A Prediction and Motion-Planning Scheme for Visually Guided Robotic Capturing of Free-Floating Tumbling Objects With Uncertain Dynamics

Farhad Aghili, *Senior Member, IEEE*

***Abstract*—Visually guided robotic capturing of a moving object often requires long-term prediction of the object motion not only for a smooth capture but because visual feedback may not be continually available, e.g., due to vision obstruction by the robotic arm, as well. This paper presents a combined prediction and motion-planning scheme for robotic capturing of a drifting and tumbling object with unknown dynamics using visual feedback. A Kalman filter estimates the states and a set of dynamics parameters of the object needed for long-term prediction of the motion from noisy measurements of a vision system. Subsequently, the estimated states, parameters, and predicted motion trajectories are used to plan the trajectory of the robot's end-effector to intercept a grapple fixture on the object with zero relative velocity (to avoid impact) in an optimal way. The optimal trajectory minimizes a cost function, which is a weighted linear sum of travel time, distance, cosine of a line-of-sight angle (object alignment for robotic grasping), and a penalty function acting as a constraint on acceleration magnitude. Experiments are presented to demonstrate the robot-motion planning scheme for autonomous grasping of a tumbling satellite. Two robotics manipulators are employed: One arm drifts and tumbles the mockup of a satellite, and the other arm that is equipped with a robotic hand tries to capture a grapple fixture on the satellite using the visual guidance system.**

***Index Terms*—Active orbital debris removal, capture of a tumbling satellite, capturing free-floating objects, de-orbiting space debris, on-orbit servicing, optimal guidance and control, optimal robot trajectory, robot-motion planning, robot vision, space robotics.**

NOMENCLATURE

$\boldsymbol{A}$	Rotation matrix.
a	Maximum acceleration.
$\boldsymbol{e}$	Feedback error.
$\boldsymbol{H}$	Observation sensitivity matrix.
$\boldsymbol{h}$	Observation vector.
$\boldsymbol{I}_c$	Inertia tensor of the target.
$I_{xx}, \ldots, I_{zz}$	Principal inertia components of the target.
$\boldsymbol{K}$	Kalman filter gain matrix.
$\boldsymbol{k}, \boldsymbol{k}'$	Unit vectors expressed in $\{\mathcal{B}\}$ and $\{\mathcal{A}\}$.
$\boldsymbol{P}$	Kalman filter covariance matrix.
$\boldsymbol{p}$	Vector containing the inertia ratios.
$\boldsymbol{Q}$	Covariance matrix of process noise.
$\boldsymbol{q}$	Target quaternion: rotation $\{\mathcal{B}\}$ w.r.t. $\{\mathcal{A}\}$.
$\boldsymbol{q}_r$	End-effector quaternion.
$\boldsymbol{R}$	Covariance matrix of measurement noise.
$\boldsymbol{r}$	Position of the target center of mass; expressed in $\{A\}$.
$\boldsymbol{r}_s$	Grapple fixture position expressed in $\{\mathcal{A}\}$.
$\boldsymbol{r}_r$	End-effector position expressed in $\{\mathcal{A}\}$.
t_f	Terminal time.
t_o	Initial time.
t_Δ	Sampling time.
$\boldsymbol{x}_s$	State vector of the target.
$\boldsymbol{x}$	State vector.
$\boldsymbol{u}$	Control input.
$\boldsymbol{v}$	Pose measurement noise.
$\boldsymbol{z}$	Measured pose.
$\mathcal{H}$	Hamiltonian.
λ	Costate.
$\boldsymbol{\mu}$	Quaternion: rotation $\{\mathcal{C}\}$ w.r.t. $\{\mathcal{B}\}$.
$\boldsymbol{\upsilon}$	Linear velocity of the target center of mass.
$\boldsymbol{\eta}$	Quaternion: rotation $\{\mathcal{C}\}$ w.r.t. $\{\mathcal{A}\}$.
$\boldsymbol{\omega}$	Angular velocity of target satellite.
$\boldsymbol{\rho}$	Location of the target center of mass; expressed in $\{\mathcal{B}\}$.
$\boldsymbol{\Phi}$	State transition matrix.
θ	Line-of-sight angle.
$\epsilon_{\rm th}$	Threshold.
$\boldsymbol{\epsilon}_\tau, \boldsymbol{\epsilon}_f$	Process noise.
$\mathbf{1}_n$	$n \times n$ identity matrix.
$[\cdot\times]$	Matrix form of cross-product.
$\otimes$	Quaternion product.

I. INTRODUCTION

SINCE mid-1990s, the paradigm of on-orbit servicing using a space manipulator has attracted many researchers [1]–[14]. These works were motivated by several national and international missions not only for repairing, rescuing, and refueling failed satellites [15], [16] but for removal of defunct satellites or space debris as well [17], [18]. Orbital debris removal using space manipulator is becoming of particular interest as space debris is on the rise and the population growth has reached an unstable point in some congested orbits due to risk of the collision of large debris [19]. Several studies that are

Manuscript received February 15, 2011; revised June 6, 2011 and September 2, 2011; accepted December 1, 2011. Date of publication January 5, 2012; date of current version June 1, 2012. This paper was recommended for publication by Associate Editor T. Murphey and Editor G. Oriolo upon evaluation of the reviewers' comments.

The author is with the Space Exploration of Canadian Space Agency, Saint-Hubert, QC J3Y 8Y9, Canada (e-mail: farhad.aghili@asc-csa.gc.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2011.2179581

conducted by national space agencies concluded that the space environment can be only preserved upon continual removal of large orbital debris objects, such as dead satellites [20], [21]. All these robotic servicing missions require that a robotic arm captures the client satellite in a safe and secure manner given several operational and environmental constraints. Since defunct satellites usually do not have a functional attitude control system (ACS), it is common for them to tumble. This is because the angular momentum that is stored in the gyros and wheels of the satellite starts migrating to its body as soon as failure occurs. Ground observations [17] also confirm that many space debris objects have indeed tumbling motion, making the robotic capturing a very challenging task. In fact, although several missions for free-flyer capture using a robotic arm on-orbit have been demonstrated so far [9], [15], [22], [23], [24], robotic capturing of a tumbling satellite has yet to be attempted. Robotic manipulators that are used in future on-orbit serving missions will be operated from ground or autonomously depending on mission constraints, requirements, and the level of technology readiness. However, in the presence of command latency, even a trained operator cannot successfully capture a free-flyer unless the angular rate is extremely low. Therefore, increased autonomy for robotic systems for on-orbit servicing missions is one of the key technology identified by many space agencies [25].

Fig. 1 illustrates a typical on-orbit satellite servicing operation where the servicer satellite caries a robotic arm (with a grappling device on it), which is driven by a vision system so as to capture a client satellite (target). The servicer satellite is under its own attitude controller, whereas the target is assumed to be a noncooperative satellite with uncertain dynamics. Clearly, only after the robot successfully captures and stabilizes the tumbling satellite, a repairing, rescuing, or de-orbiting operation can be started. Therefore, a common task for such on-orbit servicing consists two primitive robot operations: 1) pregrasping phase[1] and 2) postgrasping phase. In the pregrasping phase, the manipulator arm moves from its home position to intercept a grapple fixture on the target at a rendezvous point with zero relative velocity. In the postgrasping phase, the space manipulator has to bring the target satellite to rest in such a way that the interaction torque/force remains below a safe value [26]. This paper focus on a robot guidance scheme for pregrasping phase.

The capture will be without impact if the manipulator approaches the target in such a manner that, at the time of capture, the relative velocity between the end-effector and the target's grapple fixture is zero [27]. Otherwise, the detrimental effect of the impact on the free-floating space robot has to be taken into account [28]. In addition, the target orientation at the interception time has to be adequate such that the fixture becomes accessible for robotic grasping. Satisfying theses grasping requirements is not possible without prediction of the target motion so that the motion of the end-effector can be carefully planned and executed. Since external force/moment acting upon a free-floating object in space is small, the object motion can be predicted provided that the relevant dynamics parameters and the initial states are accurately known. However, inertia parameters of spacecraft are usually uncertain, because there is no practical way to measure remaining propellant fuel in zero gravity. Consequently, when the dynamics of the target spacecraft is not known beforehand, not only the linear and angular velocities of the target but also its inertia parameters must be accurately estimated in real time so that the servicer satellite can reliably predict the states of the target in future.

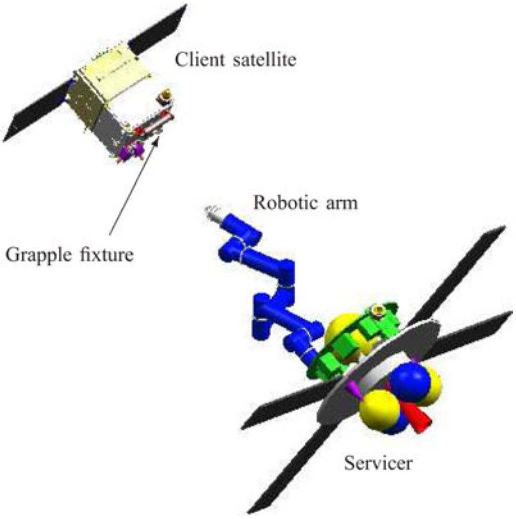

Fig. 1. Servicer satellite trying to capture a client satellite.

Controlling of a free-floating space robot, such that when the manipulator is controlled, the reaction of the manipulator motion on the manipulator base is nullified, has been established in the literature [29]–[32]. Although zero reaction maneuvers has been proven, particularly, useful [32], reaction wheels are also used to stabilize the base attitude [33]. In fact, it was shown that nearly any control schemes that can be used for fixed-base manipulators can also be implemented on free-floating space robot, provided that the platform attitude and position are estimated and then taken into account [27], [29], [31], [34]. Despite a variety of control strategies for retrieval and capturing a space object by a free-flying space manipulator [1], [2], [4], guidance of a robotic manipulator in order to rendezvous and capture a tumbling free-floating object [7] in a safe and secure manner still remains a challenging task. The majority of the techniques that are developed for the real-time motion planning and the robotic interception of a moving target assume a *point mass* target [12], [35]– [42]. Interception of a spinning or tumbling satellite is a more difficult task, however, due to the complex dynamics of rotating rigid bodies. For the rendezvous with a tumbling satellite in 2-D, a navigation technique was developed in [40]. Ma *et. al* [10] formulated and analyzed the rendezvous control problem of a servicing spacecraft for optimal approach to and alignment with a rotating rigid body with known dynamics. An iterative algorithm for a planar case is proposed to solve the optimal trajectory for minimum time/fuel consumption. Trajectory planning for robotic capturing of a tumbling satellite was presented in our earlier work [43], [44], without taking the requirements at the time of grasping into account. Similar motion

[1] Also known as approach phase.

planning scheme with inclusion of joint and velocity limits was proposed in [45].

Vision guided approaches for automated rendezvous and docking of nonspinning spacecraft have been proposed for various missions [46]–[50]. Wertz and Bell [46] give an overview of hardware and software technologies (sensors and actuators) required for autonomous rendezvous and docking of two spacecraft started at a remote distance. The terminal phase of the Demonstration of Autonomous Rendezvous Technology mission that includes proximity maneuvers for rendezvous to a cooperative spacecraft under an advanced video guidance sensor is described in [47]. The application of rendezvous and docking for space exploration missions regardless of whether the rendezvous is fully automated or has human in the loop is discussed in [48]. The design and application of the rendezvous guidance and navigation flight software for the Orbital Express flight demonstration are examined in [49]. Adaptive control law for spacecraft rendezvous and docking under measurement uncertainty, such as aggregation of sensor calibration parameter, systematic bias, or some stochastic disturbances, is proposed in [50]. Rouleau *et al.* [51] present a framework for robotic capture and servicing of satellites considering the shortcomings of the communication links such as bandwidth limitations and communication dropouts. There are different vision systems that are capable of estimating the pose of moving objects. Among them, an active vision system, such as the laser camera system (LCS), is preferable because of its robustness in face of the harsh lighting conditions of space [52]. As successfully verified during the STS-105 space mission, the 3-D imaging technology that is used in the LCS can indeed operate in the space environment. The use of laser range data has also been proposed for the motion estimation of *free-floating* space objects [53]–[55]. Linchter and Dubowsky [53] employed two separate Kalman filters (KFs) for the rotational and translational dynamics of a free-floating space object in order to reduce the noise of a range sensor. Since principal inertia of the target are directly included in the state vector to be estimated by KF, a normalization and reparameterization of the estimated inertia values have to be performed at every step of the KF cycle. Hillenbrand and Lampariello [54] developed a least-squares estimation method for motion estimation and the model identification of a free-floating space object that relies on the measurement range data, which consists displacements, and approximations of the velocity and acceleration obtained from differentiation of the visual data. The unknowns in this identification procedure consist the six inertia parameters of the target inertia tensor.

In this paper, we employ optimal control and estimation theory [56] for planning the motion of a robotic manipulator to intercept a noncooperative target satellite with unknown dynamics. A schematic diagram of the guidance and control system, as considered herein, is shown in Fig. 2. In this “prediction-planning-control” approach, first, an extended Kalman filter (EKF) gives reliable estimations of the motion states and the dynamics parameters of the free-floating target from noisy pose measurements of the target obtained from an LCS. This allows the motion of the target to be reliably predicted as soon as the filter converges. The convergence can be detected by monitoring the filter covariance matrix. Next, the robot motion to intercept the object at a rendezvous point is optimally planned and executed. The robot motion is planned to minimize a cost function subject to the constraint that the robot’s end-effector and the target’s grapple fixture arrive at their rendezvous point simultaneously at the same velocity. A line-of-sight angle is defined in such a way that zero angle corresponds to the best possible target alignment for robotic grasping of the fixture. The cost function is a weighted linear sum of travel time, distance, cosine of the line-of-sight angle, and a penalty function acting as a constraint on the acceleration magnitude during the capturing maneuvers. The weights in the cost function can be specified for optimization of each objectives. For example, time-optimal trajectory or distance optimal trajectory or a combination can be achieved by proper selection of the weights. The problem of finding optimal rendezvous point is formulated as solving the system Hamiltonian function for the terminal time, which can be solved by using a numerical technique such as the Newton–Raphson (NR) method. Subsequently, the position and velocity of the target at the time of rendezvous can be computed from the dynamics model of target object. Finally, the optimal trajectory generator is presented in a simple feedback form, where the acceleration at any given time is obtained by time-varying gains acting on the differences between the position/velocity of the end-effector at current time and those at the time of rendezvous. All matrices that are required to design the EKF are derived in closed-form rendering the prediction and motion-planning scheme suitable for real-time implementation. Moreover, to get around the quandary of nonidentifiability of the components of the inertia tensor, new parameters pertaining to the inertia tensor are used in the state vector to be estimated by EKF. Experiments are conducted to demonstrate the application of the proposed visual guidance method that can be used for robotic capturing of a free-floating object. A controlled robotic manipulator moves the mockup of a satellite as if it drifts and tumbles in zero-gravity. Another robotic manipulator is used to intercept a grapple fixture installed on the mockup using the proposed visual guidance scheme.

