checked to determine if the computed mesh point falls within the thruster plume cone. Only mesh points falling within the thruster plume cone are summed using Equation 10. The total number of valid mesh points is N_{good} and the width of the cone half-angle in pixels is l_{cone}.

$$\sum_{i=-\frac{N_x}{2}}^{\frac{N_x}{2}} \hat{T}_{i,j} \frac{|\vec{T}|}{N_{good}} \tag{10}$$

The relative geometry and kinematics are still computed for this thruster plume, but are performed automatically by the Vizard visualization as part of rendering the objects at the locations and orientations specified by the Basilisk simulation at each timestep. The range images leverage these internal calculations of geometry and depth from the camera. This approach also removes the main limitations of the previous method in a single stroke. Unity directly ingests CAD geometry so that the user need not create specific plate modes or any additional geometry for their spacecraft model. It will additionally easily accommodate non-rectangular plates (triangles, cylinders, spheres, etc.) which would be very programmatically tedious for the user to model as rectangular plates. Additionally, because it is creating the image of what the thruster "sees," the self-shadowing of the

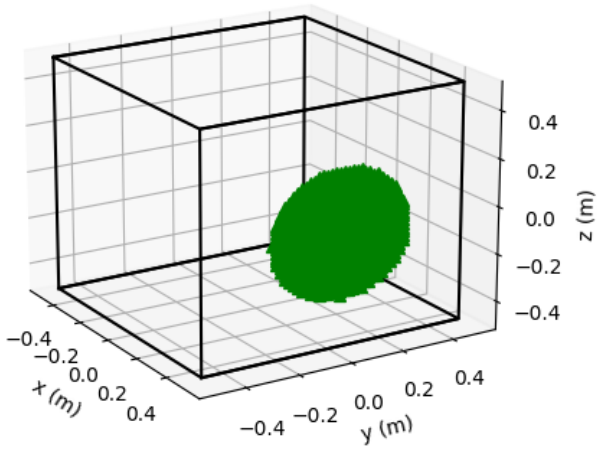

Figure 13. Depth-shader orthogonal plume interaction with client spacecraft

client spacecraft's geometry will fall out from the examination of the range image. This would also work for self-shadowing from the servicer and even multi-spacecraft shadowing for complex multi-spacecraft geometries.

In order to perform an apples-to-apples comparison between the two techniques presented here, the simulation is always driven closed-loop by the simple geometry model. Then, the outputs of the range-shader model (force/torque on the spacecraft) are compared to the plate model from above. The same simulation conditions with the servicer plume hitting a single cube face orthogonally and the spinning cube were implemented here for comparison purposes. The first case analyzed is that of the orthogonal thrust onto the spacecraft plate. Figure 13 shows a scatter plot of the computed plume impact points along with a rendering of the cubic spacecraft object being plumed.

As this diagram shows, the plume shape is qualitatively identical to that generated from the previous analysis approach with Figure 6. The plume strikes the center of the +x face of the client spacecraft, with a circular plume deposition, with a radius out from the center of the plate of approximately 0.3 m. All of the thruster points hit the single plate so the momentum transfer is complete just as with the other analysis technique. Ideally the total force imparted on to the client spacecraft should match closely in this case as well. Figure 14 shows a comparison between the force values computed for the two techniques with the depth-map shader approach computed when the force is active and shown by the green x points.

As this figure demonstrates, the agreement between the two techniques is quite good. The total value of the force imparted in the -x direction is approximately 0.27 N, and the differences in both the direction of the force and in the orthogonal directions are much less than 1% of the total force imparted by the thrust. The two techniques agree for this simple case.

