Experiments on Autonomous Spacecraft Rendezvous and Docking Using an Adaptive Artificial Potential Field Approach

Zappulla, Richard II

http://hdl.handle.net/10945/50864
EXPERIMENTS ON AUTONOMOUS SPACECRAFT RENDEZVOUS AND DOCKING USING AN ADAPTIVE ARTIFICIAL POTENTIAL FIELD APPROACH

Richard Zappulla II,* Hyeongjun Park,† Josep Virgili-Llop† and Marcello Romano‡

Numerous missions over the past decades have pushed the state-of-the-art in autonomous rendezvous and proximity operations (RPO). The paramount requirement for the various guidance algorithms performing RPO is obstacle avoidance. The Artificial Potential Function (APF) method is one such method that provides robust obstacle avoidance while attempting to complete RPO objectives. However, inherent to its formulation, it is not optimal; as such, an Adaptive Artificial Potential Function (AAPF) method has been developed in an effort to reduce fuel consumption while still providing effective and flexible obstacle avoidance that is offered by traditional (APF) guidance methods. In this paper, the APF and AAPF guidance methods are developed from a theoretical standpoint and experimentally tested in a RPO-like environment in order to validate previous simulations. The experiments are performed using the Spacecraft Robotics Laboratory (SRL) Floating Spacecraft Simulator (FSS) test bed. The FSS test bed consists of a highly planar, polished, 15-ton granite-monolith, atop which spacecraft simulators float on approximately five microns of compressed air. Lastly, implementation considerations and experimental results are discussed.

INTRODUCTION

Over the past decade, numerous missions have been proposed and attempted exploring various techniques for safely conducting rendezvous and proximity operations (RPO), such as, the Air Force Research Laboratory (AFRL) XSS-10 and XSS-11 missions;1, 2 the National Aeronautics and Space Administration (NASA) Demonstration of Autonomous Rendezvous Technology (DART) mission;3 the Defence Advanced Research Projects Agency (DARPA) Orbital Express, SUMO/FREND, and Phoenix missions;4–6 the Swedish Space Corporation PRISMA mission;7 and the Georgia Institute of Technology Prox-1 mission.8, 9 Regardless of the mission type, attention was given to ensuring safe operations in the vicinity of other spacecraft. Resultantly, to achieve autonomous RPO, any guidance algorithm utilized must provide real-time collision avoidance in order to ensure collision-free operations.

One such method guaranteeing real-time onboard execution and collision-free RPO is an artificial potential function (APF) based guidance scheme.10, 11 The APF method is guaranteed to achieve real-time execution since it analytically obtains control inputs given the current navigation solution. Onboard passive or active sensors can be used for relative navigation, thus allowing the APF
guidance method to achieve real-time obstacle avoidance. It has been successfully implemented for autonomous RPO in either simulated or experimental environments. An adaptive artificial potential function (AAPF) method has been developed in order to improve APF and reduce both control effort and fuel usage. The AAPF method was shown by Muñoz to achieve improved performance in maneuvering time and propellant usage when compared to the APF method via Monte-Carlo simulations. It is worthwhile to note that both methods are powerful tools for path-constrained problems where the shape of a target platform is complicated with both static and dynamic obstacles, such as solar panels, antennae, and possibly robotic manipulators.

Motivated by the effectiveness of the AAPF method in simulation, this paper focuses on experimental verification by implementing and comparing the APF and AAPF real-time guidance algorithms on the Floating Spacecraft Simulator (FSS) test-bed illustrated in Figure 1. For this experiment, a Chaser FSS will be utilized to rendezvous and dock with a Deputy FSS that includes appendages attached to its body. The resulting rendezvous and proximity maneuvers include constraints on the Chaser FSS to perform the approach to docking via a Line-of-Sight (LoS) cone and keep out of the exclusion zone created by the operational range of the appendage. The experimental campaign will consist of varying the initial conditions of the Chaser FSS as well as the positions of a various numbers of obstacles to produce stressing cases for both the algorithms tested. Several comparison metrics will be derived from the Chaser FSS telemetry including rendezvous time, control effort, and constraint violation to compare the two guidance algorithms.

**Figure 1.** FSS Test Facility Illustrating the 4x4m Granite Monolith, Floating Spacecraft Simulators (FSS), and Overhead Motion Capture System (VICON) for Inertial Position and Attitude Information