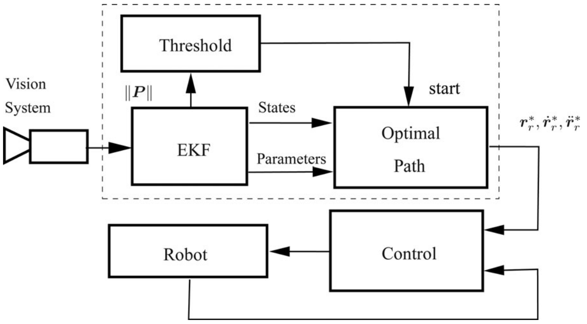

Fig. 2. Prediction-planning-control approach for robotic capturing of a free-floating object.

II. Dynamics Model

Fig. 3 illustrates a robot trying to capture a drifting and tumbling object. $\{\mathcal{A}\}$ is assumed the inertial frame, while coordinate

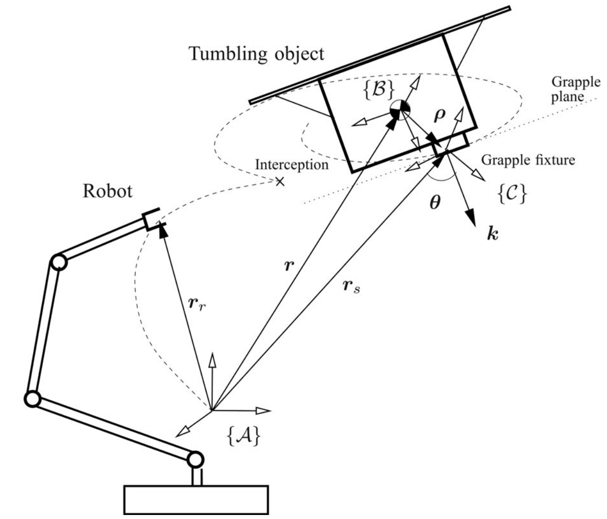

Fig. 3. Robotic capturing of a free-floating tumbling object.

frame $\{\mathcal{B}\}$ is attached to the body of the target. The origin of $\{\mathcal{B}\}$ is located at the target center of mass (CM) and the axes of $\{\mathcal{B}\}$ are oriented so as to be parallel to the principal axes of the target. Coordinate frame $\{\mathcal{C}\}$ is fixed to the target's grapple fixture located at $\boldsymbol{\rho}$ from the origin of $\{\mathcal{B}\}$; it is the pose of $\{\mathcal{C}\}$ which is measured by the vision system. We, further, assume that the target tumbles with angular velocity $\boldsymbol{\omega}$. Herein, both $\boldsymbol{\omega}$ and $\boldsymbol{\rho}$ are expressed in the target coordinate frame $\{\mathcal{B}\}$.

Let quaternion $\boldsymbol{q}$ represent the rotation of the target, i.e., the orientation of $\{\mathcal{B}\}$ with respect to $\{\mathcal{A}\}$. Then, the relation between the time derivative of the quaternion and the angular velocity can be readily expressed by

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\boldsymbol{q}, \quad \text{where} \quad \boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -[\boldsymbol{\omega}\times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}. \tag{1}$$

Coincidentally, the matrix function $\boldsymbol{\Omega}(\cdot)$ also appears in the quaternion product. Let us define operator $\otimes$ acting on quaternion $\boldsymbol{q}$ as

$$\boldsymbol{q}\otimes \overset{\triangle}{=} q_0\mathbf{1}_4 + \boldsymbol{\Omega}(\boldsymbol{q}_v) \tag{2}$$

where $\boldsymbol{q}_v$ is the vector part of the quaternion, and q_o is the scalar part such that $\boldsymbol{q} = [\boldsymbol{q}_v^T \;\; q_o]^T$. Then, the product of two quaternions $\boldsymbol{q}_1 \otimes \boldsymbol{q}_2$ corresponds to rotation matrix $\boldsymbol{A}(\boldsymbol{q}_2)\boldsymbol{A}(\boldsymbol{q}_1)$, where the rotation matrix $\boldsymbol{A}(\boldsymbol{q})$ is related to the corresponding quaternion by

$$\boldsymbol{A}(\boldsymbol{q}) = (2q_o^2 - 1)\mathbf{1}_3 + 2q_o[\boldsymbol{q}_v\times] + 2\boldsymbol{q}_v\boldsymbol{q}_v^T. \tag{3}$$

Dynamics of the rotational motion of the object can be described by Euler's equations written in the terms of the moment of inertia. We will need these equations for not only trajectory planning purposes but also estimation of the target states and parameters. However, it is important to note that it is not possible to identify the absolute values of the inertia parameters from observations of torque-free tumbling motion of an object. For instance, one can observe that the motion trajectories of a rigid body with zero input torque remains unchanged when all inertia variables are scaled up or down by the same factor. To circumvent the nonidentifiability problem, we use the following nondimensional inertia parameters:

$$\begin{aligned} p_x &= \frac{I_{yy} - I_{zz}}{I_{xx}} \\ p_y &= \frac{I_{zz} - I_{xx}}{I_{yy}} \\ p_z &= \frac{I_{xx} - I_{yy}}{I_{zz}} \end{aligned} \tag{4}$$

where I_{xx}, I_{yy}, and I_{zz} denote the principal moments of inertia of the target, and $\boldsymbol{I}_c = \mathrm{diag}(I_{xx}, I_{yy}, I_{zz})$ is the corresponding inertia tensor. Notice that since the absolute inertial parameters satisfy the following inequalities:

$$\begin{aligned} I_{xx} + I_{yy} &> I_{zz} \\ I_{yy} + I_{zz} &> I_{xx} \\ I_{zz} + I_{xx} &> I_{yy} \end{aligned}$$

the nondimensional inertia parameters can only take values grater than -1, i.e.,

$$p_x > -1, \quad p_y > -1, \quad p_z > -1.$$

The Euler's equations can be expressed in terms of the new parameters as

$$\dot{\boldsymbol{\omega}} = \boldsymbol{d}(\boldsymbol{\omega}, \boldsymbol{p}) + \boldsymbol{B}(\boldsymbol{p})\boldsymbol{\epsilon}_\tau \tag{5}$$

where

$$\boldsymbol{d}(\boldsymbol{\omega}, \boldsymbol{p}) = \begin{bmatrix} p_x\omega_y\omega_z \\ p_y\omega_x\omega_z \\ p_z\omega_x\omega_y \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} b_x(\boldsymbol{p}) & 0 & 0 \\ 0 & b_y(\boldsymbol{p}) & 0 \\ 0 & 0 & b_z(\boldsymbol{p}) \end{bmatrix}$$

and the entries of the aforementioned diagonal matrix can be computed from inertial ratio parameters $\boldsymbol{p} = [p_x \;\; p_y \;\; p_z]^T$ as

$$\begin{aligned} b_x(\boldsymbol{p}) &= 1 + \frac{1 + p_x}{1 - p_y} + \frac{1 - p_x}{1 + p_z} \\ b_y(\boldsymbol{p}) &= 1 + \frac{1 + p_y}{1 - p_z} + \frac{1 - p_y}{1 + p_x} \\ b_z(\boldsymbol{p}) &= 1 + \frac{1 + p_z}{1 - p_x} + \frac{1 - p_z}{1 + p_y} \end{aligned} \tag{6}$$

and

$$\boldsymbol{\epsilon}_\tau = \frac{\boldsymbol{\tau}}{\mathrm{tr}(\boldsymbol{I}_c)} \tag{7}$$

is defined as the acceleration disturbance, in which $\mathrm{tr}(\cdot)$ denotes the trace of a matrix (see the Appendix for detailed derivations). Note that $\boldsymbol{\epsilon}_\tau$ represents the lumped effects of flexible appendages like solar panels, fuel sloshing, and gravity gradient, which were not considered in our model. Nevertheless, in practice, it is usually sufficient to model them as process noise in the state estimation process [57], [58]. It is worth noting that $\boldsymbol{\epsilon}_\tau$ is independent of the choice of local frame because $\mathrm{tr}(\boldsymbol{I}_c)$ is an invariant quantity. Therefore, the covariance of $\boldsymbol{\epsilon}_\tau$ is equal to that of torque disturbance $\boldsymbol{\tau}$ scaled by constant scalar $[\mathrm{tr}(\boldsymbol{I}_c)]^{-1}$. Here, the process noise is assumed to be with covariance $E[\boldsymbol{\epsilon}_\tau\boldsymbol{\epsilon}_\tau^T] = \sigma_\tau^2\mathbf{1}_3$.

Denoting $\boldsymbol{r}$ as the location of the target CM expressed in the inertial frame $\{\mathcal{A}\}$ and $\boldsymbol{v} = \dot{\boldsymbol{r}}$ as the drift velocity, the translational motion of the target can be simply expressed by

$$\dot{\boldsymbol{v}} = \boldsymbol{\epsilon}_f \tag{8}$$

where $\boldsymbol{\epsilon}_f$ is the disturbance force per unit mass, which is assumed to be white noise with covariance $E[\boldsymbol{\epsilon}_f \ \boldsymbol{\epsilon}_f^T] = \sigma_f^2 \mathbf{1}_3$. In addition, the values of the position and velocity of the grapple fixture can be obtained from

$$\boldsymbol{r}_s = \boldsymbol{r} + \boldsymbol{A}(\boldsymbol{q})\boldsymbol{\rho} \tag{9a}$$

$$\dot{\boldsymbol{r}}_s = \boldsymbol{v} + \boldsymbol{A}(\boldsymbol{q})(\boldsymbol{\omega} \times \boldsymbol{\rho}). \tag{9b}$$

In the rest of this paper, the stochastic model of the target embodied in (1), (5), and (8) will be used for optimal estimation of the motion states and parameters, whereas the deterministic part of the model, i.e., when $\boldsymbol{\epsilon}_\tau = \boldsymbol{\epsilon}_f = \mathbf{0}$, will be employed for optimal motion planning of the robotic manipulator.

III. Optimal Robot Guidance

A. Optimal Rendezvous Point and Path Planning

This section presents an optimal approach to robot-motion planning for smooth grasping of a free-floating object. The robot-motion planner must determine when, where, and how to grasp the moving grapple fixture to satisfy some objectives. We assume that the dynamics parameters of the target $\{\boldsymbol{p}, \ \boldsymbol{\rho}\}$ are known. This assumption will be relaxed in the next section. The motion planning of the robot is based on prediction of the motion of target computed from the values of the states (and parameters) of the target estimated at the initial time of the capturing time interval. This predicting-planning scheme relives the need for continuous visual feedback information during the capturing operation. This is important for a successful robotic grasping because most likely the vision system will be obstructed by the robotic arm at one point during its approach to the target. We also assume that the chaser satellite is placed in a suitable position relative to the client satellite so that robotic capturing within the workspace envelope becomes feasible. Moreover, although the solution of the optimal path is given in the task space, one may use a singularity robust scheme for inverse kinematic solution of nonredundant manipulators [59], [60], or a singularity management and self-collision avoidance scheme for kinematic control of redundant manipulators [61]–[65]. Despite space manipulators mounted on a free-floating base provide further redundancy, they also introduce dynamic singularities [66]. Several algorithms have been proposed to solve the inverse kinematics of redundant free-floating robotic systems [67]–[69] by means of the so-called generalized Jacobian. Notice that generalized Jacobian, owing to its dynamic nature, captures dynamic singularities.

As schematically illustrated in Fig. 3, the end-effector position is represented by $\boldsymbol{r}_r$. The end-effector and the grasping point are expected to arrive at a rendezvous-point simultaneously with the same velocity in order to avoid impact at the time of grasping. Therefore, the main objective is to bring the end-effector from its initial position $\boldsymbol{r}_r(t_o)$ at initial time $t = t_o$ to a grasping location along the trajectory of the grapple fixture on the target satellite, while satisfying the acceleration magnitude constraint $\|\ddot{\boldsymbol{r}}_r(t)\| \le a$ for $t \in [t_o \ \ t_f]$ and terminal conditions

$$\boldsymbol{r}_r(t_f) = \boldsymbol{r}_s(t_f) \quad \text{and} \quad \dot{\boldsymbol{r}}_r(t_f) = \dot{\boldsymbol{r}}_s(t_f). \tag{10}$$

Note that the initial time t_o is assumed to be fixed and t_f variable. The aforementioned terminal constraints for a successful grasping are necessary but not sufficient because the grapple fixture should be accessible for the robotic interception too. In order to gauge the accessibility of the grapple fixture by the robot, let us define the line of sight angle θ that is made by the opposite direction of vector $\boldsymbol{r}_r$ (at the time of interception) and unit vector $\boldsymbol{k}$ attached to the grapple fixture (see Fig. 3). It is assumed that vector $\boldsymbol{k}$ is expressed in the target frame $\{\mathcal{C}\}$ in such a way that $\theta = 0$ corresponds to the most suitable object alignment for robotic grasping, and conversely, the grasping is not possible when $\theta = \pi$. Therefore, it will make sense to maximize the cosine of the line-of-sight angle at the time of interception.

Now, assume that the optimal trajectory is generated by the following second-order system:

$$\ddot{\boldsymbol{r}}_r = \boldsymbol{u} \tag{11}$$

which can be thought of as the linearized dynamics of a rigid manipulator. Now, denoting the state vector of the entire system as

$$\boldsymbol{x} = [\boldsymbol{q}^T \quad \boldsymbol{\omega}^T \quad \boldsymbol{r}^T \quad \boldsymbol{r}_r^T \quad \dot{\boldsymbol{r}}_r^T]^T$$

we get the following autonomous system (time-invariant):

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\boldsymbol{q} \\ \boldsymbol{d}(\boldsymbol{\omega}) \\ \boldsymbol{v} \\ \dot{\boldsymbol{r}}_r \\ \boldsymbol{u} \end{bmatrix} \tag{12}$$

because the drift velocity $\boldsymbol{v}$ is constant.