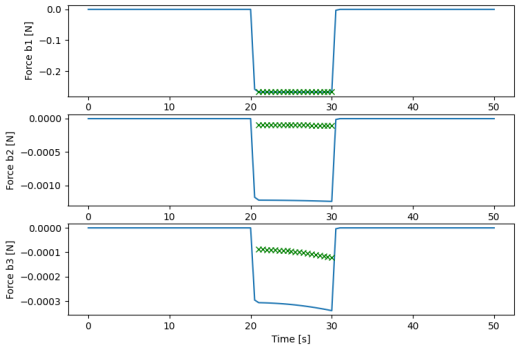

Figure 14. Depth-shader orthogonal plume force computation compared to spiral with plates

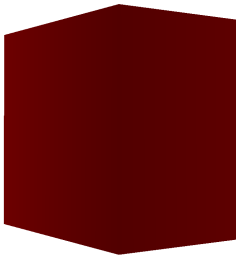

Figure 15. Depth-shader range map for cubic spacecraft model

Similarly, the case of the tumbling client spacecraft was also computed with conditions identical to that described above for the relative spacecraft. This case engages the model in computing force/torque values for multiple plates simultaneously. For the data-point in question, Figure 15 provides the range-map that the thruster model ”sees” after 50 seconds of thruster firing. In the image, essentially all of the -x face of the spacecraft can be seen, and then a portion of the -z face of the spacecraft is also seen, which terminates at the right side of the image.

Then with this range image, the plume intersection points for the spacecraft are computed and the net thruster plume is assembled. Figure 16 shows the scatter plot for the points of the plume and where they intersect with the spacecraft. This figure also shows very good qualitative agreement with the spiral analysis technique in that the same portions of the spacecraft cube are covered with plume points at approximately the same density as is observed with the spiral.

The force comparison between the two techniques is shown in Figure 17 with the spiral plate approach again show in blue with the depth-map technique marked with green xs. While the comparison between the approaches is not in as precise of agreement as in the orthogonal cube case, it is still quite good. The largest deviation between the two approaches occurs at about 60 seconds of simulation time and has a deviation in thrust magnitude of 10% with a smaller deviation in thrust direction. The source of this difference is still under investigation, but the overall agreement in results is encouraging for both of the methods under test.