**PROBLEM FORMULATION**

**System Dynamics**

In this work, the problem of autonomous rendezvous and docking of two spacecraft in the Clohessy-Wiltshire-Hill (CWH) framework is considered. The Deputy spacecraft is assumed to be in a circular orbit and not maneuvering while the Chaser spacecraft is maneuvering to dock with
the Deputy spacecraft. The resulting relative motion between the Deputy and Chaser is described by the CWH equations as,

\[ \ddot{x} - 3n^2 x - 2ny = \frac{F_x}{m} = u_x, \]
\[ \ddot{y} + 2n\dot{x} = \frac{F_y}{m} = u_y, \]
\[ \ddot{z} + n^2 \dot{z} = \frac{F_z}{m} = u_z, \]

where \( x \) is outwards along the inertial radius vector of the Deputy spacecraft, commonly referred to as the radial direction; \( z \) is in the direction of the orbital angular momentum vector of the Deputy spacecraft, commonly referred to as the cross-track direction; \( y \) completes the right-handed orthonormal basis vectors and is commonly referred to as the in-track direction; \( n \) is the mean motion of orbit of the Deputy; \( m \) is the mass of the Chaser spacecraft; and \( u_x, u_y, u_z \) are the acceleration components of the Chaser in the radial, in-track, and cross-track directions, respectively. This radial, in-track, cross-track coordinate frame is commonly referred to as the RSW frame and is illustrated in Figure 2. In the context of the FSS testbed, since the distance between the Deputy and Chaser is sufficiently small (less than 4 meters) over the time period of interest (100s of seconds), the relative motion of the Chaser can be approximated as a double integrator to a relatively high accuracy.17 Constraining the rendezvous and docking to only planar motion, the resulting equations of motion are given as,

\[ \ddot{x}_c = \frac{F_x}{m_c} = u_x, \]
\[ \ddot{y}_c = \frac{F_y}{m_c} = u_y, \]
\[ \ddot{\theta}_c = \frac{\tau_z}{I_z} = u_z, \]

where \( \ddot{x}_c, \ddot{y}_c \) are the inertial Chaser accelerations; \( \dot{\theta}_c \) is the angle of the Chaser in the inertial frame; \( F_x, F_y \) are the thrust components in the inertial \( x \) and \( y \) directions respectively; \( \tau_z \) is the torque.
along the z-axis, which is orthogonal to the x-y-plane; and $m_c$ and $I_z$ are the mass and moment of inertia of the Chaser, respectively.

The Deputy and Chaser FSS are illustrated atop the polished granite surface in Figure 3.

![Figure 3. Image of the Deputy and Chaser FSS Atop the Polished Granite Monolith](image)

**Deputy Obstacle Constraint**

As illustrated in Figure 3, the shape of the Deputy can be time-variant due to the presence of a multi-link robotic manipulator attached to the body of the Deputy. Resultantly, the rendezvous and docking between the Chaser and Deputy with the manipulator becomes a path-constrained problem. To deal with the non-symmetric obstruction caused by the manipulator and minimize the area encompassed by the obstacle constraint boundary, a cardioid-like function consisting of three semi-ellipses anchored at the Deputy center of mass (CoM) is utilized to form the obstacle constraint encompassing the Deputy FSS. The obstacle constraint is expressed as

$$
\begin{align*}
    d &= \begin{cases} 
        \frac{2a_1b_1^2 \cos \alpha}{b_1^2 \cos^2 \alpha + a_1^2 \sin^2 \alpha} & \text{if } \alpha \in \left[0, \frac{\pi}{2}\right) \\
        \frac{-2a_2b_2^2 \cos \alpha}{b_2^2 \cos^2 \alpha + a_2^2 \sin^2 \alpha} & \text{if } \alpha \in \left[\frac{\pi}{2}, \pi\right) \\
        \frac{2a_2b_3}{2a_1b_3} & \text{if } \alpha \in \left[\pi, \frac{3\pi}{2}\right) \\
        \frac{\sqrt{b_3^4 \cos^2 \alpha + 4a_2^2 \sin^2 \alpha}}{2a_1b_3} & \text{if } \alpha \in \left[\frac{3\pi}{2}, 2\pi\right) \\
    \end{cases}
\end{align*}$$

(3)

where $\alpha$ is the angle between the -Y-axis of the Deputy coordinate frame and a point along the obstacle constraint; $a_1, b_1$ are the respective semi-major and semi-minor axis lengths of the semi-ellipse defined in the region $\alpha \in \left[0, \frac{\pi}{2}\right)$; $a_2, b_2$ are the respective semi-major and semi-minor axis lengths of the semi-ellipse defined in the region $\alpha \in \left[\frac{\pi}{2}, \pi\right)$; and $b_3$ is the semi-minor axis of the semi-ellipse defined in the region $\alpha \in \left[\pi, 2\pi\right)$. These parameters are illustrated in Figure 4.

Since ellipses I & II are tangent to the docking axis and encompass the docking cone, the Chaser will complete the terminal approach to docking inside the docking cone and along the docking axis.
Figure 4. Visualization of the Obstacle Constraint Function Parameters in the Deputy Coordinate Frame

Note, the obstacle constraint encompasses a significant area of the usable workspace on the FSS testbed, as illustrated in Figure 5. This constraint and how it is dealt with is further discussed in the Experiment Test Cases section.