In the following analysis, an optimal solution to the input $\boldsymbol{u}$ of system (12) is sought to minimize the following cost function:

$$J = \phi[\boldsymbol{x}(t_f)] + \int_{t_o}^{t_f} 1 + c(\boldsymbol{u}) d\tau \tag{13}$$

in which

$$c(\boldsymbol{u}) = \begin{cases} \kappa(\|\boldsymbol{u}\|^2 - a^2), & \text{if} \quad \|\boldsymbol{u}\| \ge a \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

is a penalty function added to the cost function so that it acts as a “soft” constraint [56] to the acceleration magnitude, subject to the following terminal state constraint:

$$\boldsymbol{\psi}[\boldsymbol{x}(t_f)] = \mathbf{0} \tag{15}$$

and

$$\boldsymbol{\psi}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{r}_s(\boldsymbol{x}_s) - \boldsymbol{r}_r \\ \dot{\boldsymbol{r}}_s(\boldsymbol{x}_s) - \dot{\boldsymbol{r}}_r \end{bmatrix} \tag{16}$$

and vector $\boldsymbol{x}_s = [\boldsymbol{q}^T \ \ \boldsymbol{\omega}^T \ \ \boldsymbol{r}^T]^T$ contains the time-varying states of the target. A “soft constraint” with very large penalty becomes equivalent to a “hard constraint” [56]. This means that scalar $\kappa > 0$ in penalty function (14) should be selected to be sufficiently large. If the nonintegral part of the cost function (13) is set to zero, then the time of travel $t_f - t_o$ is minimized, i.e., time-optimal trajectory. However, the integral cost function is augmented by the following term:

$$\phi(\boldsymbol{x}) = w_1 \|\boldsymbol{r}_r\| - w_2 \cos\theta \tag{17}$$

where $w_1 > 0$ and $w_2 > 0$ are scaler weights. Clearly, the incorporation of ϕ in the cost function tends to minimize the distance $\|\boldsymbol{r}_r(t_f)\|$ and maximize the value of the cosine of the line-of-sight angle $\theta(t_f)$. Denoting $\boldsymbol{k}'(\boldsymbol{q}) = \boldsymbol{A}(\boldsymbol{q})\boldsymbol{k}$ as the unit vector expressed in the inertial frame, one can readily express the cosine as a function of the states, i.e.,

$$\cos\theta = -\frac{\boldsymbol{r}_r \cdot \boldsymbol{k}'(\boldsymbol{q})}{\|\boldsymbol{r}_r\|}.$$

In summary, cost function (13) is a weighted linear sum of the various objectives. Therefore, one can specify scalar weights for each objective to be optimized. For instance, $w_1 \gg w_2 \gg 1$ corresponds to minimum distance path, $w_2 \ll w_1 \ll 1$ corresponds to time-optimal path, and $w_2 \gg w_1 \gg 1$ corresponds to minimum line-of-sight angle at the time of rendezvous.

The Hamiltonian for the system (12) and (13) can be written as

$$\mathcal{H} = 1 + c(\boldsymbol{u}) + \lambda^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}). \tag{18}$$

The Hessian of the Hamiltonian

$$\frac{\partial^2 \mathcal{H}}{\partial^2 \boldsymbol{u}} = \begin{cases} 2\kappa \mathbf{1}_3, & \text{if } \|\boldsymbol{u}\| \geq a \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

is positive definite only if the input satisfies $\|\boldsymbol{u}\| \geq a$, and thereby satisfying the sufficient condition for local minimality. According to the optimal control theory [70], the costate and optimal input, i.e., $\boldsymbol{u}^*$, must satisfy the following partial derivatives:

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}, \qquad \frac{\partial \mathcal{H}}{\partial \boldsymbol{u}} = \boldsymbol{0}. \tag{20}$$

Splitting the vector of costate as $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_s^T \; \boldsymbol{\lambda}_r^T \; \boldsymbol{\lambda}_u^T]^T$, where $\boldsymbol{\lambda}_s \in \mathbb{R}^9$ and $\boldsymbol{\lambda}_r, \boldsymbol{\lambda}_u \in \mathbb{R}^3$, and applying (20) to our Hamiltonian (18) yields the following identities:

$$\dot{\boldsymbol{\lambda}}_r = \boldsymbol{0} \tag{21a}$$

$$\dot{\boldsymbol{\lambda}}_u = -\boldsymbol{\lambda}_r \tag{21b}$$

$$\boldsymbol{\lambda}_u = -2\kappa \boldsymbol{u}^*. \tag{21c}$$

One can readily infer from (21a)–(21c) that the expression of the acceleration trajectory is

$$\boldsymbol{u}^* = \boldsymbol{\alpha}(t - t_o) + \boldsymbol{\beta} \tag{22}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are defined such that

$$\begin{aligned} \boldsymbol{\lambda}_r &= 2\kappa\boldsymbol{\alpha} \\ \boldsymbol{\lambda}_u &= -2\kappa\boldsymbol{\alpha}(t - t_o) - 2\kappa\boldsymbol{\beta}. \end{aligned} \tag{23}$$

Subsequently, one can obtain the position and velocity trajectories via integration, i.e.,

$$\dot{\boldsymbol{r}}_r(t) - \dot{\boldsymbol{r}}_r(t_o) = \frac{1}{2}\boldsymbol{\alpha}(t - t_o)^2 + \boldsymbol{\beta}(t - t_o)$$

$$\boldsymbol{r}_r(t) - \boldsymbol{r}_r(t_o) = \frac{1}{6}\boldsymbol{\alpha}(t - t_o)^3 + \frac{1}{2}\boldsymbol{\beta}(t - t_o)^2 + \dot{\boldsymbol{r}}_r(t_o)(t - t_o). \tag{24}$$

The constant polynomial coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are solved by imposing the initial and terminal conditions (16)

$$\begin{aligned} \boldsymbol{\alpha} &= \frac{6}{(t_f - t_o)^2}[\dot{\boldsymbol{r}}_s(t_f) + \dot{\boldsymbol{r}}_r(t_o)] \\ &\quad - \frac{12}{(t_f - t_o)^3}\big[\boldsymbol{r}_s(t_f) - \boldsymbol{r}_r(t_o)\big] \\ \boldsymbol{\beta} &= -\frac{2}{t_f - t_o}[\dot{\boldsymbol{r}}_s(t_f) + 2\dot{\boldsymbol{r}}_r(t_o)] \\ &\quad + \frac{6}{(t_f - t_o)^2}\big[\boldsymbol{r}_s(t_f) - \boldsymbol{r}_r(t_o)\big]. \end{aligned} \tag{25}$$

If the terminal time t_f is given, then $\boldsymbol{r}_s(t_f)$ and $\dot{\boldsymbol{r}}_s(t_f)$ can be computed from (9). However, the optimal terminal time t_f^* remains to be found. The *optimal Hamiltonian* $\mathcal{H}^* = \mathcal{H}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda})$ that is calculated at optimal point $\boldsymbol{u}^*$ and $\boldsymbol{x}^*$ must satisfy $\mathcal{H} = 0$, i.e.,

$$\mathcal{H}^*(t_f^*) = 1 - \kappa a^2 - 2\kappa\|\boldsymbol{\beta}\|^2 + \boldsymbol{\lambda}_s^T \dot{\boldsymbol{x}}_s = 0. \tag{26}$$

This gives the extra equation required to determine the optimal terminal time. We will show in the following analysis that $\boldsymbol{\lambda}_s^T \dot{\boldsymbol{x}}_s$ can be expressed as a function of t_f. The final values of the costate can be obtained from the end-point constraint equation referred to as the *transversality* condition:

$$\boldsymbol{\lambda}(t_f) = \left(\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\right)^T \boldsymbol{\nu} + \frac{\partial \phi}{\partial \boldsymbol{x}}\bigg|_{\boldsymbol{x}(t_f)} \tag{27}$$

where vector $\boldsymbol{\nu} = [\boldsymbol{\nu}_1^T \; \boldsymbol{\nu}_2^T]^T$, where $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2 \in \mathbb{R}^3$, is the Lagrangian multiplier owing to the constraint (16). Using (16) in (27) yields

$$\boldsymbol{\lambda}(t_f) = \begin{bmatrix} (\nabla_{\boldsymbol{x}_s}\boldsymbol{r}_s)^T\boldsymbol{\nu}_1 + (\nabla_{\boldsymbol{x}_s}\dot{\boldsymbol{r}}_s)^T\boldsymbol{\nu}_2 + \nabla_{\boldsymbol{x}_s}\phi \\ -\boldsymbol{\nu}_1 + \nabla_{\boldsymbol{r}_r}\phi \\ -\boldsymbol{\nu}_2 \end{bmatrix}_{\boldsymbol{x}(t_f)}. \tag{28}$$

Now, equating right-hand side (RHS) of (23) and (28) will result in a set of three equations with three unknowns $\boldsymbol{\nu}_1$, $\boldsymbol{\nu}_2$, and $\boldsymbol{\lambda}_s^T \dot{\boldsymbol{x}}_s$. Solving the set of equations gives the following useful identity:

$$\begin{aligned} \boldsymbol{\lambda}_s^T \dot{\boldsymbol{x}}_s &= \boldsymbol{\nu}_1^T \dot{\boldsymbol{r}}_s + \boldsymbol{\nu}_2^T \ddot{\boldsymbol{r}}_s + \nabla_{\boldsymbol{x}_s}\phi^T \dot{\boldsymbol{x}}_s \\ &= (\nabla_{\boldsymbol{r}_r}\phi - 2\kappa\boldsymbol{\alpha})^T \dot{\boldsymbol{r}}_s + 2\kappa\big(\boldsymbol{\alpha}(t_f - t_o) + \boldsymbol{\beta}\big)^T \ddot{\boldsymbol{r}}_s + \nabla_{\boldsymbol{q}}\phi^T \dot{\boldsymbol{q}} \end{aligned} \tag{29}$$

in which we use identity $\nabla_{\boldsymbol{x}_s}\phi^T \dot{\boldsymbol{x}}_s = \nabla_{\boldsymbol{q}}\phi^T \dot{\boldsymbol{q}}$ (note that function ϕ does not depend on $\boldsymbol{\omega}$ and $\boldsymbol{r}$). Finally, substituting (29) into (26) and using (24), we will arrive at

$$\begin{aligned} \mathcal{H}(t_f) &= 1 - \kappa a^2 - 2\kappa\|\boldsymbol{\beta}\|^2 + 2\boldsymbol{\kappa}\boldsymbol{\beta}^T \ddot{\boldsymbol{r}}_s + \frac{1}{2}\nabla_{\boldsymbol{q}}\phi^T \boldsymbol{\Omega}(\boldsymbol{\omega})\boldsymbol{q} \\ &\quad + \big(\boldsymbol{\beta}^T \nabla_{\boldsymbol{r}_r}\phi - 2\kappa\boldsymbol{\alpha}^T\boldsymbol{\beta} + 2\kappa\boldsymbol{\alpha}^T \ddot{\boldsymbol{r}}_s\big)(t_f - t_o) \\ &\quad + \left(\frac{1}{2}\boldsymbol{\alpha}^T \nabla_{\boldsymbol{r}_r}\phi - \kappa\|\boldsymbol{\alpha}\|^2\right)(t_f - t_o)^2 = 0 \end{aligned} \tag{30}$$

where

$$\nabla_{\boldsymbol{r}_r}\phi[\boldsymbol{x}(t_f)] = w_1\frac{\boldsymbol{r}_s}{\|\boldsymbol{r}_s\|} + w_2\frac{\boldsymbol{r}_s\times(\boldsymbol{r}_s\times\boldsymbol{k}')}{\|\boldsymbol{r}_s\|^3}$$

$$\nabla_{\boldsymbol{q}}\phi[\boldsymbol{x}(t_f)] = \frac{w_2}{\|\boldsymbol{r}_s\|}\begin{bmatrix}2q_o\boldsymbol{k}\times\boldsymbol{r}_s + (\boldsymbol{q}_v^T\boldsymbol{k})\boldsymbol{r}_s + (\boldsymbol{q}_v^T\boldsymbol{r}_s)\boldsymbol{k}\\ 2\boldsymbol{r}_s^T(\boldsymbol{q}_v\times\boldsymbol{k}+2q_o\boldsymbol{k})\end{bmatrix} \tag{31}$$

and the predicted acceleration, i.e., $\ddot{\boldsymbol{r}}_s(t_f)$, can be obtained from the time derivative of (9b) as

$$\ddot{\boldsymbol{r}}_s = \boldsymbol{A}(\boldsymbol{q})\Big(\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{\rho}) + \boldsymbol{d}(\boldsymbol{\omega})\times\boldsymbol{\rho}\Big). \tag{32}$$

In derivation of (31), we use the identity

$$[\boldsymbol{r}_s\times]^2 = -\|\boldsymbol{r}_s\|^2\,\mathbf{1}_3 + \boldsymbol{r}_s\boldsymbol{r}_s^T.$$

It should be noted that the expression of the optimal rendezvous time t_f in (30) is not given explicitly. Therefore, (30) should be solved in terms of t_f using a numerical method such as the NR method (see the Appendix for details). It should be pointed out that the NR gives only one solution, whereas $\mathcal{H}(t_f) = 0$ may have multiple solutions. Nevertheless, there are several other numerical methods that are available to seek all real solutions to a nonlinear equation [71], [72].

B. Motion Planning Using Feedback

In the previous section, the optimal rendezvous point and robot path are fixed after being planned at initial time t_o based on the future trajectories of the position, velocity, and acceleration; see (30), (22), and (25), respectively. However, as time goes by, more accurate estimates of the target trajectories may be available, and therefore, it seems logical to update the robot trajectories accordingly. The section presents the optimal control input $\boldsymbol{u}^*$ in a feedback form, which is suitable for continual update in the face of any modification in the prediction of the target motion that may occur during $t\in[t_o\ t_f)$.

