(Preprint) AAS 23-212

FUZZY-AIDED CLOSED-LOOP INVERSE KINEMATICS CONTROL FOR CAPTURING A TUMBLING SATELLITE

Sathya Karthikeyan,[*] Anirudh Chhabra,[†] Daegyun Choi,[‡] and Donghoon Kim[§]

Commercialization of the space industry is leading to the saturation of orbital carrying capacity. With the expected growth rate of satellites, the problem is only bound to get worse. Therefore, the capture of space debris is of great significance. This paper proposes a fuzzy-aided closed-loop inverse kinematics control methodology for the capture of a tumbling target satellite. For this, a space manipulator system comprising a 3 degrees-of-freedom (DOF) servicer spacecraft and a 2 DOF robotic manipulator arm is considered, and the dynamics of the system is designed and verified. Further, the proposed method is validated using a simulation study that considers the dynamics of the manipulator, servicer spacecraft, and target satellite.

INTRODUCTION

As space technologies advance, there has been an increase in the number of spacecraft and satellites being launched into space for various applications like deep space exploration,[1] earth surface monitoring,[2,3] communication,[4] navigation,[5] in-space assembly[6], etc. These missions are highly dependent on the health condition of the components used in these space systems.[7] To extend the life of satellites and spacecraft, it is necessary to perform maintenance operations on the orbit itself.[8] In order to perform on-orbit service missions on non-co-operative satellites, the target should be captured. However, this capture operation is rather challenging and therefore is generally broken down into three phases. The first is the orbital rendezvous phase, where the orbital motion of the target satellite is tracked and the distance between the servicer spacecraft and the target satellite is reduced. The second is the capture phase in which the tumbling motion of the target satellite is tracked using the manipulator for safely capturing the target satellite. Finally, the post-capture phase deals with minimizing the inertial impact of the tumbling target on the servicer spacecraft and stabilizing the target satellite.

The desirability of using space manipulators for on-orbit capturing has invoked research interests in various aspects. One of the most important missions is to capture a tumbling satellite, perform detumbling, and ensure stability in orbit. This tumbling behavior can be caused by several reasons

[*]Graduate Student, Department of Aerospace Engineering & Engineering Mechanics, University of Cincinnati, Cincinnati, OH 45221, USA, karthist@mail.uc.edu

[†]Graduate Student, Department of Aerospace Engineering & Engineering Mechanics, University of Cincinnati, Cincinnati, OH 45221, USA, chhabrad@mail.uc.edu

[‡]Graduate Student, Department of Aerospace Engineering & Engineering Mechanics, University of Cincinnati, Cincinnati, OH 45221, USA, choidg@mail.uc.edu

[§]Assistant Professor, Department of Aerospace Engineering & Engineering Mechanics, University of Cincinnati, Cincinnati, OH 45221, USA, kim3dn@ucmail.uc.edu

such as the loss of attitude control, failure of the attitude determination system on the target satellite, or as a result of collisions. Several relevant studies have been done in capturing a tumbling satellite.[9] Inaba and Oda[10] presented the world's first on-orbit experiment focusing on capturing a free-floating target using a manipulator mounted on a space shuttle. This experiment was conducted on the Japanese robot satellite ETS-VII using image processing and marker detection, followed by controlling the manipulator to track a path to the target, and then successfully capturing the target. Papadopoulos and Moosavian[11] studied the motion control of a multiple-arm space robot to track the motion of and capture a passive object in its proximity. They derived two dynamic models and validated the performance using three different control laws. Nakasuka and Fujiwara[12] studied an important aspect of this problem. They posited that most conventional methods focus only on the movement of the manipulator while maintaining a constant position and attitude of the servicer spacecraft itself, thereby limiting the rotational capability of the manipulator to capture the tumbling satellite. Xu et al.[13] proposed an autonomous motion control approach to capture a target using the coordinated motion of a dual-arm space robot. They presented two ways to achieve this mission: 1) using both arms for the coordinated capture, or 2) using one arm for capturing and the other for maintaining the servicer spacecraft at inertial rest. They further divided the resulting model into sub-equations to solve them using practical and simple solutions. It can be seen from this study that, in general, the use of a multi-arm manipulator is undesirable. This is because the reaction wheels on the servicer spacecraft may not be able to absorb the momentum generated by the motion of the manipulator leading to instability. Aghili[14] proposed the use of an optimal controller to capture a tumbling satellite by minimizing the operation time and the relative velocity between the robot tip and the target. However, the uncertainties in the model parameters were not considered. To address the problem of both large position and attitude deviations of the tumbling target, an autonomous capture framework was proposed by Jiao et al.[15] to capture different parts of a tumbling target by a dual-arm manipulator using predictive motion control. Cyril et al.[16] investigated the dynamics and control of a spacecraft-mounted robotic manipulator for capturing a target satellite. It primarily focuses on the post-capture effects of capturing a spinning target satellite on the dynamics of the manipulator and the servicer spacecraft. Cong et al.[17] investigated proper manipulator motion or configuration to minimize the base attitude disturbance during impact. This method seemed promising to minimize the base attitude disturbance during target capture. Most conventional control techniques are not efficient when dealing with complex systems in uncertain environments, whereas intelligent control is designed to adapt, learn, and plan under uncertainties. Kirk et al.[18] studied the use of artificial intelligence techniques to perform a robotic arm capture of a spinning target. Deep reinforcement learning was used to learn a guidance strategy to capture and simultaneously stabilize the target. However, studies on the use of intelligent control techniques to capture a tumbling satellite are scarce.