OVERVIEW OF ARTIFICIAL POTENTIAL FUNCTIONS

Artificial Potential Function (APF) Method

Overview The APF method utilizes the gradient of a potential field composed of both attractive and repulsive potentials to derive the necessary control inputs to reach the desired or goal position. The attractive potential not only establishes a global minimum at the goal (or target) state (i.e. 0),
but also aids in driving the Chaser state, \( x_c = [x_c, y_c, \theta_c]^T \) to the goal state, \( x_t = [x_t, y_t, \theta_t]^T \). The repulsive potential creates an area of higher potential in area(s) of the workspace that should be avoided, such as obstacles or exclusion-zones. The total potential function, \( \phi_{\text{tot}} \), is defined as the superposition of the sum of the attractive potential and all repulsive potentials in the workspace,\(^{10,14,18}\)

\[
\phi_{\text{tot}} = \phi_a + \sum_{i=0}^{N_o} \phi_{r_i} \tag{4}
\]

where \( N_o \) is the number of obstacles and \( \phi_a \) is the attractive potential and is defined by the quadratic function,

\[
\phi_a = \frac{k_a}{2} (x_c - x_t)^T Q_a (x_c - x_t) \tag{5}
\]

where \( k_a \) is a positive real number; \( Q_a \) is the symmetric, positive definite, attractive potential shaping matrix; and \( \phi_{r_i} \) is the \( i \)-th repulsive potential.

**Repulsive Potential Functions** The repulsive potential functions are divided into two groups: the Deputy obstacle boundary constraint and all other obstacles in the workspace. The repulsive potential function due to the Deputy obstacle boundary constraint is considered as the zeroth obstacle, while all other obstacles are assigned a number ranging from 1 to \( N_o \). The resulting repulsive function combining the Deputy obstacle boundary constraint and \( N_o \) generic obstacles is defined as,

\[
\phi_{r_i} = \begin{cases} 
\frac{1}{2} k_r \frac{(x_c - x_t)^T Q_{o_i} (x_c - x_t)}{Q_{o_i} (x_c - x_{o_i}) - 1} & \text{if } i = 1, 2, \ldots, n \\
\frac{1}{2} k_r \exp \left[ \frac{(x_c - x_t)^T Q_{d} (x_c - x_t)}{P_d (x_c - x_0)^T (x_c - x_0) - 1} \right] & \text{if } i = 0
\end{cases} \tag{6}
\]

where \( Q_{o_i} \) and \( P_{o_i} \) are the symmetric, positive definite generic obstacle repulsive potential shaping matrices; \( Q_d \) and \( P_d \) are the symmetric, positive definite Deputy boundary obstacle constraint shaping matrices; \( k_r \) is a positive constant; and \( (x_c - x_{o_i}) \) and \( (x_c - x_t) \) is the relative position of the chaser with respect to a generic obstacle and Deputy obstacle boundary constraint, respectively. From a geometric perspective, the shaping matrix \( Q \) aids in shaping the height of the potential function while the shaping matrix \( P \) aids in shaping the width of the potential function. The two shaping matrices are selected to ensure the resulting obstacle potential field encompasses the entirety of the obstacle such the Chaser cannot collide with the object.

**Control Law** In the far-field (hundreds to tens of meters from the goal position), the APF feedback controller can be defined as to provide an impulsive burn when the angle between the Chaser’s velocity vector and the desired velocity vector exceeds some defined threshold angle.\(^{8,10}\) While this coarse trajectory control may be adequate at long distances, finer control is desired as the Chaser approaches its goal position. Resultantly, a continuous-time APF control law is adopted\(^{8,10}\) and is defined as,

\[
u(x_c, x_c, x_t, x_1, x_o) = -k \left( \nabla \phi_{\text{tot}} - (\dot{x}_t - \dot{x}_c) \right) \tag{7}
\]

where \( k \) is a positive constant and the gradient of the total potential function,

\[
\nabla \phi_{\text{tot}} = k_a Q_a r_{ct} + k_r \exp \left[ 1 - r_{ch}^T P_{ch} r_{ch} \right] (Q_d r_{ct} - r_{ct}^T Q_d r_{ct} P_d r) \tag{8}
\]

\[
+ \sum_{i=1}^{N_o} k_r \left( r_{co_i}^T P_{o_i} r_{co_i} - 1 \right)^2 (Q_o r_{ct} - r_{ct}^T Q_o r_{ct} P_o r_{co_i}) \left( r_{co_i}^T P_o r_{co_i} - 1 \right)^2
\]
where \( \mathbf{r}_{ct} \) is the relative position of the Chaser with respect to the goal position; \( \mathbf{r}_{co_i} \) is the relative position of the Chaser with respect to the \( i \)-th obstacle; and \( \mathbf{r}_{cb} \) is the relative position of the Chaser with respect to the boundary as defined by Equation (3). Since the goal position is assumed to be stationary, the Equation (9) becomes

\[
\mathbf{u}(x_c, \dot{x}_c, x_t, x_o) = -k (\nabla x\phi_{tot} + \dot{x}_c) \tag{9}
\]

AAPF Method

Overview

The AAPF method has been developed in an effort to reduce fuel consumption while still providing effective and flexible obstacle avoidance. In doing so, the attractive potential shaping matrix, \( \mathbf{Q}_a \), is no longer time-invariant and is updated according to a derived adaptive law in order to follow a reference trajectory which is generated by the solution of a given optimal control problem (OCP).\(^{14, 15} \) It is worthwhile to note, in the case of the AAPF method, the reduction in control effort is bounded by the reference trajectory.