The expression of the optimal control input in (22) and (25) can be written as a function of initial time t_o and current time t in this form

$$\boldsymbol{u}^*(t_o,t) = \boldsymbol{\Gamma}(t_o,t)\boldsymbol{e}(t_o) + \boldsymbol{\gamma}(t_o,t), \qquad t\in[t_o,t_f) \tag{33}$$

where

$$\boldsymbol{e}(t) = \begin{bmatrix}\boldsymbol{r}_s(t_f) - \boldsymbol{r}_r(t)\\ \dot{\boldsymbol{r}}_s(t_f) - \dot{\boldsymbol{r}}_r(t)\end{bmatrix}$$

can be treated as the feedback error on which a controller with following time-varying feedback gain acts:

$$\boldsymbol{\Gamma}(t_o,t) = \begin{bmatrix}\dfrac{-12t+6t_o+6t_f}{(t_f-t_o)^3}\mathbf{1}_3 & \dfrac{6t-4t_o-2t_f}{(t_f-t_o)^2}\mathbf{1}_3\end{bmatrix}$$

$$\boldsymbol{\gamma}(t_o,t) = \frac{12t-6t_o-6t_f}{(t_f-t_o)^2}\dot{\boldsymbol{r}}_s(t_f). \tag{34}$$

Now, the control input at any given time $t\in[t_o,t_f)$ can be also calculated by setting the values of the position and velocity at that time as the initial values in (33). Therefore, by setting the floating initial time $t_o = t$ in (34), we will arrive at

$$\boldsymbol{\Gamma}(t,t) = \begin{bmatrix}\dfrac{6}{(t_f-t)^2}\mathbf{1}_3 & \dfrac{-2}{t_f-t}\mathbf{1}_3\end{bmatrix}$$

$$\boldsymbol{\gamma}(t,t) = \frac{-6}{t_f-t}\dot{\boldsymbol{r}}_s(t_f). \tag{35}$$

Finally, using shorthand notation $\boldsymbol{u}^*(t,t) = \boldsymbol{u}^*(t)$, $\boldsymbol{\Gamma}(t,t) = \boldsymbol{\Gamma}(t)$, and $\boldsymbol{\gamma}(t,t) = \boldsymbol{\gamma}(t)$, one can rewrite (33) as

$$\boldsymbol{u}^*(t) = \begin{cases}\boldsymbol{\Gamma}(t)\boldsymbol{e}(t) + \boldsymbol{\gamma}(t), & \text{if } t_o\le t\le t_f-\epsilon_u\\ \mathbf{0}_{3\times1}, & \text{otherwise}\end{cases} \tag{36}$$

where $\epsilon_u > 0$ can be selected to be arbitrary small. Notice that optimal trajectories (33) and (36) corresponding to the fixed and floating initial times are equivalent as long as the final values $\boldsymbol{r}_s(t_f)$ and $\dot{\boldsymbol{r}}_s(t_f)$ remain constant. However, (36) is at advantage of being able to automatically update the path trajectories when the final values change.

IV. Simultaneous Estimation of States and Parameters

The implementation of the optimal motion-planning scheme requires the values of the full motion states as well as the dynamic parameters of the target. Since our vision system gives only measurement of the object pose, we need to estimate the rest of variables. This section describes an EKF-based estimator [73], [74] to estimate not only the system states but also all the relevant parameters. First, linearized models for the observation and dynamics systems will be presented; then, the filter design follows.

A. Observation Equations

Let us assume that the vision system gives a noisy measurement of the position and orientation of coordinate $\{\mathcal{C}\}$ attached to the grapple fixture with respect to the inertial coordinate system $\{\mathcal{A}\}$ (see Fig. 3). Denoting the orientation of $\{\mathcal{C}\}$ with respect to $\{\mathcal{A}\}$ by unit quaternion $\boldsymbol{\eta}$, one can write the measurement vector as

$$\boldsymbol{z} = \begin{bmatrix}\boldsymbol{r}_s\\ \boldsymbol{\eta}\end{bmatrix} + \boldsymbol{v} \tag{37}$$

where $\boldsymbol{v}$ is the measurement noise, which is assumed to be white with covariance $\boldsymbol{R} = E[\boldsymbol{v}\boldsymbol{v}^T]$. The orientation of the principal axes, i.e., the alignment of coordinate frame $\{\mathcal{C}\}$ with respect to $\{\mathcal{B}\}$, is not *a priori* known. Therefore, introducing unit quaternion $\boldsymbol{\mu}$ to represent the misalignment between coordinates $\{\mathcal{B}\}$ and $\{\mathcal{C}\}$, the quaternion $\boldsymbol{\eta}$ can be considered as two successive orientations as

$$\boldsymbol{\eta} = \boldsymbol{\mu}\otimes\boldsymbol{q}. \tag{38}$$

In order to avoid introducing a new variable, we redefined vector $\boldsymbol{x}$ as

$$\boldsymbol{x} = [\boldsymbol{q}_v^T \quad \boldsymbol{\omega}^T \quad \boldsymbol{r}^T \quad \boldsymbol{v}^T \quad \boldsymbol{p}^T \quad \boldsymbol{\rho}^T \quad \boldsymbol{\mu}_v^T]^T \tag{39}$$

to be the state vector of the EKF estimator, where $\boldsymbol{q}_v$ and $\boldsymbol{\mu}_v$ denote the vector parts of the corresponding quaternions. Now, the observation equations can be written as

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v}$$

where

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{r} + \boldsymbol{A}(\boldsymbol{q})\boldsymbol{\rho}\\ \boldsymbol{\mu}\otimes\boldsymbol{q}\end{bmatrix} \tag{40}$$

is a nonlinear function of quaternions $\boldsymbol{q}$ and $\boldsymbol{\mu}$. To linearize the observation vector, one needs to derive the sensitivity of the nonlinear observation vector with respect to small variations $\delta\boldsymbol{q}_v$ and $\delta\boldsymbol{\mu}_v$, where $\|\delta\boldsymbol{q}_v\| \ll 1$ and $\|\delta\boldsymbol{\mu}_v\| \ll 1$. To this end, consider quaternion variations

$$\delta\boldsymbol{q} = \boldsymbol{q} \otimes \hat{\boldsymbol{q}}^{-1} \tag{41a}$$

$$\delta\boldsymbol{\mu} = \hat{\boldsymbol{\mu}}^{-1} \otimes \boldsymbol{\mu} \tag{41b}$$

as the deviations of the corresponding quaternions from their nominal values $\hat{\boldsymbol{q}}$ and $\hat{\boldsymbol{\mu}}$. Here, the inverse of quaternion is defined such that $\boldsymbol{q} \otimes \boldsymbol{q}^{-1} = [0\ \ 0\ \ 0\ \ 1]^T$, where $[0\ \ 0\ \ 0\ \ 1]^T$ is the identity quaternion. It can be readily seen from (3) that, for a small rotation, where $\|\delta\boldsymbol{q}_v\| \ll 1$ and $\delta q_o \approx 1$, the first-order approximation of the rotation matrix can be written as

$$\boldsymbol{A}(\delta\boldsymbol{q}) = \mathbf{1}_3 + 2[\delta\boldsymbol{q}_v \times] + \text{H.O.T.}$$

where H.O.T. stands for second and higher order terms. Hence, the first row of the RHS of (40) can be approximated (first-order approximation) as

$$\boldsymbol{r}_s = \boldsymbol{r} + \boldsymbol{A}(\hat{\boldsymbol{q}})(\mathbf{1}_3 + 2[\delta\boldsymbol{q}_v \times])\boldsymbol{\rho} + \text{H.O.T.} \tag{42}$$

Now, we need to linearize the orientation component of the observation vector. Substituting $\boldsymbol{q}$ and $\boldsymbol{\mu}$ from (41) into (38) yields

$$\begin{aligned}\boldsymbol{\eta} &= \hat{\boldsymbol{\mu}} \otimes \delta\boldsymbol{\mu} \otimes \delta\boldsymbol{q} \otimes \hat{\boldsymbol{q}} \\ &= \hat{\boldsymbol{\mu}} \otimes \delta\boldsymbol{\eta} \otimes \hat{\boldsymbol{q}}\end{aligned} \tag{43}$$

where

$$\delta\boldsymbol{\eta} = \delta\boldsymbol{\mu} \otimes \delta\boldsymbol{q}. \tag{44}$$

Using the definition of the quaternion product (2) and neglecting the second and higher order terms, one can rewrite (43) in the following form:

$$\boldsymbol{\eta} \approx \hat{\boldsymbol{\eta}} + \boldsymbol{\Gamma}(\hat{\boldsymbol{q}}, \hat{\boldsymbol{\mu}})\delta\boldsymbol{\eta}_v$$

where $\hat{\boldsymbol{\eta}} = \hat{\boldsymbol{\mu}} \otimes \hat{\boldsymbol{q}}$, and

$$\boldsymbol{\Gamma} = \begin{bmatrix} \hat{q}_o\hat{\eta}_o\mathbf{1}_3 - \hat{q}_o[\hat{\boldsymbol{\mu}}_v \times] + \hat{\eta}_o[\hat{\boldsymbol{q}}_v \times] - [\hat{\boldsymbol{q}}_v \times][\hat{\boldsymbol{\mu}}_v \times] - \hat{\boldsymbol{q}}_v\hat{\boldsymbol{\mu}}_v^T \\ (\hat{\boldsymbol{q}}_v \times \hat{\boldsymbol{\mu}}_v)^T - \hat{\eta}_o\hat{\boldsymbol{q}}_v^T - \hat{q}_o\hat{\boldsymbol{\mu}}_v^T \end{bmatrix}.$$

Moreover, from (44), we get

$$\begin{aligned}\delta\boldsymbol{\eta}_v &= \delta\boldsymbol{q}_v + \delta\boldsymbol{\mu}_v - \delta\boldsymbol{\mu}_v \times \delta\boldsymbol{q}_v \\ &\approx \delta\boldsymbol{q}_v + \delta\boldsymbol{\mu}_v\end{aligned} \tag{45}$$

and therefore

$$\boldsymbol{\eta} = \hat{\boldsymbol{\eta}} + \boldsymbol{\Gamma}(\hat{\boldsymbol{q}}, \hat{\boldsymbol{\mu}})[\delta\boldsymbol{q}_v + \delta\boldsymbol{\mu}_v] + \text{H.O.T.} \tag{46}$$

Finally, from the first-order approximation of the observation vector in (42) and (46), one can infer the observation sensitivity matrix as

$$\begin{aligned}\boldsymbol{H} &= \frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} \\ &= \begin{bmatrix} -2\boldsymbol{A}(\hat{\boldsymbol{q}})[\hat{\boldsymbol{\rho}}\times] & \mathbf{0}_3 & \mathbf{1}_3 & \mathbf{0}_{3\times 6} & \boldsymbol{A}(\hat{\boldsymbol{q}}) & \mathbf{0}_3 \\ \boldsymbol{\Gamma}(\hat{\boldsymbol{q}}, \hat{\boldsymbol{\mu}}) & \mathbf{0}_{4\times 3} & \mathbf{0}_{4\times 3} & \mathbf{0}_{4\times 6} & \mathbf{0}_{4\times 3} & \boldsymbol{\Gamma}(\hat{\boldsymbol{q}}, \hat{\boldsymbol{\mu}}) \end{bmatrix}.\end{aligned} \tag{47}$$

B. Extended Kalman Filter Design

Assume that the dynamics parameters, i.e., $\boldsymbol{\theta} = [\boldsymbol{p}^T\ \boldsymbol{\rho}^T\ \boldsymbol{\mu}_v^T]^T$, remained constant, and therefore

$$\dot{\boldsymbol{\theta}} = \mathbf{0}_{9\times 1}. \tag{48}$$

Now, in view of (1), (5), and (8) together with (48), the process dynamics can be described in this compact form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_s(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{\epsilon} \tag{49}$$

where

$$\boldsymbol{G}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ \boldsymbol{B}(\boldsymbol{p}) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{1}_3 \\ \mathbf{0}_{9\times 3} & \mathbf{0}_{9\times 3} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\epsilon} = \begin{bmatrix} \boldsymbol{\epsilon}_\tau \\ \boldsymbol{\epsilon}_f \end{bmatrix}$$

contains the entire process noise. Equation (49) can be used for state propagation. However, the state transition matrix of the linearized system will be also needed to be used for covariance propagation of EKF. Adopting a linearization technique similar to that in [57] and [58], one can linearize the equation of quaternion (1) as

$$\frac{\mathrm{d}}{\mathrm{dt}}\delta\boldsymbol{q}_v = -\boldsymbol{\omega} \times \delta\boldsymbol{q}_v + \frac{1}{2}\delta\boldsymbol{\omega} \tag{50a}$$

$$\frac{\mathrm{d}}{\mathrm{dt}}\delta q_o = 0. \tag{50b}$$

Note that, since δq_o is not an independent variable and it has variations of only the second order, its time derivative can be ignored, as suggested in [57]. Moreover, linearization of function $\boldsymbol{d}(\boldsymbol{\omega}, \boldsymbol{p})$ in the Euler equation (5) yields

$$\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{\omega}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \boldsymbol{M}(\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{p}}) = \begin{bmatrix} 0 & \hat{p}_x\hat{\omega}_z & \hat{p}_x\hat{\omega}_y \\ \hat{p}_y\hat{\omega}_z & 0 & \hat{p}_y\hat{\omega}_x \\ \hat{p}_z\hat{\omega}_y & \hat{p}_z\hat{\omega}_x & 0 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{p}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \boldsymbol{N}(\hat{\boldsymbol{\omega}}) = \begin{bmatrix} \hat{\omega}_y\hat{\omega}_z & 0 & 0 \\ 0 & \hat{\omega}_x\hat{\omega}_z & 0 \\ 0 & 0 & \hat{\omega}_x\hat{\omega}_y \end{bmatrix}.$$

Thus

$$\begin{aligned}\boldsymbol{F} &= \frac{\partial \boldsymbol{f}_s(\boldsymbol{x})}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} \\ &= \begin{bmatrix} -[\hat{\boldsymbol{\omega}}\times] & \frac{1}{2}\mathbf{1}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 6} \\ \mathbf{0}_3 & \boldsymbol{M}(\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{p}}) & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{N}(\hat{\boldsymbol{\omega}}) & \mathbf{0}_{3\times 6} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{1}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 6} \\ \mathbf{0}_{12\times 3} & \mathbf{0}_{12\times 3} & \mathbf{0}_{12\times 3} & \mathbf{0}_{12\times 3} & \mathbf{0}_{12\times 3} & \mathbf{0}_{12\times 6} \end{bmatrix}.\end{aligned}$$

The equivalent discrete-time system of the linearized system can be written as

$$\delta\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k\delta\boldsymbol{x}_k + \boldsymbol{w}_k \tag{51}$$

where $\boldsymbol{w}_k$ is discrete-time process noise, and the state transition matrix $\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(t_k, t_\Delta)$ at time t_k over time interval $t_\Delta = t_{k+1} - t_k$ is given by

$$\boldsymbol{\Phi}(t_k, t_\Delta) = e^{\boldsymbol{F}(t_k)t_\Delta} \approx \mathbf{1}_{21} + t_\Delta\boldsymbol{F}(t_k). \tag{52}$$