This study focuses on a single-arm manipulator mounted on top of a servicer spacecraft for capturing a target satellite, called a space manipulator system (SMS). The concept of operation is shown in Figure 1, where the target satellite is undergoing tumbling motion, and the SMS aims to capture the target by successfully attaching the robotic arm end-effector to the capture point on the target satellite. A motion planning and control methodology is proposed to maneuver the single-arm manipulator for capturing the target satellite. The SMS is a 5 degrees-of-freedom (DOF) system including 3 DOF for the rotational motion of the servicer spacecraft and 2 DOF for the robotic manipulator. The closed-loop inverse kinematics (CLIK) control method is a holistic control technique utilized for redundant manipulators.[19] However, it is often dependent on control gains that must be tuned to obtain the best performance and satisfaction of constraints. This makes the process

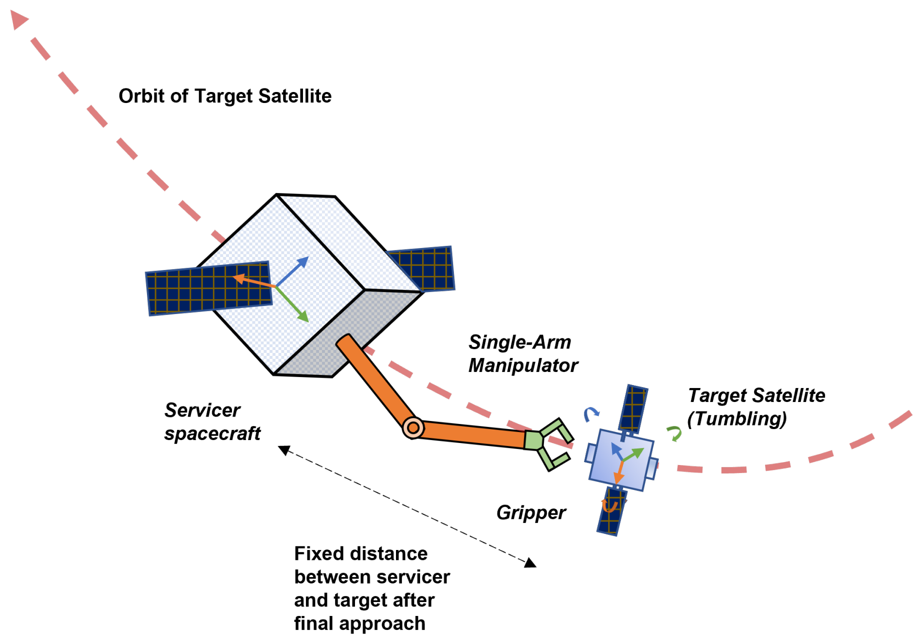


Figure 1. Concept of Motion Planning and Control

less intuitive due to the difficulty in tuning the control gains which must be adaptive to provide a desirable solution for different scenarios. In addition, due to the complex formulation based on the pseudo-inverse of the Jacobian (from the differential kinematics equation), certain computational errors can arise alongside potential singularities. The CLIK control method offers the advantage of successfully achieving multiple tasks such as tracking, physical constraint satisfaction, singularity avoidance, and collision avoidance.[20,21] However, it isn't adept at adaptively prioritizing these tasks. Hence, this study aims to resolve this issue and presents a fuzzy-aided CLIK control method. A fuzzy inference system (FIS) is used to provide the weights to secondary accelerations for optimizing the motion of the servicer spacecraft based on physical constraints such as joint limits and collision avoidance with the target satellite. FISs are used to adaptively define the priorities since they are highly effective in simplifying complex engineering systems to simple *if-then* statements and provide a more explainable methodology.[22] A FIS is composed of membership functions (MFs) and *if-then* rules which form the knowledge base of the system.[23] However, FISs cannot always ensure an optimal solution since they depend on the *expert knowledge* of the system designer. Therefore, optimization techniques such as the genetic algorithm (GA) can be employed to optimize the MFs and rules of the FIS.[22] GA is an evolutionary optimization algorithm based on *natural selection*.[24] It is a highly effective method to optimize FISs as it easily avoids local optima and provides a solution very close to the global optima.[24] This study considers certain uncertainties like modeling errors and external disturbances to validate the robustness of the proposed methodology. The proposed method is validated using simulation studies that consider the dynamics of the manipulator, the servicer spacecraft, and the target satellite.