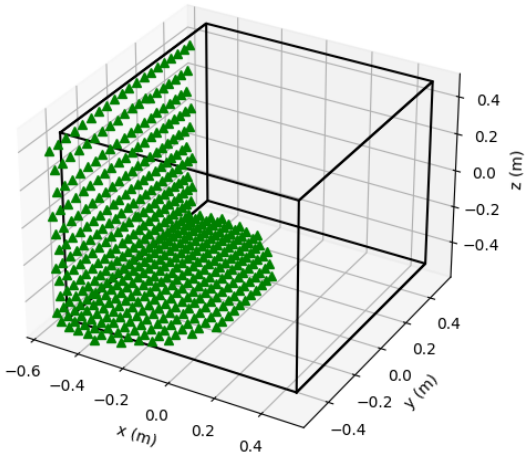

Figure 16. Depth-shader plume interaction for spinning cube case

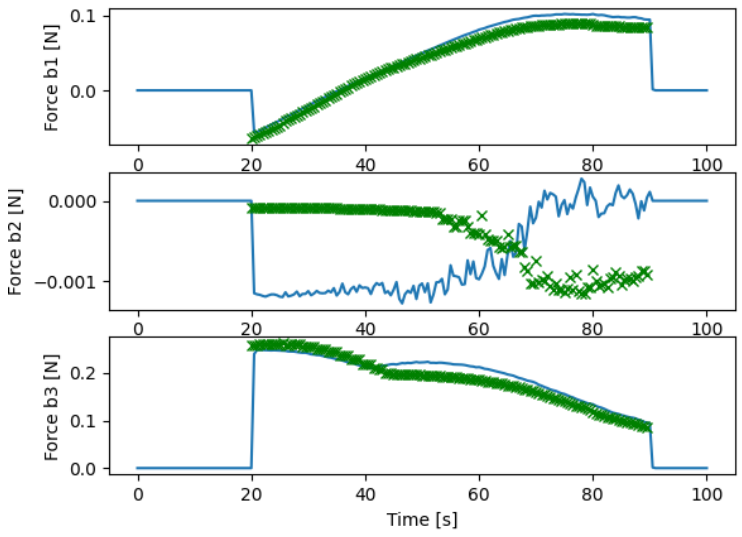

Figure 17. Depth-shader comparison with spiral plates for spinning cube case

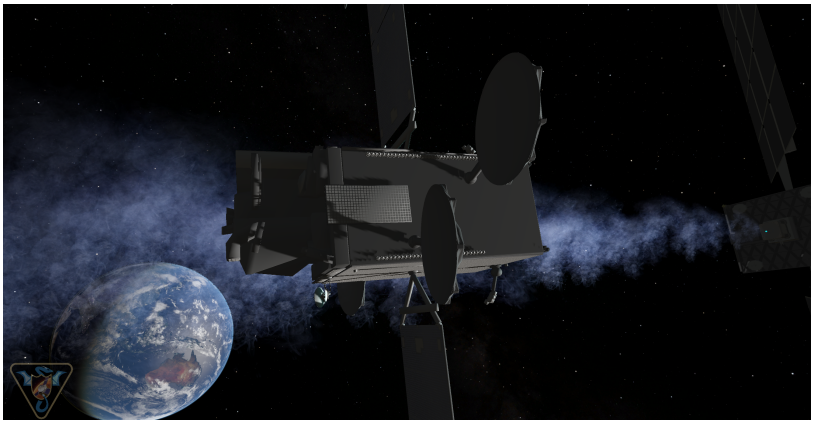

Figure 18. Example geometry of the servicer spacecraft pluming a Loral bus

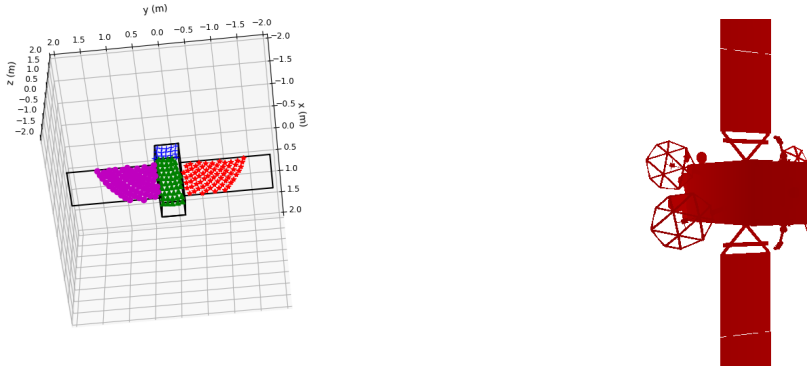

Figure 19. Plume mesh intersection with Loral Bus plate/shader surface during thrusting

COMPARISON OF COMPLEX SPACECRAFT GEOMETRIES

After demonstrating consistent performance with simple cube geometry, the complexity of the plumed geometry is increased by including a model of an existing real-world spacecraft design from NASA's 3D model repository in [12]. It is a 3D model for the Loral 1300 bus, which is a bus-type used primarily for geostationary communications satellites. This model was imported into Vizard via the included import utility. Figure 18 shows this 3D model imported into Vizard along with the servicer's plume impacting the Loral vehicle. The geometry of this model was also approximated as finite plates per the other approach, with the spacecraft bus sides as flat plates and the spacecraft solar arrays as single flat plates as well.

The lengths of the primary sides of the spacecraft bus, the dimensions of the solar arrays, and their relative locations were implemented into the Basilisk simulation plate model. The resulting plate model is deliberately coarse compared to the geometry seen in Figure 18, and was done to compare an 8-plate bus plate model against the depth-shader approach that is using the full vehicle geometry. The two spacecraft were placed into the same relative geometry as the spinning cube case from above with the thruster activated at the same time as that case as well. Figure 19 shows the plume impacting the simplified plate model next to the depth-shader image for the same run.