Adaptive Update Law Formulation

First consider the obstacle-free minimum fuel, fixed-time OCP for a double-integrator system whose solution is given as a bang-coast-bang type impulsive control history where,\(^{20} \)

\[
\mathbf{u}(t) = \begin{cases} 
\text{sign}(x_f - x_o) u \\
0 \\
-\text{sign}(x_f - x_o) u
\end{cases} 
\quad \text{for } t = t_0 \\
\forall t \in (t_0, t_f) \\
\text{for } t = t_f \tag{10}
\]

where the magnitude of the control is \( \|\mathbf{u}\| = \frac{\|x_f - x_o\|}{t_f - t_0} \). It is important to realize, this OCP formulation does not take into account obstacles and assumes that double-integrator system can proceed from its initial position to its final position unimpeded.

Despite this inherit limitation, this solution will form the basis of reference trajectory for the adaptive update law. Next, let the attractive potential shaping matrix be time-varying such that \( \mathbf{Q}_a(t) = \mathbf{R}(t)^T \mathbf{R}(t) \) enforces the symmetric positive definite condition imposed earlier. Since the translational and rotational motions are decoupled, it is beneficial to develop two independent adaptive laws in order to allow each adaptive law to update according to the speed of each set of dynamics. Consider first the translational motion of the Chaser; the upper-triangular matrix \( \mathbf{R}(t)^* \) is defined as,

\[
\mathbf{R}(t) = \begin{bmatrix} \rho_{11}(t) & \rho_{12}(t) \\
0 & \rho_{22}(t) \end{bmatrix} \tag{11}
\]

It is important to note since only the translation motion is being considered in this particular update law, the state \( x \) is composed of the respective \( x \) and \( y \) Cartesian coordinates. Next, the elements of the attractive potential shaping matrix are chosen in a manner which yields improved performance of the guidance algorithm using the same control law given in Equation (9). To do so, an error function is chosen to be the difference between a reference trajectory \( \dot{x}_{ref} = [v_{x,ref}, v_{y,ref}]^T \) and the negative gradient of the attractive potential,

\[
\mathbf{e}(x_c, \dot{x}_c, x_t) = \dot{x}_{ref} - (\nabla x\phi_a) \\
= -v_0 \frac{x_c - x_t}{\|x_c - x_t\|} + k_a \mathbf{R}(x_c - x_t) \tag{12}
\]

\(^{1} \)For conciseness, "(t)" will be dropped from further notation.
where \( v_0 \) is the velocity the Chaser is to maintain along the negative relative position unit vector in order to reach the goal position. This value is chosen a priori. The choice of \( v_0 \) effective acts as a speed-limiter for the Chaser and should be chosen such that the resultant velocity is within the capability of the system. The time derivative of the error function defined in Equation (12) for a stationary goal position is,

\[
\dot{e} = -v_0 \left( \frac{x_c - x_i}{\|x_c - x_i\|} - \frac{(x_c - x_i)(x_c - x_i)^T x_c}{\|x_c - x_i\|^3} \right) + k_a \left( R^T \dot{R} + R^T \dot{R} \right) \hat{x}_c \tag{13}
\]

Letting \( \rho = [\rho_{11}, \rho_{12}, \rho_{22}, \rho_{33}]^T \), the attractive potential gradient portion of Equation (13) can be parameterized as

\[
k_a \left( R^T \dot{R} + R^T \dot{R} \right) \hat{x}_c = k_a \mathbf{K} \hat{\rho}
\]

where \( \mathbf{K} \) is defined as

\[
\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & 0 \\ k_{21} & k_{22} & k_{23} \end{bmatrix}
\]

and whose elements are

\[
k_{11} = 2\rho_{11} \dot{x} + \rho_{12} \dot{y}, \quad k_{12} = \rho_{11} \dot{y}, \quad k_{21} = \rho_{12} \dot{x}, \quad k_{22} = \rho_{11} \dot{x} + 2\rho_{12} \dot{y}, \quad k_{23} = 2\rho_{22} \dot{y},
\]

where \( \dot{x} = x_c - x_i \) and \( \dot{y} = y_c - y_i \), respectively.

In order for the error to converge to zero asymptotically, it is desired to have

\[
\dot{e} = -k_e
\]

where \( k \) is strictly positive. This implies an adaptive update law of

\[
\dot{\rho} = \mathbf{K}^T (\mathbf{K} \mathbf{K}^T)^{-1} \left[ v_0 \left( \frac{x_c - x_i}{\|x_c - x_i\|} - \frac{(x_c - x_i)(x_c - x_i)^T x_c}{\|x_c - x_i\|^3} \right) - k_a \mathbf{R}^T \mathbf{R} \hat{x}_c - k e \right] \tag{16}
\]

\( \mathbf{K} \) becomes singular when \( x_c \rightarrow x_i \).