The covariance of the discrete-time process noise is related to the continuous process noise covariance by

$$\boldsymbol{Q}_k = E[\boldsymbol{w}_k \boldsymbol{w}_k^T] = \int_{t_k}^{t_k + t_\Delta} \boldsymbol{\Phi}(t_k, \tau)\boldsymbol{G}\boldsymbol{\Sigma}\boldsymbol{G}^T\boldsymbol{\Phi}^T(t_k, \tau)\mathrm{d}\tau$$

where $\boldsymbol{\Sigma} = E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = \mathrm{diag}(\sigma_\tau^2 \mathbf{1}_3, \sigma_f^2 \mathbf{1}_3)$. After integration, the covariance matrix takes the form

$$\boldsymbol{Q}_k = \mathrm{diag}\big(\sigma_\tau^2 \boldsymbol{\Lambda}_1(t_k), \sigma_f^2 \boldsymbol{\Lambda}_2(t_k), \mathbf{0}_9\big) \tag{53}$$

where

$$\boldsymbol{\Lambda}_1 = \begin{bmatrix} \frac{t_\Delta^3}{12}\boldsymbol{B}^2 & \frac{t_\Delta^3}{6}\boldsymbol{B}^2\boldsymbol{M}^T + \frac{t_\Delta^2}{4}\boldsymbol{B}^2 \\ \times & \frac{t_\Delta^3}{3}\boldsymbol{M}\boldsymbol{B}^2\boldsymbol{M}^T + \frac{t_\Delta^2}{2}(\boldsymbol{B}^2\boldsymbol{M}^T + \boldsymbol{M}\boldsymbol{B}^2) + \boldsymbol{B}^2 t_\Delta \end{bmatrix}$$

$$\boldsymbol{\Lambda}_2 = \begin{bmatrix} \frac{t_\Delta^3}{3}\mathbf{1}_3 & \frac{t_\Delta^2}{2}\mathbf{1}_3 \\ \times & t_\Delta \mathbf{1}_3 \end{bmatrix}.$$

Now, one can design an EKF based on the derived models (47), (52), and (53). The EKF-based observer for the associated noisy discrete system (51) is given in two steps: 1) Estimate update

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^T \big(\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k\big)^{-1} \tag{54a}$$

$$\Delta\hat{\boldsymbol{x}}_k = \boldsymbol{K}\big(\boldsymbol{z}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-)\big) \tag{54b}$$

$$\boldsymbol{P}_k = \big(\mathbf{1}_{21} - \boldsymbol{K}_k \boldsymbol{H}_k\big)\boldsymbol{P}_k^- \tag{54c}$$

and 2) estimate propagation

$$\hat{\boldsymbol{x}}_{k+1}^- = \hat{\boldsymbol{x}}_k + \int_{t_k}^{t_k + t_\Delta} \boldsymbol{f}_s(\boldsymbol{x})\,\mathrm{d}t \tag{55a}$$

$$\boldsymbol{P}_{k+1}^- = \boldsymbol{\Phi}_k \boldsymbol{P}_k \boldsymbol{\Phi}_k^T + \boldsymbol{Q}_k \tag{55b}$$

where $\hat{\boldsymbol{x}}_k^-$ and $\hat{\boldsymbol{x}}_k$ are *a priori* and *a posteriori* estimations of the state vector, and $\boldsymbol{H}_k = \boldsymbol{H}(\hat{\boldsymbol{x}}_k^-)$. The state update follows the error-state update, i.e., $\Delta\hat{\boldsymbol{x}} = [\Delta\hat{\boldsymbol{q}}_v^T \; \Delta\hat{\boldsymbol{\chi}}^T \; \Delta\hat{\boldsymbol{\mu}}_v^T]^T$, in the *innovation step* of the KF (54b). The update of "nonquaternion part" of the state vector, i.e., $\hat{\boldsymbol{\chi}}$, can be easily obtained by adding the corresponding error to *a priori* estimation. However, quaternion updates are more complicated because only the vectors parts of the quaternion errors $\Delta\hat{\boldsymbol{q}}_{v_k}$ and $\Delta\hat{\boldsymbol{\mu}}_{v_k}$ are given at every step time. Therefore, first, the scalar parts of the quaternion variations should be computed from the corresponding vector parts. Second, the quaternion updates are computed using the quaternion multiplication rule. Consequently, the state update may proceed right after (54b) as

$$\text{state update:} \quad \begin{cases} \hat{\boldsymbol{\chi}}_k = \hat{\boldsymbol{\chi}}_k^- + \Delta\hat{\boldsymbol{\chi}}_k \\ \hat{\boldsymbol{q}}_k = \begin{bmatrix} \Delta\hat{\boldsymbol{q}}_{v_k} \\ \sqrt{1 - \|\Delta\hat{\boldsymbol{q}}_{v_k}\|^2} \end{bmatrix} \otimes \hat{\boldsymbol{q}}_k^- \\ \hat{\boldsymbol{\mu}}_k = \begin{bmatrix} \Delta\hat{\boldsymbol{\mu}}_{v_k} \\ \sqrt{1 - \|\Delta\hat{\boldsymbol{\mu}}_{v_k}\|^2} \end{bmatrix} \otimes \hat{\boldsymbol{\mu}}_k^- \end{cases} \tag{56}$$

V. Experiment

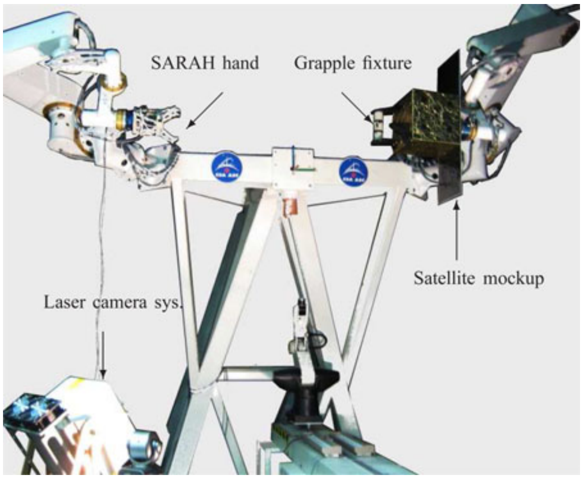

Fig. 4. Experimental setup.

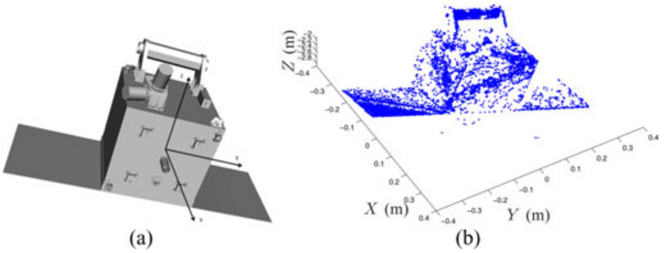

Fig. 5. ICP algorithm matches points from the CAD model and the scanned data to estimate the object pose. (a) Satellite CAD model. (b) Point-cloud data.

This section reports experimental results showing the performance of the vision-guided robotic system for autonomous interception of a tumbling satellite. Fig. 4 illustrates the experimental setup with two manipulator arms. One arm is employed to drift and tumble the mockup of a target satellite[2] according to dynamics motion in zero-gravity [75]. The other arm, which is equipped with a robotic hand known as SARAH [76], is used to autonomously approach the mockup and capture its grapple fixture. In this experiment, the servicer satellite to which the space manipulator is attached is assumed to have an ACS to perfectly cancel the dynamic effect of the manipulator motion on the attitude of the servicer base. Therefore, only the vision guidance and robot planning method can be verified without taking imperfect attitude control or space manipulator control into account. Each robotic arm has seven joints and, hence, is redundant. The redundancy resolution algorithm that is proposed by Nelson and Khosla [64] is used to control the self-motion of the manipulators in such a way that joint limits and singularities are avoided as much as possible. Neptec's LCS [52] is used to obtain the range data at a rate of 2 Hz. The target pose is estimated by comparing the range data obtained from the LCS against the CAD model of the mockup using the iterative closest point (ICP) algorithm [55], [77]. The ICP algorithm is an iterative procedure which minimizes a distance between point cloud in one dataset and the closest points in the other [78], [79]. One dataset is a set of 3-D point cloud acquired by scanning the target, while the other one is a CAD surface model as illustrated in Fig. 5.

The entire control system has been developed using Simulink. MATLAB's Real-Time Workshop [80] generates portable C

[2] A two-third scaled model of a Canadian micro-satellite called Quicksat.

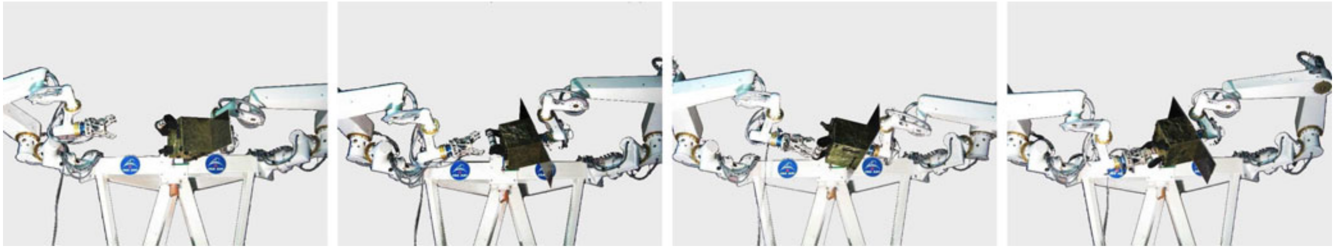

Fig. 6. Last steps of robotic grasping of the satellite mockup.

code from the Simulink model. Matrix manipulations are carried out by using the DSP blockset of Matlab/Simulink [81]. The code, upon compilation, is executed on the QNX real-time operating system. Four 2.5-GHz P4 computers are dedicated to the processes associated with the ICP, the EKF estimator, the robot path planner and control, and the satellite simulator.

The robotic arm to which the satellite mockup is attached is controlled to precisely mimic the dynamics motion of a target satellite. In other words, the satellite mockup is moved as if it is drifting and tumbling in zero-gravity. The moment of inertia values of the target satellite are selected as $I_{xx} = 4\,\mathrm{kg{\cdot}m^2}$, $I_{yy} = 8\,\mathrm{kg{\cdot}m^2}$, $I_{zz} = 5\,\mathrm{kg{\cdot}m^2}$, and the location of CM is $\boldsymbol{\rho} = [-0.15\ 0\ 0]^T$ m. Therefore, $\mathrm{tr}(\boldsymbol{I}_c) = 17\,\mathrm{kg{\cdot}m^2}$, and the inertia ratio parameters are

$$\boldsymbol{p} = \begin{bmatrix} -0.125 \\ -0.75 \\ 0.8 \end{bmatrix}.$$

These values are used to determine the manipulator-driven motion of the target as free-floating object with initial drift velocity $\dot{\boldsymbol{r}}(0) = [-3\ \ -2\ 1.5]$ mm/s and initial angular velocity $\boldsymbol{\omega}(0) = [0.09\ \ -0.04\ 0.03]$ rad/s. Random force and torque perturbation signals with covariances

$$\sigma_f = 1 \times 10^{-3}\mathrm{m/s^2} \quad \text{and} \quad \sigma_\tau = 2 \times 10^{-3}\mathrm{rad/s^2}$$

are fed to the simulator so that the produced motion is not completely deterministic. The motion trajectories of the target satellite are generated off-line using MATLAB Simulink.

The LCSs provides pose measurements of the object at 2 Hz. The measurement noise matrix, i.e., $\boldsymbol{R}$, for the vision system is

$$\boldsymbol{R} = \mathrm{diag}\big((0.045)^2\mathbf{1}_3\mathrm{m}^2, (0.06)^2\mathbf{1}_4\big).$$

For the first iteration of the EKF, an adequate guess of the initial states is required. The initial linear and angular velocities of the object at $t = 0$ s are selected to be zero. In addition, coordinate frame $\{\mathcal{C}\}$ is assumed to be initially aligned with the object's principal axes, i.e., $\hat{\boldsymbol{q}}(0) = \boldsymbol{q}(0) = \boldsymbol{\eta}(0)$. Given initial values of the parameters $\boldsymbol{p}(0)$ and $\boldsymbol{\rho}(0)$, the initial value of the CM position can be computed from the reciprocal of the observation equation, and thus, the state vector is initialized as

$$\boldsymbol{x}(0) = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{r}_s(0) - \boldsymbol{A}^T(\boldsymbol{\eta}(0))\boldsymbol{\rho}(0) \\ \mathbf{0} \\ \boldsymbol{p}(0) \\ \boldsymbol{\rho}(0) \\ \mathbf{0} \end{bmatrix}.$$

The objective is to capture the grapple fixture of the tumbling satellite mockup smoothly using the robotic arm despite of not knowing the target dynamics parameters. The parameters of the optimal motion planner are set to

$$w_1 = 2, \quad w_2 = 50, \quad \kappa = 1000, \quad a = 0.01. \tag{57}$$

The time history of the position and orientation of the mockup satellite obtained from ICP algorithm is shown in Fig. 7. Apparently, the vision system is obstructed by the robotic arm during its final approach at time $t = 112$ s, and thereafter, the visual pose measurements are incorrect. During the learning phase, the EKF estimator receives these noisy measurements and, subsequently, estimates the sets of inertial parameters plus the linear and angular velocities of the tumbling satellite. Figs. 8 and 9 show the trajectories of the estimated velocities and the inertial parameters, respectively. The motion planner automatically generates robot trajectory once the EKF estimator converges. To this end, the convergence is when the Euclidean norm of the state covariance matrix drops below a threshold ϵ_{th}, i.e.,

$$\|\boldsymbol{P}_k\| \le \epsilon_{\mathrm{th}}.$$

It should be pointed out that the threshold must be selected carefully: A high value can lead to inaccurate estimation results, while a very low value can lead to an unnecessarily long convergence time. The time history of the norm of the covariance matrix is plotted in Fig. 10. In this experiment, the filter covariance matrix is initialized as

$$\boldsymbol{P}(t_o) = \mathbf{1}_{21}$$

whereas the threshold value for the convergence detection is set to 0.1. Apparently, at time $t = 85$ s, the trajectory reaches the prescribed threshold $\epsilon_{\mathrm{th}} = 0.1$ indicating the filter convergence. The initial time, however, is set to $t_o = 90$ s, leaving a margin of 5 s to accommodate the computation time needed for path planning. Trajectories of predicted position, velocity, and acceleration of the grapple fixture calculated from the latest