As these plots show, for equivalent time, roughly equivalent surfaces of the spacecraft are seen

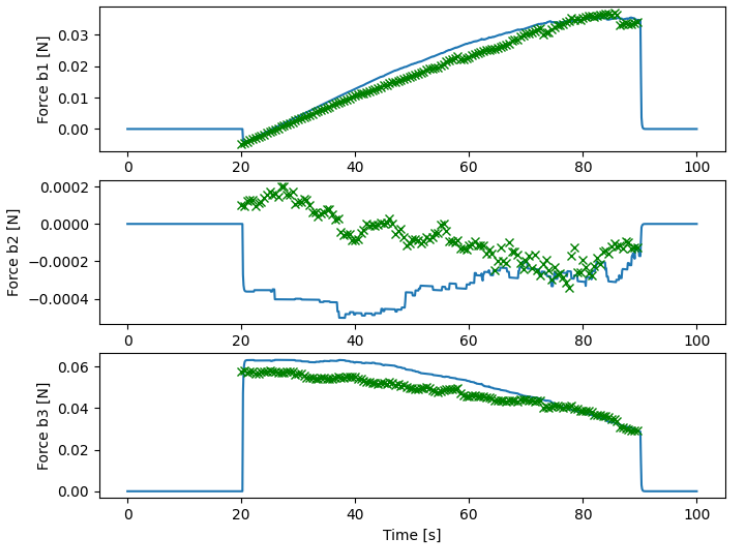

Figure 20. Depth-shader comparison with spiral plates for complex Loral Bus case

by the two different models. The Loral-Bus model has much more complex geometry with various antennae, some external thruster models, and solar array yokes/panels, but overall, the same surfaces in both models are providing similar bulk plume percentage absorption. Each piece of the depth-shader model could be individually encoded into the plate-model instantiation in Basilisk, but at significant time investment on the part of the person creating the model. Future work could evaluate if the 3D CAD model could be batch-processed in order to automatically generate the plate topology.

With the differences in implementation for this model, the differences between the computed forces are expected to be significantly higher. The benefit of the simple geometry lies in the speed of computation for simple surfaces and being able to perform all of the force/torque processing inside a single simulation model without having to connect to Vizard. But the downside of the simple model is that inserting arbitrary geometry is difficult and time-consuming and inter-plate shadowing cannot be handled by the model. Figure 20 shows a comparison of the forces computed by the spiral plate model against the depth-shader model for this complex geometry.

CONCLUSIONS

This paper has described two new methods for simulating thruster plumes emanating from one spacecraft and impacting a seconds spacecraft. The first technique uses rectangular plate geometry to assemble spacecraft structures of arbitrary complexity. Doing this manually for complex spacecraft geometry is time-intensive, but can likely be automated in the future. The thruster plumes themselves are modeled as Archimedean spirals and projected down to the surface of each plate model and then each point along the spiral is checked for intersection with each plate model. The force is then discretized across the surface of the spiral and the total force/torque from the discretized thruster plume is summed across all mesh points that intersect with each plate. Composite plumes can also be visualized across the mesh points to see which plates are being plumed the most

for a given geometrical setup.

When connected to the Vizard spacecraft visualization system, it is possible to use Vizard's depth-buffer-camera model to perform a high-fidelity representation of what the rays of the plume will intersect with on the spacecraft structure. Vizard allows the user to integrate CAD geometry of their spacecraft directly into the application, and then Basilisk can pass it the relative geometry of the objects and receive back a depth-shader image of the client in the FOV of the thruster. This is much less effort for the user of the system, provides shadowing analysis directly, and renders the most accurate output of the plume mesh of the options. The one downside to this method is that the connection of Basilisk to Vizard with the image traffic flowing back to Basilisk is computationally expensive.