Following a the same procedure, an adaptive update law for the rotational motion of the Chaser can be derived. Since the Chaser rotational motion is constrained to a single axis of rotation, \( R = \rho_{33} \). The resultant rotational motion error function and adaptive update law are

\[
e_{\theta}(\theta_c, \dot{\theta}_c, \dot{\theta}_t) = \dot{\theta}_{ref} - (\nabla_{\theta} \phi_{\theta}) = -\omega_0 \frac{\theta_c - \theta_t}{\|\theta_c - \theta_t\|} + k_{\theta} \rho_{33}^2 (\theta_c - \theta_t) \tag{17}
\]

\[
\dot{\rho}_{33} = \frac{\omega_0 \left( \|\theta_c - \theta_t\|^{-1} - (\theta_c - \theta_t)^2 \|\theta_c - \theta_t\|^{-3} \right) - k_{\theta} \rho_{33}^2 (\theta_c - \theta_t) - k e \theta}{2k_{\theta} \rho_{33} (\theta_c - \theta_t)} \tag{18}
\]

Analogous to the translational update law, the time derivative of \( \rho_{33} \) becomes undefined as \( \theta_c \rightarrow \theta_t \).

**Implementation Considerations**  This section will address several implementation considerations for the AAPF method. First, in order to ensure the time-varying attractive shaping parameters are bounded, a lower and upper bound are defined such the shaping parameter estimates form a convex set,

\[
\rho_i = \{\rho \in \mathbb{R}| \rho_i^- \leq \rho \leq \rho_i^+ \\forall i = 1, 2, \ldots, 6
\]

4468
If, after the shaping parameter propagation step, any element of $\rho$ is above or below $\rho^+$ or $\rho^-$ respectively, that element is set to its respective upper or lower bounded limit. Next, a threshold for zero is set such that if $|\rho_i| < \epsilon$, where $\epsilon$ is a small positive number, than $\rho_i = \text{sign}(\dot{\rho}_i) \epsilon$. This step, while creating a discontinuity, preserves the direction of the update law while providing several benefits. First, it avoids a divide-by-zero or singularity condition which may introduce numerical instabilities into the algorithm. Secondly, it ensures that no shaping parameter estimate converges to a local minimum around zero. However, as a result of this zero-thresholding, any parameter which is being driven to zero by the update law will chatter around $\pm \epsilon$ and never go to zero. Lastly, it is worthwhile note that due to the singularity and divide-by-zero conditions that arise when the Chaser states reach the desired goal states, the AAPF method may not be well suited as a final approach method. That is, as a result of this conditions, the AAPF method will attempt to shape the attractive potential such that it maintains the specified translational and angular speeds until some docked condition is met. By bounding the shaping parameter estimates, it becomes possible to utilize the AAPF method as an approach and terminal guidance method. In this work, the bounds of the shaping parameter matrix are chosen such that reasonable docking speeds are maintained.

**EXPERIMENTAL SETUP & RESULTS**

**Overview**

The experiments were conducted utilizing two Floating Spacecraft Simulators (FSS) which float atop approximately five microns of compressed air over a 4-by-4 meter polished granite monolith surface allowing for quasi-frictionless translational and rotational motion. The 15 ton granite monolith surface is has a horizontal leveling accuracy of less than 0.01° with a planar accuracy of $\pm 0.0127\text{mm}$ ($\pm 0.0005$ in). Each FSS is propelled via eight thrusters supplied from an onboard composite compressed air tank. Additionally, the onboard power system, sensors, and computer allow each FSS to perform real-time in-situ computations. Navigation data is provided by an overhead VICON motion capture system providing position and attitude and augmenting by an onboard single-axis fiber-optic gyroscope (FOG). Intra-test bed communication is performed via TCP/UDP communication over an ad-hoc WiFi network. The FSS test bed communications architecture is illustrated in Figure 6.

**Hardware & Software Setup**

*Floating Spacecraft Simulator* The FSS floats over the polished granite surface using three air-pads mounted on ball bearings that expel compressed air through a porous carbon media. Figure 7 shows the three air-pads mounted at the base of the FSS. Airflow to the pads is controlled by a pressure regulator and solenoid valve. A particulate air filter located before the air-pads prevents damage to the porous material from foreign debris in the air. The quasi-frictionless environment provided by the air-pads over the surface of the polished granite monolith provides an analogous kinematic and dynamic environment for which to test close proximity operations. The FSS endurance is limited by the maximum amount of propellant and typically reaches 15 minutes. Heavy usage of the thrusters can greatly reduce the floating time. The eight thrusters can be switch on and off independently by the onboard computer creating the forces and torques that propel the FSS over the granite surface. An additional pressure regulator reduces the tank outlet pressure and feeds the eight cold-gas thrusters. Each thruster is equipped with its own fast response solenoid valve. The air is then routed to custom made supersonic convergent nozzles. Each thruster produces around
16 mN of thrust when the air pressure is 60 psi. Different thrust levels can be achieved by adjusting the pressure regulator outlet pressure.