PRELIMINARIES

Problem Considerations

This problem aims at controlling the servicer spacecraft and the attached robotic arm to successfully aim the end-effector at a capture point located on a tumbling target satellite. The capture process is assumed to be simplistic in nature such that tracking the capture point with the end-effector of the robotic arm is considered to be a successful mission. Since the capture phase of the detumbling mission is generally preceded by the rendezvous phase, the relative translational velocity between the servicer spacecraft and the target satellite is considered to be zero, i.e., the distance between the two bodies is constant. Furthermore, since the control is applied to the servicer spacecraft, the inertial frame is assumed to be located at the center of the servicer spacecraft.

For simplifying the modeling of system dynamics, the servicer spacecraft and the target satellite are assumed to be rigid bodies, and the robotic arm links are assumed to be rigid, thin rods. For the controller implementation, it is assumed that the inertial properties and dynamic models of both the servicer spacecraft and the target satellite are known, i.e., the servicer spacecraft knows the position and velocity of the capture point on the target satellite. Furthermore, it is assumed that the SMS is fully controllable, and the subsequent motion is fully continuous.

Dynamic Model of the Space Manipulator System (SMS)

The SMS consists of a 2 DOF robotic arm mounted on a 3 DOF spacecraft as shown in Figure 2. The robotic arm consists of two links, each with a single DOF. In this study, the considered SMS has i (i = 0, 1, and 2) bodies, where body 0 is the servicer spacecraft, body 1 is the arm link 1, and body 2 is the arm link 2. As shown in Figure 2, the position vector of the spacecraft is represented in the inertial frame as ${}^N\boldsymbol{r}_{S/N} \in \mathcal{R}^3$, or simply $\boldsymbol{r}_S$ where $S : \{\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \hat{\mathbf{s}}_3\}$ denotes the body frame attached to the center of the servicer spacecraft. Similarly, the position vectors of the two links of the robotic arms are represented in the inertial frame as ${}^N\boldsymbol{r}_{B_1/N} \in \mathcal{R}^3$ (or $\boldsymbol{r}_{B_1}$) and ${}^N\boldsymbol{r}_{B_2/N} \in \mathcal{R}^3$