Using these methods, strategies for plume detumbling guidance and control can be developed and assessed. An example of one of these is to hold a constant offset compared to the client spacecraft and to aim the servicer plume such that it will tend to decrease the angular velocity of the client as was shown with the angular velocity of the spinning client. While these strategies were only briefly discussed in this work, the models described here will provide an excellent basis for establishing the costs and capabilities of plume detumbling strategies.

Both finite plate modeling and the use of a 3D visualization tool have been shown to be very effective in analyzing the impact of thruster plumes on relative dynamics between two spacecraft. Both techniques can be used directly to look at plume self-impingement geometries like those discussed in [1] for single spacecraft whose thrusters impact themselves. They can also be used to model the probability of a given thruster firing impacting critical structure or optical services on the remote spacecraft for approach cases to delicate structures like the ISS. And finally, these can also be included for the case where two active spacecraft are maneuvering around each other and there isn't a true servicer or client, it is simply different instantiations of the models on the two different vehicles.

REFERENCES

[1] Teil, T., Schaub, H. Force and Torque Disturbance Modeling Due to General Thruster Plume Impingement, *68th International Astronautical Congress, Adelaide, Australia*, IAC-17,C1,1,2,x38510 (2017).

[2] Fox, K., Steagall, C., Usher, T., Deal, A., Worthy, E. International Space Station Bipropellant Plume Contamination Model update for short thruster pulse widths, *International Symposium for Materials in the Space Environment (ISMSE15)*, NAS15-10000 (2022).

[3] Rochelle, W., Reid, E., Terry, C., Smith, R., Lumpkin, F., Thermal Analysis for Orbiter and ISS Plume Impingement on International Space Station, *35th AIAA Thermophysics Conference: Plume Effects on ISS*, JSC-CN-6776 (2001).

[4] Tevepaugh, J., Penny, M., Space Shuttle Plume and Plume Impingement Study, *Final Report, Contract NAS9-14845*, Lockheed Missiles and Space Company report.

[5] Yim, J., Sibe, F., Ierardo, N., Plume Impingement Analysis for the European Service Module Propulsion System, *AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, GRC-E-DAA-TN16742, 2014.

[6] Penny, M., Wojciechowski, C., Prozan, R.SPACE SHUTTLE VEHICLE ROCKET PLUME IMPINGEMENT STUDY FOR SEPARATION ANALYSIS, *Final Report, Contract NAS9-11156*, Lockheed Missiles and Space Company report, 1971.

[7] Clark, C., Ricard, M., Hastings, D., Masterson, R. Comparative Assessment of Target Capture Techniques for Space Debris Removal with CubeSats.

[8] Nakajima, Y., Mitani, S., Tani, H., Murakami, N., Yamamoto, T., Yamanaka, K. Detumbling Space Debris via Thruster Plume Impingement, *AIAA/AAS Astrodynamica Specialist Conference*, 10.2514/6.2016-5660, September 2016

[9] Liu, Y., Cai, G. Detumbling a Non-Cooperative Tumbling Target Using a Low-Thrust Device, *AIAA Journal*, DOI:10.2514/1.J06098, May 2022.

[10] Nakajima, Y., Tani, H., Murakami, N., Mitani, S., Yamanaka, K. Contactless Space Debris Detumbling: A Database Approach Based on Computational Fluid Dynamics. *Journal of Guidance, Control, and Dynamics*, DOI: 10.2514/1.G003451, September 2018.

[11] J. Dennis Lawrence. *A Catalog of Special Plane Curves.* Dover Publications; First Edition, 2014.

[12] Meaney, C. "NASA 3D Resources, Loral-1300 Bus." Last modified November 6 2014. Loral 1300 bus

[13] C. Bombardelli and J. Peláez, "Ion Beam Shepherd for Contactless Space Debris Removal ", Journal of Guidance, Control, and Dynamics, Vol. 34, No. 3, May–June 2011, pp 916–920