The electronic systems are powered using two 95 Wh rechargeable lithium-ion batteries and a battery management module that regulates the battery power and provides a charging port. A PC-104 form-factor computer based on an Intel Atom 1.6 GHz 32 bit processor and includes 2 GB of ram and a 8 GB solid-state drive (SSD) provides all the required computing. A WiFi module allows the onboard computer to exchange data over a TCP/UDP ad-hoc WiFi network. Additionally a relay board connected as a PC-104 expansion board provides the solenoid valves for both air-pad and thruster switching capabilities. A DSP-3000 Fiber Optic Gyro connected via the onboard computer serial ports provides angular rate readings at a 100 Hz. Other peripherals (as stereo-cameras) can be connected but they have not been used for the experiments.
All of this equipment is housed inside of a structure composed of a carbon fiber reinforced polymer base plate. Four 1010-Al columns provide the primary structure for the FSS. The equipment mounting points and outer shell are fabricated from polycarbonate using additive manufacturing, providing a modular and easy to adapt structure where to mount the different hardware components.

Although the FSS are autonomous some external hardware is required. In particular the overhead VICON positioning system and an air compressor used to refill the FSS air tanks are used. The VICON system is composed of ten overhead cameras that track circular reflectors mounted on the FSS. The VICON server processes this information and combined with the VICON Tracker software running on an external PC provide position and attitude updates at a 100 Hz. Using the VICON SDK for MATLAB the position and attitude updates are received in a MATLAB script and sent without further processing to the FSS via TCP/UDP over the ad-hoc wifi network.

Floating Spacecraft Software  A RTAI-patched Ubuntu 14.04 Server Edition operating system is used to provide multi-rate real-time execution capabilities for the FSS. Prior to deployment to the FSS for testing, guidance, navigation and control (GN&C) algorithms are developed in a MATLAB-Simulink simulated environment using a custom library containing both FSS-standardized navigation and control blocks in addition to both simulated and actual interfaces with the various FSS sensors. Next, the GN&C algorithms are transitioned to the MATLAB-Simulink hardware models where they are then cross-compiled for the FSS 32-bit (i686) architecture and later transferred to the robot via secure copy (scp) to the FSS SSD prior to execution.

The real-time FSS software architecture is illustrated in Figure 8. First, the navigation block receives the data from the VICON motion capture system as well as the Fiber Optic Gyro data. Next, the state of each thruster solenoid value is used to estimate the force from each thruster and is fed along with the navigation data into a Discrete Kalman Filter (DKF). The DKF then provides a state estimate of the FSS (position, attitude, and velocities) on the granite surface.

![Figure 8. General FSS Software Architecture](image)

Given data from both the VICON and FOG is provided asynchronously from their respective sensors and the filter can run at some scenario-specified rate, dropped or corrupted samples are inevitable. To cope with the first reality of dropped samples, the filter was designed such that it can individually process updates from either the VICON or the FOG sensor independently. To cope with the second reality of corrupted samples, the filter utilizes \( \chi^2 \)-gating for measurement association to reject measurements that are not within a 99.95% likelihood of being associated with the measurement.\(^{23} \)
Lastly, the control block converts the inputs from the guidance (forces and torques) into commands for the thrusters' solenoid valves. Requested forces and torques are mapped to the thrusters and then passed through a delta-sigma modulator to generate the firing impulses. Solenoid valves' minimum firing times are respected in the delta-sigma modulator. Based on the solenoid valves' states, forces and torques are estimated and then passed back to the DKF.

A telemetry block is then used to send FSS telemetry streams to an external PC for debugging and logging purposes.

**Coordinate Frame Definition & Assumptions**

**Body-Fixed Frame Definition** The Chaser body-frame dextral orthonormal basis vectors \( \{ \hat{X}_C, \hat{Y}_C, \hat{Z}_C \} \) are anchored at the geometric center of the Chaser FSS body; the positive X-axis is normal to and through the docking cone; the positive Z-axis is normal to and through the top of the FSS; and the positive Y-axis via right-hand rule (RHR). Likewise, the Deputy body-frame dextral orthonormal basis vectors \( \{ \hat{X}_D, \hat{Y}_D, \hat{Z}_D \} \) are anchored at the geometric center of the Deputy FSS body with the positive X-axis is normal to and through the docking cone; positive Z-axis is normal to and through the top of the FSS; and the positive Y-axis via RHR. It is worthwhile to note the coordinate frames utilized by the overhead VICON system do not exactly match the FSS body-fixed frames illustrated in Figure 9. These differences are taken into account when defining the target position and orientation of the Chaser.