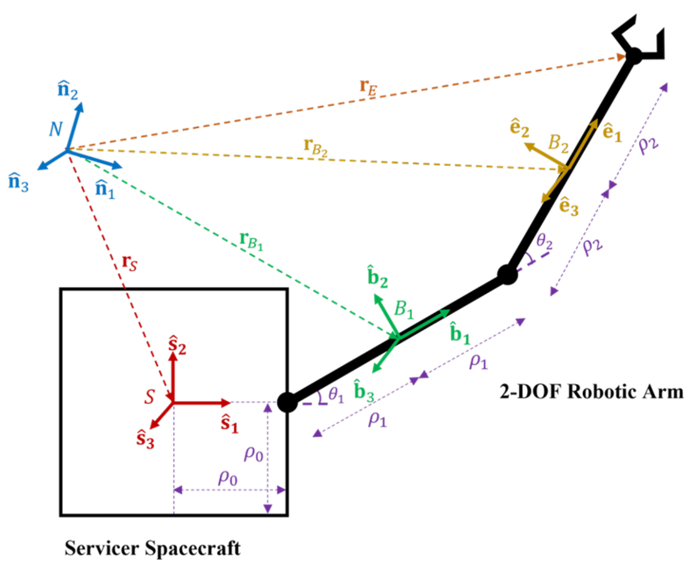


Figure 2. Frame Diagram of the Space Manipulator System

(or $\boldsymbol{r}_{B_2}$), respectively. Here, the manipulator body frames $B_1 : \{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \hat{\mathbf{b}}_3\}$ and $B_2 : \{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3\}$ are attached to the center-points of the two links of the manipulator. For simplicity, the superscript N is omitted hereafter.

The position vectors of each body frame are expressed as follows:

$$\boldsymbol{r}_{B_1} = \boldsymbol{r}_S + C_{NS}\left({}^{S}\begin{bmatrix}\rho_0\\0\\0\end{bmatrix} + C_{SB_1}\,{}^{B_1}\begin{bmatrix}\rho_1\\0\\0\end{bmatrix}\right), \tag{1}$$

$$\boldsymbol{r}_{B_2} = \boldsymbol{r}_{B_1} + C_{NS}C_{SB_1}\left({}^{B_1}\begin{bmatrix}\rho_1\\0\\0\end{bmatrix} + C_{B_1B_2}\,{}^{B_2}\begin{bmatrix}\rho_2\\0\\0\end{bmatrix}\right), \tag{2}$$

where $C_{\alpha\beta}$ represents the direction cosine matrix from β to α. From the position vectors defined in the inertial frame, one obtains the following expressions for the velocities of the 3 bodies as

$$\boldsymbol{v}_S = \frac{\mathrm{d}\boldsymbol{r}_S}{\mathrm{d}t} \in \mathcal{R}^3, \tag{3}$$

$$\boldsymbol{v}_{B_1} = \frac{\mathrm{d}\boldsymbol{r}_{B_1}}{\mathrm{d}t} \in \mathcal{R}^3, \tag{4}$$

$$\boldsymbol{v}_{B_2} = \frac{\mathrm{d}\boldsymbol{r}_{B_2}}{\mathrm{d}t} \in \mathcal{R}^3. \tag{5}$$

Similar to the expressions of the position vectors, the expressions for the angular velocity of each body with respect to the inertial frame are defined as follows:

$$\boldsymbol{\omega}_{B_1} = \boldsymbol{\omega}_S + C_{NS}C_{SB_1}\,{}^{B_1}\begin{bmatrix}0\\0\\\dot{\theta}_1\end{bmatrix} \in \mathcal{R}^3, \tag{6}$$

$$\boldsymbol{\omega}_{B_2} = \boldsymbol{\omega}_{B_1} + C_{NS}C_{SB_1}C_{B_1B_2}\,{}^{B_2}\begin{bmatrix}0\\0\\\dot{\theta}_2\end{bmatrix} \in \mathcal{R}^3, \tag{7}$$

where $\boldsymbol{\omega}_S$ is the angular velocity of the spacecraft with respect to the inertial frame, which is expressed in the N-frame. This is expressed using the first-order kinematic differential equation of a spacecraft[25] in terms of the (3-2-1) Euler Angles as follows:

$$\boldsymbol{\omega}_S = C_{NS}\begin{bmatrix}-\sin\theta_\mathrm{s} & 0 & 1\\ \sin\phi_\mathrm{s}\cos\theta_\mathrm{s} & \cos\phi_\mathrm{s} & 0\\ \cos\phi_\mathrm{s} & -\sin\phi_\mathrm{s} & 0\end{bmatrix}\begin{bmatrix}\dot{\psi}_\mathrm{s}\\\dot{\theta}_\mathrm{s}\\\dot{\phi}_\mathrm{s}\end{bmatrix} \in \mathcal{R}^3, \tag{8}$$