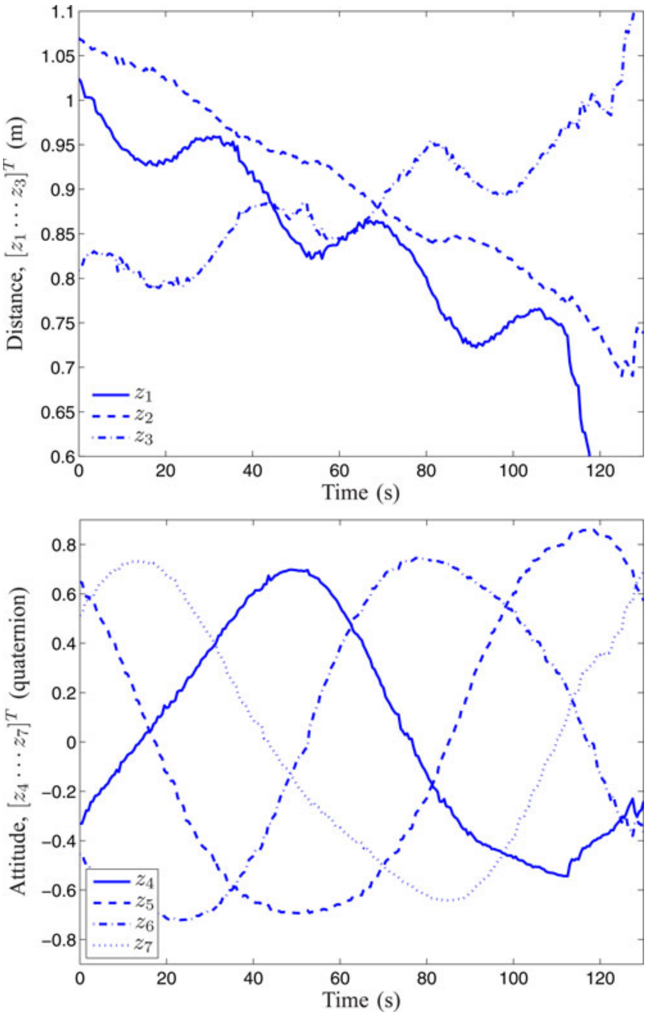

Fig. 7. Target pose measurements obtained from LCS.

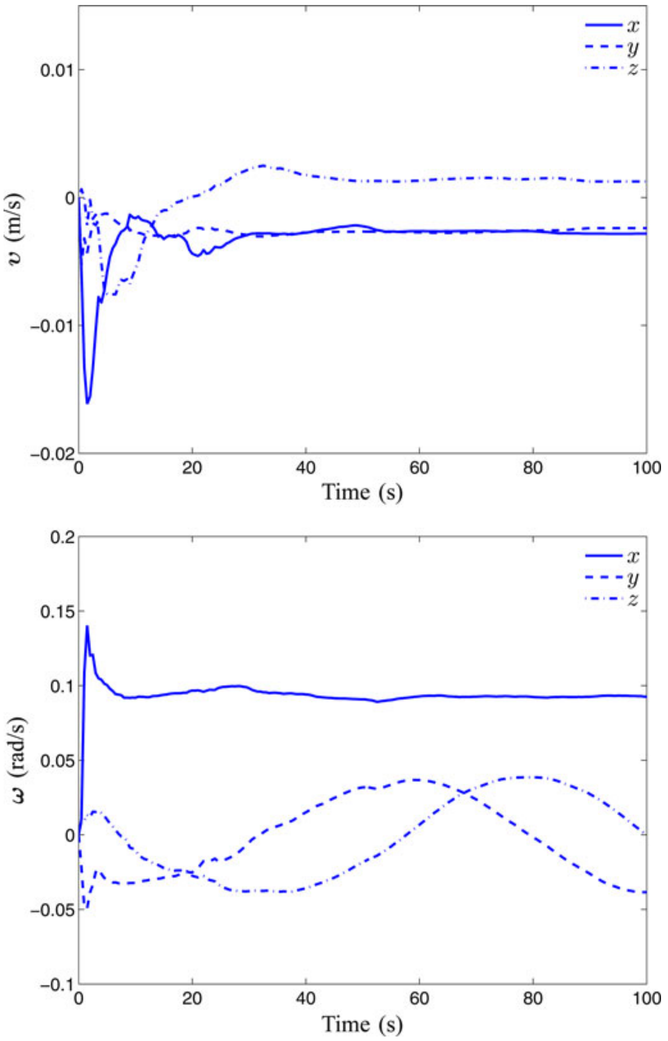

Fig. 8. Estimated drift velocity and angular velocity of the target.

estimation of the motion states and parameters of the target are shown in Fig. 11. A comparison of the trajectories of the visual measurements and those of the predicted position in Fig. 11(a) reveals that the visual measurements are not reliable at the final stages of the robot approach after time $t = 112$ s. The optimal rendezvous time t_f is obtained from solving the Hamiltonian as function of t_f in (30) using the predicted motion trajectories. The zero-crossing of the Hamiltonian is found numerically using the NR method as illustrated in Fig. 12. It is clear from the figure that the NR method quickly converges to optimal solution $t_f - t_o = 36.5$ s., i.e., within about eight iterations, in spite of far off initial guess 10 s from the actual value. Thus

$$t_o = 90 \text{ s} \quad \text{and} \quad t_f = 126.5 \text{ s}$$

is the rendezvous time minimizing the weighted linear sum of the objectives corresponding to the specified weights (57). For the sake of comparing the optimization results with respect to different objectives, the optimal rendezvous time for minimum-time path, minimum-distance path, and minimum line-of-sight angle are also calculated to be $t_f = 105$ s, $t_f = 120$ s, and $t_f = 132$ s, respectively, for this particular example.

The last steps of robotic capturing of the satellite's grapple fixture are illustrated in Fig. 6. Trajectories of the robot's end-effector and the satellite grapple fixture are shown in Fig. 13. It is evident that the robot intercepts the handle at the designated final time. Fig. 14 shows a phase plane showing trajectories of the distance between the end-effector and the grapple fixture, $\boldsymbol{r}_s - \boldsymbol{r}_r$ versus the relative velocity, $\dot{\boldsymbol{r}}_s - \dot{\boldsymbol{r}}_r$. It is clear from the plot that the end-effector has reached the grapple fixture with almost zero relative velocity. The optimal rendezvous point and path are planned based on the prediction of the target's motion computed from the estimated parameters and states at initial time $t_o = 90$ s. In order to assess the accuracy of motion prediction, the prediction motion error for the grasping interval $[t_o \; t_f]$ is estimated from the actual target position, i.e., $\boldsymbol{r}_{s_{\text{act}}}$, computed from the kinematics model of the simulator manipulator and the results are plotted in Fig. 15. It is evident from the graph that the positioning error at the time of grasping is less than 1.5 cm, which is well within the grasping envelop of our robotic hand. In addition, the predicted trajectories of the line-of-sight angle in Fig. 16 shows that the robot reaches the target at angle $\theta = 21°$, which is small angle for reliable capture.

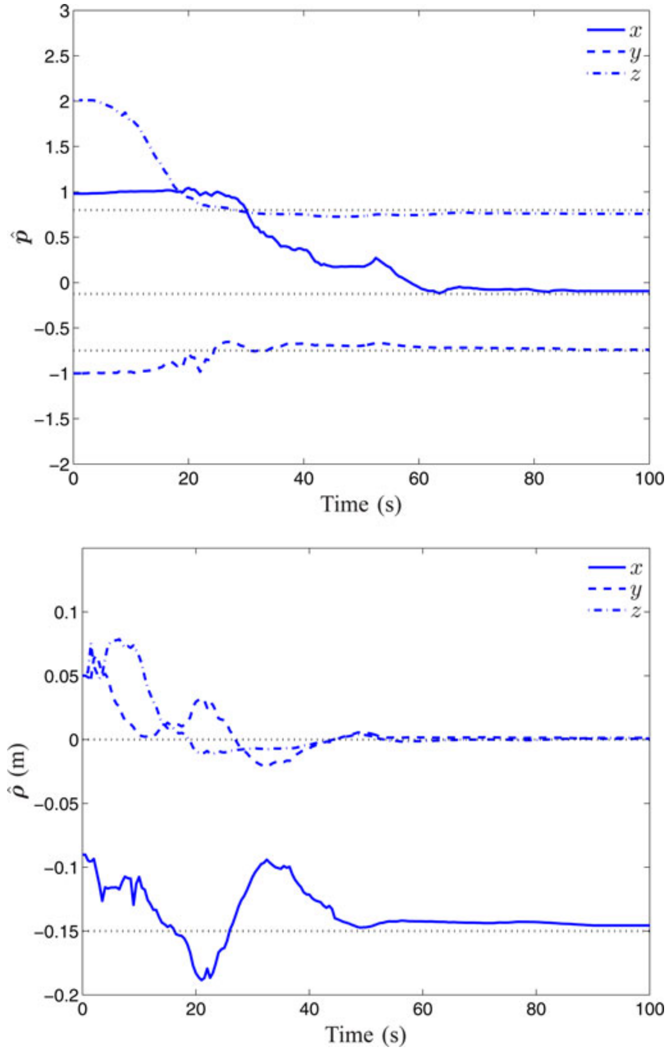

Fig. 9. Convergence of the estimated dynamics parameters of the target to their true values (depicted by dotted lines).

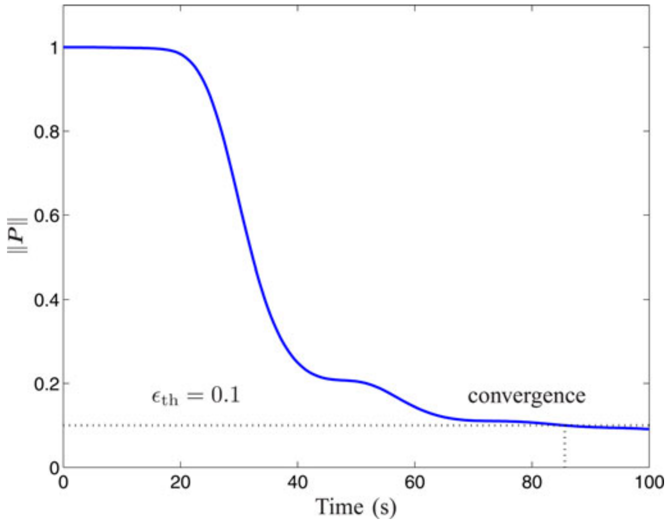

Fig. 10. Detecting the estimator convergence from the norm of the state covariance matrix.

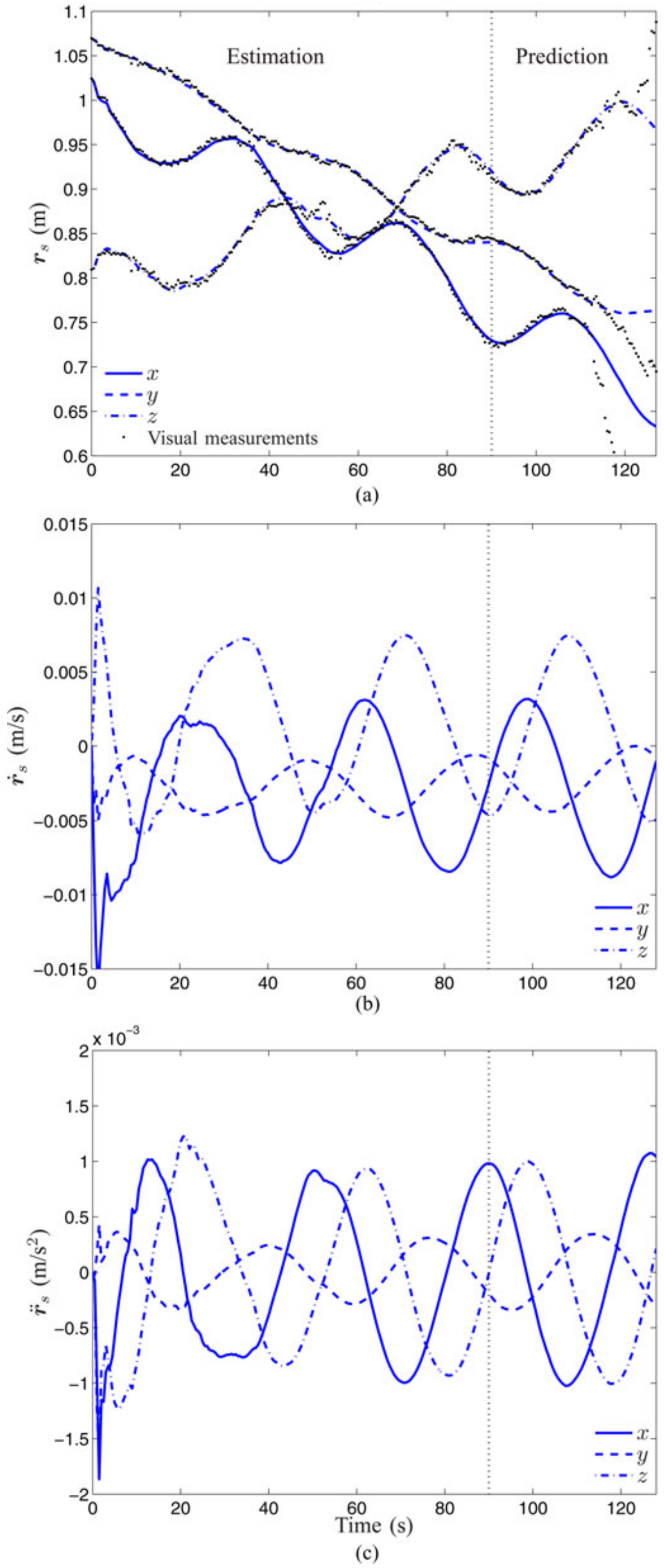

Fig. 11. (a)–(c) Estimations and predictions of the grapple-fixture position, velocity, and acceleration.