![Coordinate Frame Definition](image)

**Figure 9. Coordinate Frame Definition for the (a)Chaser and (b)Deputy FSS**

**Assumptions** In order to describe the Deputy obstacle constraint via a non-symmetric piece-wise cardioid function, it is assumed the robotic manipulator attached to the Deputy +Y-axis has a limited range of motion whose bounds are illustrated in Figure 9(b). Additionally, it is also assumed the obstacle constraint encompassing the Deputy is only considered violated if the center of mass (CoM) of the Chaser crosses this boundary. Likewise, the obstacle constraints are considered to be violated if the CoM of the Chaser crosses the boundary.
Test Cases

Test Case Definition  The experimental test cases utilized were designed to maximize the usable workspace of the FSS test bed while minimizing the number of test sets to achieve a comparative data set. Due to the large area enclosed by the Deputy obstacle constraint as illustrated in Figure 5, the Deputy was placed in a corner of the FSS test bed facing outward as illustrated in Figure 10. This position provides a sufficient tradeoff between increasing the usable workspace while still having the Chaser to consider a significant portion of the Deputy obstacle constraint and demonstrate obstacle avoidance. The remaining three corners of the FSS test bed were chosen to be the starting points to test each guidance algorithms against a given obstacle field. The initial attitude of the Chaser FSS is fixed at each starting point such that its docking axis is parallel to the +X-axis of the FSS test bed. These standard set of initial conditions allows for the placement of obstacles such that the guidance algorithms can be tested systematically and allowing for greater comparison between each test set and algorithm. Lastly, all tunable guidance parameters are held constant for each test case, further enabling greater comparison. These parameters used for both the APF and AAPF guidance algorithm are listed in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{ii}, k_{rr}$</td>
<td>1</td>
</tr>
<tr>
<td>$Q_{a}$</td>
<td>diag(0.0293, 0.0293, 0.125)</td>
</tr>
<tr>
<td>$Q_{d}$</td>
<td>diag(0.0625, 0.0625, 0.125)</td>
</tr>
<tr>
<td>$P_{d}$</td>
<td>diag(15, 15, 0)</td>
</tr>
<tr>
<td>$v_0$</td>
<td>0.040 m/s</td>
</tr>
<tr>
<td>$\omega_0$</td>
<td>0.030 rad/s</td>
</tr>
<tr>
<td>$\rho_0$</td>
<td>$[0.1712, 0, 0, 0.1712, 0, 0.3536]^T$</td>
</tr>
<tr>
<td>Bounds on $\rho$</td>
<td>$\rho^+ = 0.75 \forall \rho_i$</td>
</tr>
</tbody>
</table>

Figure 10. Schematic of Experiment Test Cases Illustrating Chaser Initial Conditions & Obstacles
**Metrics**  The APF and AAPF comparison will be assisted through the use of the following relevant metrics: rendezvous time, control effort, and constraint violation.

First, the rendezvous time is defined as the elapsed time between enabling of the FSS guidance, $t_0$, and the docked conditions are met and the guidance is disabled, $t_f$. A 'Guidance Enabled' flag is sent back from the Chaser FSS to a ground station computer via TCPs/UDP to enable its measurement with a resolution of 0.01 seconds.

Next, the control effort is defined using the $\ell_1$-norm as

$$\text{Control Effort} = \int_{t_0}^{t_f} \sum_{i=1}^{8} u_i dt$$

(20)

where $u_i$ is the solenoid valve state for each thruster, $t_0$ is the time the FSS guidance is enabled, and $t_f$ is the time the docked conditions are met and the FSS guidance is disabled. The thruster solenoid states are sent back to a ground station computer with a resolution of 0.01 seconds.

Lastly, constraint violations are computed in real-time onboard the FSS every guidance cycle, 0.1 seconds and sent back to the ground station computer as a 0- or 1-flag. A constraint violation is determined to have occurred when the distance between the Chaser FSS CoM and an obstacle is determined to be less than a certain threshold. For all the obstacles, this threshold is 0.3 meters, the threshold for the Deputy obstacle constraint is variable and is defined by the non-symmetric cardioid function.

**Results & Discussion**

**Results** The Chaser FSS trajectories using both the APF and AAPF guidance methods for Test Set 1 and Test Set 2 are illustrated in Figure 11. It is worthwhile to note the Chaser FSS orientations illustrated in both figures are equally spaced in time. The metrics for APF and AAPF method for Test Set 1 and Test Set 2 are listed in Table 2 and Table 3 respectively.

<table>
<thead>
<tr>
<th>Initial Condition</th>
<th>Guidance Algorithm</th>
<th>Control Effort (seconds)</th>
<th>Rendezvous Time (seconds)</th>
<th>Constraint Violation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>APF</td>
<td>50.66</td>
<td>142.30</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>AAPF</td>
<td>40.32</td>
<td>85.00</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>APF</td>
<td>97.14</td>
<td>152.70</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>AAPF</td>
<td>64.30</td>
<td>145.90</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>APF</td>
<td>68.08</td>
<td>231.8</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>AAPF</td>
<td>54.45</td>
<td>119.00</td>
<td>No</td>
</tr>
</tbody>
</table>