where ϕ_s, θ_s, and ψ_s are the Euler angles that represent the servicer spacecraft attitude along the x, y, and z axes respectively. From the system kinematics defined, the generalized coordinates of the system can be selected as the Euler angles of the spacecraft (ϕ_s, θ_s, ψ_s) and the joint angles of the robotic arm (θ_1 and θ_2) as given below

$$\boldsymbol{q}_\mathrm{s} = \begin{bmatrix}\phi_\mathrm{s} & \theta_\mathrm{s} & \psi_\mathrm{s} & \theta_1 & \theta_2\end{bmatrix}^\mathrm{T}. \tag{9}$$

Using the generalized coordinates, a generalized dynamic model of the system can be written as follows:[21]

$$H_{\mathrm{s}}\left(\boldsymbol{q}_{\mathrm{s}}\right)\ddot{\boldsymbol{q}}_{\mathrm{s}}+C_{\mathrm{s}}\left(\boldsymbol{q}_{\mathrm{s}},\dot{\boldsymbol{q}}_{\mathrm{s}}\right)\dot{\boldsymbol{q}}_{\mathrm{s}}=\boldsymbol{\tau}_{\mathrm{s}}+\mathbf{d}, \tag{10}$$

where $H_{\mathrm{s}}\left(\boldsymbol{q}_{\mathrm{s}}\right)\in\mathcal{R}^{5\times5}$ is defined as the inertia matrix of the SMS, $C_{\mathrm{s}}\left(\boldsymbol{q}_{\mathrm{s}},\dot{\boldsymbol{q}}_{\mathrm{s}}\right)\in\mathcal{R}^{5\times5}$ is the matrix defining the Coriolis and Centrifugal forces acting on the SMS, $\boldsymbol{q}_{\mathrm{s}}\in\mathcal{R}^{5}$ is the vector of generalized coordinates representing the SMS, $\boldsymbol{\tau}_{\mathrm{s}}\in\mathcal{R}^{5}$ is the vector of torques applicable on the corresponding generalized coordinates, and $\mathbf{d}\in\mathcal{R}^{5}$ is the disturbance vector. Note that the subscript ‘s’ represents the SMS dynamics.

To properly define the generalized system dynamics, one follows the Lagrangian method. The Lagrangian, L, is defined as

$$L=E-U, \tag{11}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial E}{\partial\dot{\boldsymbol{q}}}-\frac{\partial E}{\boldsymbol{q}}=\tau, \tag{12}$$

where τ represents the torque acting on the generalized coordinates. Let the kinetic energy of the servicer spacecraft be expressed as follows:

$$E=\frac{1}{2}\sum_{i=0}^{2}\left(\boldsymbol{v}_i^{\mathrm{T}}m_i\boldsymbol{v}_i+\boldsymbol{\omega}_i^{\mathrm{T}}I_i\boldsymbol{\omega}_i\right), \tag{13}$$

where $\boldsymbol{v}_i$ and $\boldsymbol{\omega}_i$ represent the translational and rotational velocities of the i^{th} body, and m_i and I_i denote the mass and moment of inertia of the i^{th} body. The expressions for the translational and rotational velocities can be obtained from the system kinematics. To obtain H_{s}, one can represent the kinetic energy of the servicer spacecraft in a generalized form as[21]

$$E=\frac{1}{2}\dot{\boldsymbol{q}}_{\mathrm{s}}^{T}H_{\mathrm{s}}\dot{\boldsymbol{q}}_{\mathrm{s}}. \tag{14}$$

To obtain the above representation, the linear and angular velocities in Eq. (13) must be represented in terms of the generalized coordinates as

$$\boldsymbol{v}_i={}_{L}J_i\dot{\boldsymbol{q}}_{\mathrm{s}}, \tag{15}$$

$$\boldsymbol{\omega}_i={}_{A}J_i\dot{\boldsymbol{q}}_{\mathrm{s}}, \tag{16}$$

where ${}_{L}J_i\in\mathcal{R}_{3\times5}$ and ${}_{A}J_i\in\mathcal{R}_{3\times5}$ are the linear and angular Jacobians for representing the i^{th} body’s linear and angular velocity in terms of $\boldsymbol{q}$. Note that the left-bottom subscript L and A represent the expressions with respect to the linear and angular velocity. Based on the above relationship (Eqs. (15) and (16)) and the generalized expression of kinetic energy in Eq. (14), the inertia matrix can be expressed as follows:

$$H_{\mathrm{s}}\left(\boldsymbol{q}_{\mathrm{s}}\right)=\sum_{i=0}^{2}\left(m_i\,{}_{L}J_i^{\mathrm{T}}\,{}_{L}J_i+{}_{A}J_i^{\mathrm{T}}I_i\,{}_{A}J_i\right). \tag{17}$$