Robotic capturing of the target satellite also requires that the robot end-effector is properly aligned with the target at the time of rendezvous. In this experiment, the robot attempts to track

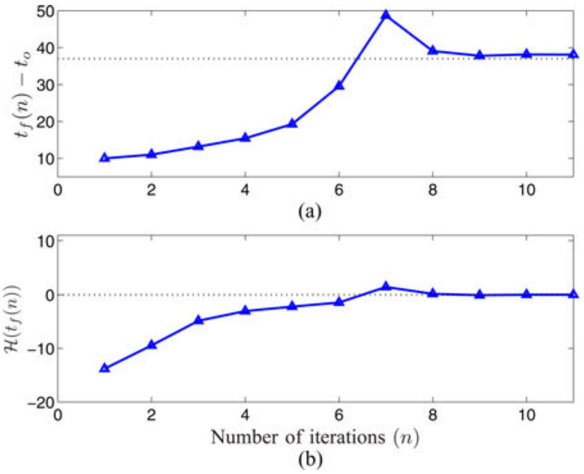

Fig. 12. NR iterations to find the zero crossing of the Hamiltonian function.

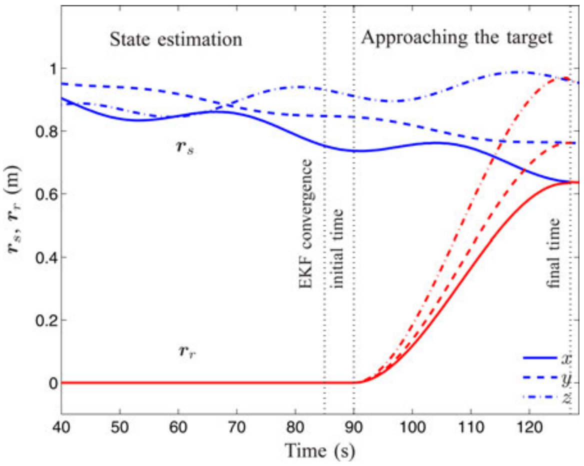

Fig. 13. Position trajectories of the target satellite and the robot hand.

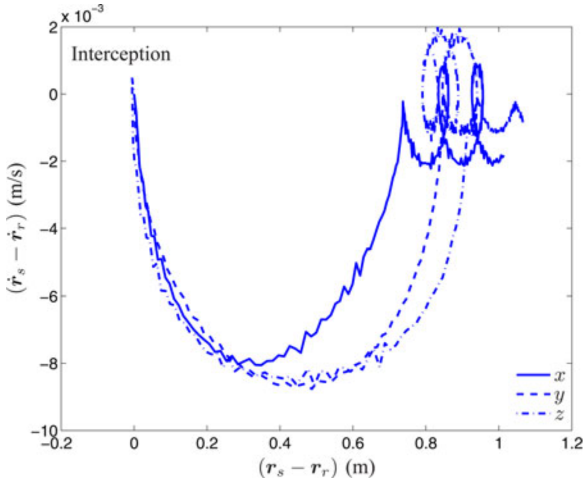

Fig. 14. Trajectories of the relative position versus relative velocity.

the target's orientation trajectories starting at initial time $t = t_o$. By virtue of (5), one can estimate the target angular acceleration from the estimated angular velocity and parameters according to $\hat{\dot{\boldsymbol{\omega}}} = \boldsymbol{d}(\hat{\boldsymbol{\omega}}, \hat{\boldsymbol{p}})$. Fig. 17 illustrates trajectories of the estimated

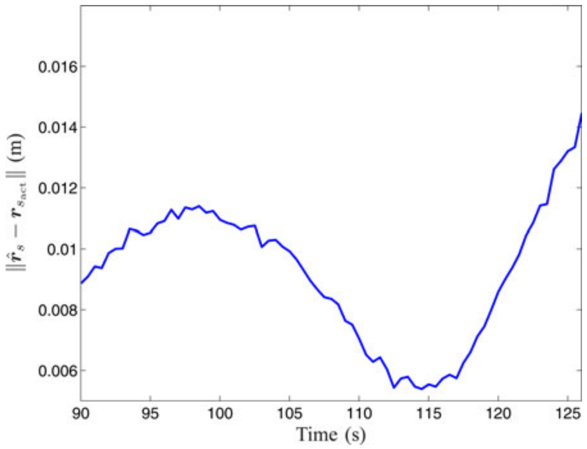

Fig. 15. Prediction error of the grapple-fixture position.

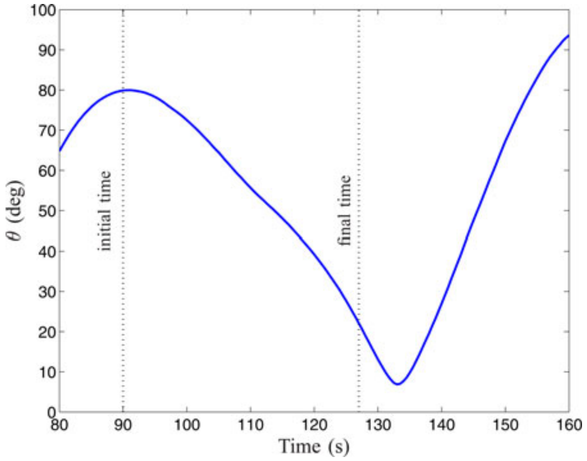

Fig. 16. Line-of-sight angle.

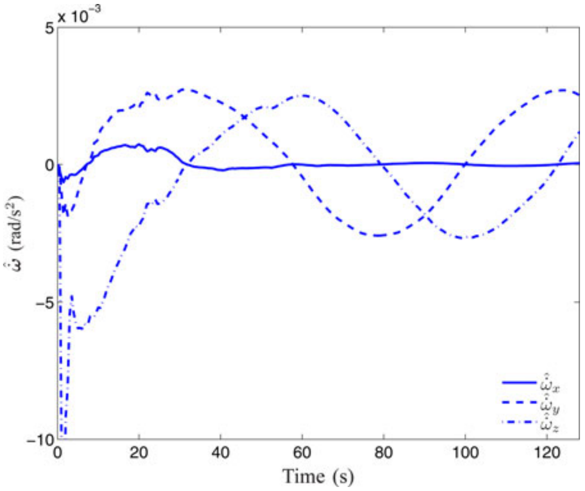

Fig. 17. Estimation of the target's angular acceleration.

angular acceleration. Consequently, $\hat{\boldsymbol{q}}$, $\hat{\boldsymbol{\omega}}$, and $\hat{\dot{\boldsymbol{\omega}}}$ constitute the desired orientation trajectories of the robot. Fig. 18 demonstrates that the robot end-effector is able to track the target attitude few seconds after the initial time t_o.

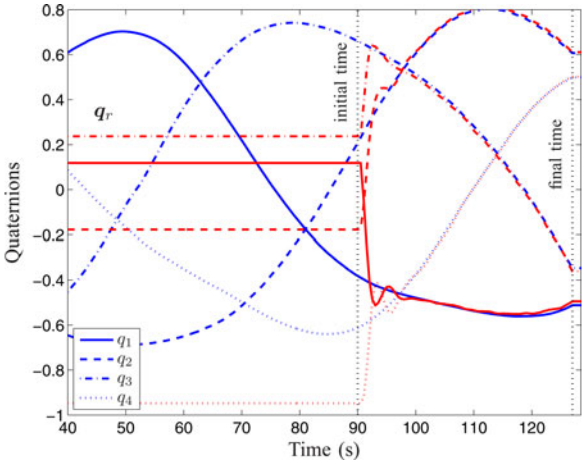

Fig. 18. Orientation trajectories of the target satellite and robot's hand.

VI. Conclusion

We presented an optimal estimation and guidance scheme in detail for smooth robotic capturing of a drifting and tumbling target with uncertain dynamics. First, a KF was used to estimate the motion states and a set of dynamics parameters of the target based on noisy pose measurements of an LCS. The dynamics equations were adequately represented in terms of inertia ratios, and trace of the inertia matrix (an invariant quantity) is factored out of the equations so that the set of remaining dynamics parameters became identifiable. Second, the estimations of the states and parameters are used by a guidance scheme to optimally plan the trajectory of the robot's end-effector to intercept a grapple fixture on the object with zero relative velocity at an optimal rendezvous point. In order to satisfy the capture requirements, a cost function was defined as a weighted linear sum of travel time, distance, and cosine of a line-of-sight angle (indicating the accessibility of the grapple fixture for robotic capturing) plus a penalty function acting as a "soft" constraint on the acceleration magnitude. Optimal control theory was used to find the trajectory and rendezvous point minimizing the cost function subject to the zero relative velocity requirements at the time of grasping. Finally, experimental results illustrating autonomous guidance of a robotic manipulator for the capture of a drifting and tumbling mockup satellite based on noisy pose measurements obtained from a laser vision system were reported. The results demonstrated that the robot's hand smoothly intercepts the grapple fixture of the tumbling satellite, i.e., with almost zero relative velocity, even though the vision sensor was obstructed by the robot arm and that target dynamics was not known *a priori* to the guidance system.

Appendix A
Euler Equations

The Euler equations written in the principal axes are given by

$$\begin{aligned} I_{xx}\dot{\omega}_x &= (I_{yy} - I_{xx})\omega_y\omega_z + \tau_x \\ I_{yy}\dot{\omega}_y &= (I_{zz} - I_{xx})\omega_x\omega_z + \tau_y \\ I_{zz}\dot{\omega}_z &= (I_{xx} - I_{yy})\omega_x\omega_y + \tau_z \end{aligned} \tag{58}$$

which are equivalent to (5) using definition (7), and

$$b_x = \frac{\mathrm{tr}(\boldsymbol{I}_c)}{I_{xx}}, \qquad b_y = \frac{\mathrm{tr}(\boldsymbol{I}_c)}{I_{yy}}, \qquad b_z = \frac{\mathrm{tr}(\boldsymbol{I}_c)}{I_{zz}}. \tag{59}$$

The following identities can be directly derived from (4):

$$\frac{I_{yy}}{I_{xx}} = \frac{1+p_x}{1-p_y}, \quad \frac{I_{zz}}{I_{xx}} = \frac{1-p_x}{1+p_z}, \quad \frac{I_{yy}}{I_{zz}} = \frac{1-p_z}{1+p_y}. \tag{60}$$

Finally, using (60) in (59) gives

$$\begin{aligned} b_x(\boldsymbol{p}) &= 1 + \frac{I_{yy}}{I_{xx}} + \frac{I_{zz}}{I_{xx}} \\ &= 1 + \frac{1+p_x}{1-p_y} + \frac{1-p_x}{1+p_z}. \end{aligned}$$

The expressions of b_y and b_z as functions of $\boldsymbol{p}$ can be derived in the same fashion.

Appendix B
Newton–Raphson

According to the NR method, the following loop

$$t_{n+1} = t_n - \frac{\mathcal{H}(t_n)}{\mathcal{H}'(t_n)} \tag{61}$$

may be worked out iteratively until the error in equation $\mathcal{H}(t) = 0$ falls into an acceptable tolerance. The differentiation of the Hamiltonian with respect to time in (61) can be approximated from

$$\mathcal{H}'(t_n) \approx \frac{\mathcal{H}(t_n) - \mathcal{H}(t_{n-1})}{t_n - t_{n-1}}. \tag{62}$$

Therefore, upon substitution of (62) into (61), we get

$$t_{n+1} = \frac{t_n\mathcal{H}(t_{n-1}) - t_{n-1}\mathcal{H}(t_n)}{\mathcal{H}(t_{n-1}) - \mathcal{H}(t_n)}. \tag{63}$$

Acknowledgment

The author would like to thank Dr. L. Hartman from Canadian Space Agency for his helpful comments to this paper.

References

[1] Z. Luo and Y. Sakawa, "Control of a space manipulator for capturing a tumbling object," in *Proc. 29th IEEE Conf. Decis. Control*, Dec. 1990, vol. 1, pp. 103–108.

[2] E. Papadopoulos and S. Moosavian, "Dynamics and control of multi-arm space robots during chase and capture operations," in *Proc. IEEE/RSJ/GI Int. Conf. Intell. Robots Syst. Adv. Robot. Syst. Real World*, Sep. 1994, vol. 3, pp. 1554–1561.

[3] D. Zimpfer and P. Spehar, "STS-71 Shuttle/MIR GNC mission overview," in *Proc. Adv. Astron. Sci., Amer. Astron. Soc.*, San Diego, CA, 1996, pp. 441–460.

[4] H. Nagamatsu, T. Kubota, and I. Nakatani, "Capture strategy for retrieval of a tumbling satellite by a space robotic manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr., 1996, vol. 1, pp. 70–75.

[5] G. Visentin and D. L. Brown, "Robotics for geostationary satellite service," in *Proc. Robot. Auton. Syst.*, 1998, vol. 23, pp. 45–51.

[6] S. Nakasuka and T. Fujiwara, "New method of capturing tumbling object in space and its control aspects," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kohala Coast, HI, Aug.1999, pp. 973–978.

[7] S. Matsumoto, Y. Ohkami, Y. Wakabayashi, M. Oda, and H. Uemo, "Satellite capturing strategy using agile orbital servicing vehicle, hyper OSV," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, May. 2002, pp. 2309–2314.

[8] E. Bornschlegl, G. Hirzinger, M. Maurette, R. Mugunolo, and G. Visentin, "Space robotics in Europe, a compendium," presented at the 7th Int. Symp. Artif. Intell., Robot., Autom. Space, Nara, Japan, May. 2003.

[9] G. Hirzinger, K. Landzettel, B. Brunner, M. Fisher, C. Preusche, D. Reintsema, A. Albu-Schaffer, G. Schreiber, and B.-M. Steinmetz, "DLR's robotics technologies for on-orbit servicing," *Adv. Robot.*, vol. 18, no. 2, pp. 139–174, 2004.

[10] Z. Ma, O. Ma, and B. Shashikanth, "Optimal approach to and alignment with a rotating rigid body for capture," *J. Astron. Sci.*, vol. 55, no. 4, pp. 407–419, 2007.

[11] K. Yoshida, D. Dimitrov, and H. Nakanishi, "On the capture of tumbling satellite by a space robot," presented at the IEE/RSJ Int. Conf. Intell. Robots Syst., Beijing, China, Oct. 2006.

[12] C. Li, B. Liang, and W. Xu, "Autonomous trajectory planning of free-floating robot for capturing space target," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 1008–1013.

[13] I. Rekleitis, E. Martin, G. Rouleau, R. L'Archevêque, K. Parsa, and E. Dupuis, "Autonomous capture of a tumbling satellite," *J. Field Robot., Spec. Issue Space Robot.*, vol. 24, no. 4, pp. 275–296, 2007.