**Discussion** Table 4 lists the percent difference in the control effort and rendezvous time between the APF and AAPF methods. The AAPF method was observed to reduce the required control effort measure on average by 26.90%, while simultaneously decreasing the average rendezvous time by approximately 14.11%. Discounting Set 2, initial condition 2, the average reduction in rendezvous time increases to just over 25%. Further summarizing the results, Test Set 1 was observed to have a control effort reduction of approximately 24.75% over the three initial conditions. Likewise, Test Set 2 was observed to have a control effort reduction of approximately 29.06%. Furthermore, it is interesting to note in both Test Sets, Initial Condition 2 was observed to have the largest reduction in
Table 3. Test Set 2 Experimental Metrics

<table>
<thead>
<tr>
<th>Initial Condition</th>
<th>Guidance Algorithm</th>
<th>Control Effort (seconds)</th>
<th>Rendezvous Time (seconds)</th>
<th>Constraint Violation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>APF</td>
<td>107.97</td>
<td>125.25</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>AAPF</td>
<td>84.91</td>
<td>175.80</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>APF</td>
<td>126.84</td>
<td>162.10</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>AAPF</td>
<td>76.96</td>
<td>134.30</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>APF</td>
<td>88.78</td>
<td>162.80</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>AAPF</td>
<td>65.26</td>
<td>139.20</td>
<td>No</td>
</tr>
</tbody>
</table>

Figure 11. Trajectories for Test Set 1 & 2 Using (a) APF Guidance and (b) AAPF Guidance
control effort. This is attributed to having the largest straight-line distance from the initial condition to the goal position.

**Table 4. Percent Difference Comparison Between the APF and AAPF Guidance Methods**

<table>
<thead>
<tr>
<th>Test Set</th>
<th>Initial Condition</th>
<th>Control Effort Reduction</th>
<th>Rendezvous Time Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>20.41%</td>
<td>40.27%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>33.81%</td>
<td>4.45%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>20.02%</td>
<td>48.66%</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>21.36%</td>
<td>-40.36%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>39.33%</td>
<td>17.15%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>26.49%</td>
<td>14.50%</td>
</tr>
</tbody>
</table>

Examining the trajectories illustrated in Figure 11, it is worthwhile to note the trajectory originating from Initial Condition 2 merges with one of the other two trajectories for the two Test Sets considered. For example, consider the Test Set 1 trajectory illustrated in Figure 11(b). The Initial Condition 2 trajectory merges with Initial Condition 3 trajectory after circumnavigating Obstacle 1. Likewise, in Test Set 2, the Initial Condition 2 trajectory merges with the Initial Condition 1 trajectory while it is in the process of circumnavigating Obstacle 1 for both the paths generated utilizing the APF and AAPF methods. This merging of trajectories is indicative of the inherent gradient-following of the two guidance methods.

An interesting behavior of the AAPF guidance algorithm is its tendency to slow down when in the vicinity of an obstacle. This behavior is demonstrated visually in Figure 11(d). In all three initial conditions, whenever the Chaser is in the vicinity of an obstacle, the translational velocity is small and then grows rapidly as the Chaser moves out of the influence of the obstacle. For example, consider Test Set 2, Initial Condition 1; as illustrated in Figure 12(b), the Chaser stays in the vicinity of Obstacle 2 for approximately 100 seconds before it begins to rapidly move toward the goal position. Furthermore, the shaping parameters are observed to not significantly change for the first 100 seconds. After 100 seconds, the diagonal terms, $\rho_{11}, \rho_{22}$, begin to change rapidly and the Chaser converges to the goal position, as illustrated in Figure 12. Depending on the relative position of the Chaser to the obstacle, this tendency can be attributed to either the formation of local minima (Test Set 2, Initial Condition 1) or dominance of the repulsive field of the obstacle in the guidance law (Test Set 2, Initial Condition 2).

**CONCLUSIONS**

In this paper, the Artificial Potential Function (APF) and Adaptive Artificial Potential Function (AAPF) guidance algorithms were implemented, tested, and experimental compared on the Floating Spacecraft Simulator (FSS) test bed in order to rendezvous and dock the Chaser FSS to the Deputy FSS. To do so, an adaptive update law was derived utilizing the solution to the minimum-fuel optimal control problem for a double integrator system as the reference trajectory. Next, a cardioid function was utilized to describe the non-symmetric boundary formed by the multi-link robotic manipulator attached to the Deputy FSS. While both methods proved to be powerful tools for real-time obstacle avoidance, the APPF method was able to achieve a reduction in control effort of upwards of 39.33%, with an average control effort reduction across all test cases of 26.09%.
Likewise, the AAPF method was observed to achieve a reduction in rendezvous time of 48.66%, with an average rendezvous time across all test cases 14.11%. However, it was observed that the AAPF guidance method exhibited a tendency to slow the velocity of the Chaser in the vicinity of the obstacles. Potential future work includes extending this method to deriving an update law that adapts the obstacle repulsive potential shaping parameters in addition to the attractive potential shaping parameters, well as implementing and comparing the efficacy of the AAPF method when utilizing reference trajectories generated by other methods.

REFERENCES