The Coriolis matrix, C_{s}, can be obtained from the remaining terms in Eq. (12) and can be simply represented using the Christoffel symbol (Z_{ijk}) as follows:[26]

$$C_{\mathrm{s}}\left(\boldsymbol{q}_{\mathrm{s}},\dot{\boldsymbol{q}}_{\mathrm{s}}\right)\{i,j\}=\sum_{k=1}^{5}Z_{ijk}\dot{\boldsymbol{q}}_k, \tag{18}$$

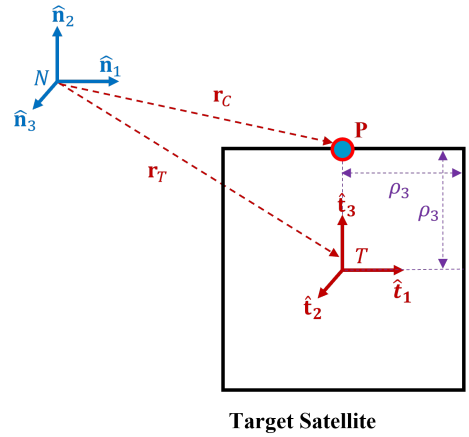


Figure 3. Target Satellite Frame Diagram

where $i, j \in \{1, 2, 3, 4, 5\}$. Here, the Christoffel symbol is expressed in terms of the partial differential of $H_{\mathrm{s}}(\boldsymbol{q}_{\mathrm{s}})$ as follows:[26]

$$Z_{ijk} = \frac{1}{2}\left(\frac{\partial H_{ij}}{\partial q_k} + \frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i}\right). \tag{19}$$

Using the expressions in Eqs. (17) and (18), the complete dynamic model of the servicer spacecraft is formulated as given in Eq. (10).

Dynamic Model of the Target (Tumbling) Satellite

The target satellite is modeled as a single rotating rigid body using Euler's rotational equations of motion[25]

$$I_{\mathrm{t}}\,{}^{T}\dot{\boldsymbol{\omega}}_T + \left({}^{T}\boldsymbol{\omega}_T \times I_{\mathrm{t}}\,{}^{T}\boldsymbol{\omega}_T\right) = \boldsymbol{\tau}_{\mathrm{t}}, \tag{20}$$

where $\boldsymbol{\omega}_T \in \mathcal{R}_3$ is the angular velocity of the target satellite in the target satellite frame ($T : \{\hat{\mathbf{t}}_1, \hat{\mathbf{t}}_2, \hat{\mathbf{t}}_3\}$), $I_{\mathrm{t}} \in \mathcal{R}_{3\times 3}$ is the inertia matrix of the satellite, and $\boldsymbol{\tau}_{\mathrm{t}} \in \mathcal{R}_3$ is the torque acting on the satellite.

The target satellite has a capture point (P) for the servicer spacecraft to aim at for capturing the target as shown in Figure 3. This is usually a circular hatch called the docking ring.[27] In this study, the docking ring is positioned on one face of the target satellite and the corresponding position vector of the capture point in inertial frame (${}^{N}\boldsymbol{r}_{C/N}$ or $\boldsymbol{r}_C$) can be represented as

$$\boldsymbol{r}_C = \boldsymbol{r}_T + C_{NT}\,{}^{T}\begin{bmatrix}0\\0\\\rho_3\end{bmatrix}, \tag{21}$$

where $\boldsymbol{r}_T$ denotes the displacement of the target satellite from the origin of the inertial frame. From the position vector and the angular velocity of the target satellite (Eqs. (20) and (21)), the linear velocity of the capture point can be obtained as

$${}^{N}\boldsymbol{v}_{C/N} = C_{NT}\left({}^{T}\begin{bmatrix}0\\0\\\rho_3\end{bmatrix} \times {}^{T}\boldsymbol{\omega}_T\right). \tag{22}$$