[14] F. Aghili and K. Parsa, "Adaptive motion estimation of a tumbling satellite using laser-vision data with unknown noise characteristics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, Oct. 29–Nov. 2, 2007, pp. 839–846.

[15] D. Whelan, E. Adler, S. Wilson, and G. Roesler, "Darpa orbital express program: Effecting a revolution in space-based systems," in *Proc. Small Payloads Space*, Nov. 2000, vol. 136, pp. 48–56.

[16] J. K. Thienel and R. M. Sanner, "Hubble space telescope angualar velocity estimation during the robotic servicing mission," *AIAA J. Guid., Control, Dyn.*, vol. 30, no. 1, pp. 29–34, Feb. 2007.

[17] S. Kawamoto, S. Nishida, and S. Kibe, "Research on a space debris removal system," *NAL Res. Prog. (Nat. Aerosp. Lab. Jpn.)*, vol. 2002/2003, pp. 84–87, 2003.

[18] D. Reintsema, J. Thaeter, A. Rathke, W. Naumann, P. Rank, and J. Sommer, "DEOS—The German robotics approach to secure and de-orbit malfunctioned satellites from low earth orbit," in *Proc. Int. Symp. Artif. Intell., Robot. Autom. Space*, Sapporo, Japan, Aug. 29–Sep. 1, 2010, pp. 244–251.

[19] G. Brachet, "Long-term sustainability of outer space activities," Committee on the Peaceful Uses of Outer Space (UNCOPUOS), Vienna, Austria, Tech. Rep., Feb. 8–19, 2010.

[20] J.-C. Liou, N. Johnson, and N. Hill, "Controlling the growth of future LEO debris populations with active debris removal," *Acta Astronautica*, vol. 66, no. 5–6, pp. 648–653, Mar./Apr. 2009.

[21] B. Barbee, S. Alfano, E. Pinon, K. Gold, and D. Gaylor, "Design of spacecraft missions to remove multiple orbital debris objects," in *Proc. IEEE Aerosp. Conf.*, Mar. 2011, pp. 1–14.

[22] M. Oda, "Space robot experiments on NASDA's ETS-VII satellite—Preliminary overview of the experiment results," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, 1999, pp. 1390–1395.

[23] N. Inaba and M. Oda, "Autonomous satellite capture by a space robot: World first on-orbit experiment on a Japanese robot satellite ETS-VII," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, pp. 1169–1174.

[24] K. Yoshida, "Engineering test satellite VII flight experiment for space robot dynamics and control: Theories on laboratory test beds ten years ago, now in orbit," *Int. J. Robort. Res.*, vol. 22, no. 5, pp. 321–335, 2003.

[25] *On-Orbit Satellite Servicing Study, Project Report*, Nat. Aeronaut. Space Admin., Goddard Space Flight Center, Greenbelt, MD, 2010. Available: http://servicingstudy.gsfc.nasa.gov/

[26] F. Aghili, "Optimal control of a space manipulator for detumbling of a target satellite," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, May.2009, pp. 3019–3024.

[27] X. Cyril, A. K. Misra, M. Ingham, and G. Jaar, "Postcapture dynamics of a spacecraft-manipulator-payload system," *AIAA J. Guid., Control, Dyn.*, vol. 23, no. 1, pp. 95–100, Jan./Feb. 2000.

[28] D. N. Nenchev and K. Yoshida, "Impact analysis and post-impact motion control issues of a free-floating space robot subject to a force impulse," *IEEE Trans. Robot. Autom.*, vol. 15, no. 3, pp. 548–557, Jun. 1999.

[29] Z. Vafa and S. Dubowsky, "On the dynamics of manipulators in space using the virtual manipulator approach," in *Proc. IEEE Int. Conf. Robot. Autom*, Raleigh, NC, pp. 579–585.

[30] Y. Umetani and K. Yoshida, "Resolved motion rate control of space manipulators with generalized Jacobian matrix," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 303–314, Jun. 1989.

[31] E. Papadopoulos and S. Dubowsky, "On the nature of control algorithms for space manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1990, pp. 1102–1108.

[32] K. Yoshida and K. Hashizume, "Zero reaction maneuver: Flight validation with ETS-VII space robot and extention to kinematically redundant arm," in *Proc. IEEE Int. Conf. Robot. Autom.*, Seoul, Korea, May 2001, pp. 441–446.

[33] T. Oki, H. Nakanishi, and K. Yoshida, "Time-optimal manipulator control of a free-floating space robot with constraint on reaction torque," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nice, France, Sep.2008, pp. 2828–2833.

[34] F. Aghili, "Cartesian control of space manipulators for on-orbit servicing," presented at the AIAA Guid., Navigat. Control Conf., Toronto, ON, Canada, Aug. 2010.

[35] Z. Lin, V. Zeman, and R. V. Patel, "On-line robot trajectory planning for catching a moving object," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1989, vol. 3, pp. 1726–1731.

[36] R. Sharma, J. Y. Herve, and P. Cuka, "Dynamic robot manipulation using visual tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, Nice, France, May.1992, pp. 1844–1849.

[37] M. Lei and B. Ghosh, "Visually guided robotic tracking and grasping of a moving object," in *Proc. 32nd IEEE Conf. Decis. Control*, Dec. 1993, vol. 2, pp. 1604–1609.

[38] Y. Chen and L. T. Watson, "Optimal trajectory planning for a space robot docking with a moving target via homotopy algorithms," *J. Robot. Syst.*, vol. 12, no. 8, pp. 531–540, 1995.

[39] M. Zhang and M. Buehler, "Sensor-based online trajectory generation for smoothly grasping moving objects," in *Proc. IEEE Int. Symp. Intell. Control*, Aug., 1994, pp. 141–146.

[40] N. Fitz-Coy and M. C. Liu, "New modified proportional navigation scheme for rendezvous and docking with tumbling targets: The planar case," in *Proc. Symp. Flight Mech./Estimat. Theory*, May 1995, pp. 243–252.

[41] E. A. Croft, R. G. Fenton, and B. Benhabib, "Optimal rendezvous-point selection for robotic inteception of moving objects," *IEEE Trans. Syst., Man, Cybern.*, vol. 28, no. 2, pp. 192–204, Apr. 1998.

[42] M. Mehrandezh, N. M. Sela, R. G. Fenton, and B. Benhabib, "Robotic interception of moving objects using an augmented ideal proportional navigation guidance technique," *IEEE Trans. Syst., Man, Cybern.*, vol. 30, no. 3, pp. 238–250, May 2000.

[43] F. Aghili and K. Parsa, "An adaptive vision system for guidance of a robotic manipulator to capture a tumbling satellite with unknown dynamics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nice, France, Sep. 2008, pp. 3064–3071.

[44] F. Aghili, "Optimal control for robotic capturing and passivation of a tumbling satellite," presented at the AIAA Guid., Navigat. Control Conf., Honolulu, HI, Aug. 2008.

[45] R. Lampariello, "Motion planning for the on-orbit grasping of an on-cooperative target satellite with c ollision avoidance," in *Proc. Int. Symp. Artif. Intell., Robot. Autom. Space*, Sapporo, Japan, Aug. 29–Sep. 12010, pp. 636–643.

[46] J. R. Wertz and R. Bell, "Autonomous rendezvous and docking technologies—Status and prospects," in *Proc. Int. Soc. Opt. Eng.*, Apr. 2003, vol. 5088, pp. 20–30.

[47] M. Ruth and C. Tracy, "Video-guidance for DART rendezvous mission," in *Proc. Int. Soc. Opt. Eng.*, Aug. 2004, vol. 5419, pp. 92–106.

[48] M. F. Machula and G. S. Sandhoo, "Rendezvous and docking for space exploration," presented at the 1st Space Explorat. Conf.: Contin. Voyage Discov., Orlando, FL, Jan. 2005.

[49] J. W. Evans, E. Piñon III, and T. A. Mulder, "Autonomous rendezvous guidance and navigation for orbital express and beyond," in *Proc. 16th AAS/AIAA Space Flight Mech. Conf.*, Jan. 2006, pp. 2–7.

[50] P. Singla, K. Subbarao, and J. L. Junkins, "Adaptive output feedback control for spacecraft rendezvous and docking under measurement uncertainty," *AIAA J. Guid., Control, Dyn.*, vol. 29, no. 4, pp. 892–902, 2006.

[51] G. Rouleau, I. Rekleitis, R. L'Archevêque, E. Martin, K. Parsa, and E. Dupuis, "Autonomous capture of a tumbling satellite," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, May 15–19, 2006, pp. 3885–3860.

[52] C. Samson, C. English, A. Deslauriers, I. Christie, F. Blais, and F. Ferrie, "Neptec 3D laser camera system: From space mission STS-105 to terrestrial applications," *Can. Aeronaut. Space J.*, vol. 50, no. 2, pp. 115–123, 2004.

[53] M. D. Lichter and S. Dubowsky, "State, shape, and parameter estimation of space object from range images," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2004, pp. 2974–2979.

[54] U. Hillenbrand and R. Lampariello, "Motion and parameter estimation of a free-floating space object from range data for motion prediction," in

Proc. 8th Int. Symp. Artif. Intell., Robot. Autom. Space, Munich, Germany, Sep. 5–8, 2005.

[55] F. Aghili, M. Kuryllo, G. Okouneva, and C. English, "Fault-tolerant pose estimation of space objects," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, Montreal, QC, Canada, Jul. 2010, pp. 947–954.

[56] R. F. Stengel, *Optimal Control and Estimation*. New York: Dover, 1993.

[57] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *J. Guid., Control, Dyn.*, vol. 5, no. 5, pp. 417–429, Sep./Oct. 1982.

[58] M. E. Pittelkau, "Kalman filtering for spacecraft system alignment calibration," *AIAA J. Guid. Contro. Dyn.*, vol. 24, no. 6, pp. 1187–1195, Nov. 2001.

[59] K.-S. Chang and O. Khatib, "Manipulator control at kinematic singularities: A dynamically consistent strategy," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Human Robot Interact. Cooperat. Robots*, Aug. 1995, vol. 3, pp. 84–88.

[60] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *J. Dyn., Sys., Control*, vol. 108, pp. 163–172, Apr. 1986.

[61] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *J. Intell. Robot. Syst.*, vol. 3, no. 3, pp. 201–212, 1989.

[62] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 398–410, Jun. 1997.

[63] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Reading, MA: Addison-Wesley, 1991.

[64] B. Nelson and P. K. Khosla, "Increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1993, vol. 3, pp. 418–423.

[65] R. V. Patel, F. Shadpey, F. Ranjbaran, and J. Angeles, "A collision-avoidance scheme for redundant manipulators: Theory and experiments," *J. Robot. Syst.*, vol. 22, pp. 737–757, 2005.

[66] S. Dubowsky and E. Papadopoulos, "The kinematics, dynamics, and control of free-flying and free-floating space robotic systems," *IEEE Trans. Robot. Autom.*, vol. 9, no. 5, pp. 531–543, Oct. 1993.

[67] F. Caccavale and B. Siciliano, "Kinematic control of redundant free-floating robotic systems," *Adv. Robot.*, vol. 15, no. 4, pp. 429–448, 2001.

[68] G. H. R. Krenn, "Modular generic inverse kinematics algorithms applied to kinematically redundant space manipulators," presented at the 8th ESA Workshop Adv. Space Technol. Robot. Autom., Noordwijk, The Netherlands, 2004.

[69] S. Cocuzza, I. Pretto, and S. Debei, "Novel reaction control techniques for redundant space manipulators: Theory and simulated microgravity tests," *Acta Astronautica*, vol. 68, no. 11, pp. 1712–1721, 2010.

[70] B. D. O. Anderson and J. B. Moore, *Optimal Control*. Englewood Cliffs, NJ: Prentice-Hall, 1990.

[71] C. D. Maranas and C. A. Floudas, "Finding all solutions of nonlinearly constrained systems of equations," *J. Global Optim.*, vol. 7, no. 2, pp. 143–182, 1995.

[72] S. K. Rahimiana, F. Jalali, J. D. Seaderc, and R. E. Whitea, "A new homotopy for seeking all real roots of a nonlinear equation," *Comput. Chem. Eng.*, vol. 35, no. 3, pp. 403–411, Mar. 2011.

[73] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. vol. 2, New York: Academic, 1982.

[74] C. K. Chui and G. Chen, *Kalman Filtering with Real-Time Applications*. Berlin, Germany: Springer-Verlag, 1998.

[75] F. Aghili and M. Namvar, "Scaling inertia properties of a manipulator payload for 0-g emulation of spacecraft," *Int. J. Robot. Res.*, vol. 28, no. 7, pp. 883–894, Jul. 2009.

[76] T. Laliberte and C. M. Gosselin, "Underactuation in space robotic hands," presented at the 6th Int. Symp. Artif. Intell., Robot. Autom. Space: New Space Odyssey, Montreal, QC, Canada, Jun. 2001.

[77] S. Ruel, C. English, M. Anctil, and P. Church, "LASSO: Real-time pose estimation from 3D data for autonomous satellite servicing," presented at the 8th Int. Symp. Artif. Intell., Robot. Autom. Space, Munich, Germany, Sep. 2005.

[78] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[79] D. A. Simon, M. Herbert, and T. Kanade, "Real-time 3-D estimation using a high-speed range sensor," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Diego, CA, May.1994, pp. 2235–2241.

[80] *Real-Time Workshop for Use With Simulink*, 5th ed. The MathWorks, Inc., Natick, MA, 2002.

[81] *DSP Blockset for Use With Simulink, User's Guide*. The MathWorks, Inc., Natick, MA, 1998.

Farhad Aghili (SM'07) received the B.Sc. degree in mechanical engineering and the M.Sc. degree in biomedical engineering from the Sharif University of Technology, Tehran, Iran, in 1988 and 1991, respectively, and the Ph.D. degree in mechanical engineering from McGill University, Montreal, QC, Canada, in 1998.

In January 1998, he joined the Canadian Space Agency (CSA), Saint-Hubert, QC, where he is currently a Research Scientist with the Space Exploration Directory. He has been working on the international space station program, robotic on-orbit servicing, and planetary exploration missions. He has published more than 120 papers in journals and conference proceedings and is the author of four patents in the U.S. and Canada. His research interests include multibody dynamics, robotics, space robotics, vision systems, control systems, and mechatronics systems.

Dr. Aghili won the Best Paper Award at the 2007 ASME/IEEE International Conference on Mechatronic and Embedded Systems and received the CSA Inventor and Innovator Certificate six times.