Closed-Loop Inverse Kinematics (CLIK)

The CLIK algorithm is a motion control technique applicable to redundant robotic systems. The CLIK control is highly preferred over other control techniques since it is a very simple approach and offers the advantage of utilizing the redundancy of the given platform to perform multiple tasks at once. Unlike conventional inverse kinematic algorithms, the CLIK does not require any previous knowledge about the robot other than its Jacobian matrix, which also makes it an attractive choice over other conventional algorithms.[28] Note that the kinematic redundancy means that a given system has more DOF than the DOF required for the desired task. The SMS (5 DOF) considered in this study has two more DOFs than required to reach the desired position $[x_t, y_t, z_t]^{\mathrm{T}}$ on the target satellite (3 DOF). Hence, the considered SMS is kinematically redundant. Redundancy is an important aspect of robotic systems since it improves the versatility and flexibility of the robot offering increased operational flexibility. It also helps overcome joint limit constraints, singularities, and collisions.[29,30] This essentially means that the position of the end-effector can be maintained by changing the internal configuration of the system. However, solving the inverse kinematics of redundant systems is tedious since they possess an infinite number of solutions. The inverse kinematics problem can be analyzed by geometric, numerical, or intelligent methods. Of the three, numerical methods such as the Jacobi pseudo-inverse, the gradient projection method, and the weighted norm are widely used.[31] This study uses the Jacobi pseudo-inverse method to derive the inverse kinematics. A Jacobian matrix of the system maps the joint velocities to the desired task velocities. This is used to represent the differential kinematics equation that defines the relationship between the joint and the task space velocities as

$$\dot{\boldsymbol{x}}_e = J_e(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{23}$$

where $J_e(\boldsymbol{q}) \in \mathcal{R}^{3\times5}$ is the Jacobian matrix, and $\dot{\boldsymbol{x}}_{\boldsymbol{e}}$ is the velocity of the end-effector. The desired joint acceleration is then obtained by differentiating Eq. (23). However, an inverse of the Jacobian doesn't exist since J_e is not a square matrix. Therefore, the Moore-Penrose inverse[32] (or pseudo-inverse) is introduced to represent the joint accelerations as follows:

$$\ddot{\boldsymbol{q}} = J_e^{+}(\ddot{\boldsymbol{x}}_{\boldsymbol{e}} - \dot{J}_e\dot{\boldsymbol{q}}), \tag{24}$$

where the pseudo inverse is defined as

$$J_e^{+} = J_e^{\mathrm{T}}(J_e J_e^{\mathrm{T}})^{-1}. \tag{25}$$

The control law to obtain the desired trajectory can be written from the above equation as follows:

$$\ddot{\boldsymbol{q}} = J_e^{+}(\ddot{\boldsymbol{x}}_{\boldsymbol{d}} + K_D\dot{\boldsymbol{e}} + K_P\boldsymbol{e} - \dot{J}_e\dot{\boldsymbol{q}}), \tag{26}$$

where $\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}_e$ is the tracking error, $\boldsymbol{x}_d$ is the desired trajectory, and K_P and K_D are the proportional and derivative gains, respectively. The derived control law, however, is not efficient in avoiding singularities. It also does not exploit the advantages like collision avoidance and join-limit avoidance of a redundant manipulator. Therefore, a CLIK control law is written as follows:

$$\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}_P + \ddot{\boldsymbol{q}}_S, \tag{27}$$

where $\ddot{\boldsymbol{q}}_P$ is the solution for the primary task from Eq. (26) and $\ddot{\boldsymbol{q}}_S$ is the additional joint acceleration vector for the secondary tasks such as satisfying physical constraints and avoiding obstacles.

To perform the aforementioned secondary tasks, a joint velocity vector ($\boldsymbol{\lambda}$) in the null space of J_e is introduced. The null-space velocity error ($\dot{\boldsymbol{e}}_N$) is then written as

$$\dot{\boldsymbol{e}}_N = (I - J_e^+ J_e)(\boldsymbol{\lambda} - \dot{\boldsymbol{q}}), \tag{28}$$

where $(I - J_e^+ J_e)$ projects the vector $(\boldsymbol{\lambda} - \dot{\boldsymbol{q}})$ onto the null space of J_e. This essentially means that the internal configuration of the system can be modified using the null space joint velocity vector without affecting the end-effector's position.

To ensure that the tracking error $\boldsymbol{e}_N$ converges to zero and the joint velocity converges to $\boldsymbol{\lambda}$ in the null space of J_e, the additional joint acceleration vector is written as

$$\ddot{\boldsymbol{q}}_S = (I - J_e^+ J_e)(\dot{\boldsymbol{\lambda}} + K_N \dot{\boldsymbol{e}}_N) - (J_e^+ \dot{J}_e J_e^+ + \dot{J}_e{}^+) J_e (\boldsymbol{\lambda} - \dot{\boldsymbol{q}}), \tag{29}$$

where K_N is a positive definite gain matrix that helps converge the tracking error $\boldsymbol{e}$ to zero in the null space. Therefore, the secondary task accelerations can be obtained by defining the null space joint velocity vector as the gradient of a given objective function

$$\boldsymbol{\lambda} = -\frac{\partial \gamma}{\partial \boldsymbol{q}}, \tag{30}$$

where γ is a scalar objective function. The objective function γ can be defined based on the secondary objectives defined for the system such as physical constraints and obstacle avoidance. Further explanation of the choice of γ is provided in the controller design section later.

Fuzzy Inference System (FIS)

A FIS is a non-linear mapping that yields an output for a given input using fuzzy logic reasoning. This is an extension of the conventional 'if-then' rules system, where outputs are derived by combining the 'if-then' rules with logical operators such as AND/OR.[33] The linguistic nature of fuzzy rules increases the interpretation capability, and the lack of complex mathematical models makes its development and implementation easy.[34] A block diagram of a general FIS is shown in Figure 4. A fuzzification unit converts the crisp numerical system inputs to fuzzy inputs (linguistic variables) using Input MFs. An inference engine makes use of 'if-then' rules and evaluates these variables to give a fuzzy output. A defuzzification unit then converts the fuzzy outputs back to crisp numerical outputs using Output MFs. FISs offer flexibility in terms of implementation as the MFs, and rules can be specified in various ways. The MFs can be either triangular, trapezoidal, or Gaussian

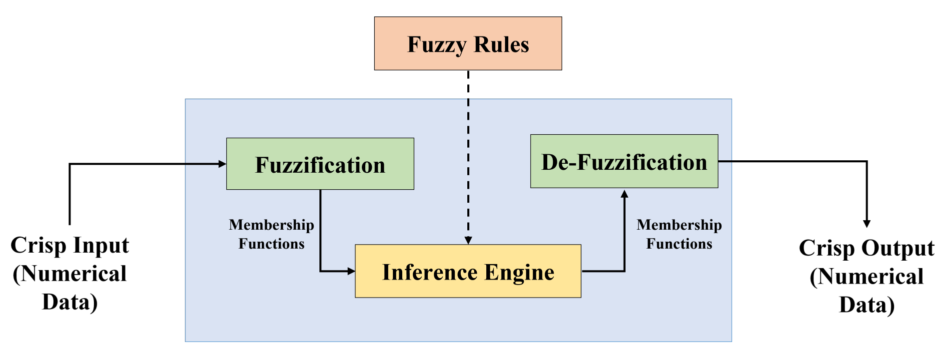


Figure 4. General Block diagram of FIS

whereas the rules can be modified based on the choice of logical operators. For this reason, FISs are used alongside conventional control techniques for adaptively updating certain parameters that need to be selected carefully.[22] Usually, such parameters may be obtained from a set of trial-and-error experiments. However, utilizing a FIS to obtain such parameters makes the control technique more adaptive to changes in the simulation/experimental conditions.[22]

Genetic Algorithm (GA)

GA is a population-based, adaptive meta-heuristic evolutionary algorithm that imitates Darwin's theory of natural selection to find a near-optimal solution in complex search spaces.[33] For a given random population of chromosomes, GA computes the fitness value of each chromosome using a user-defined fitness function. Chromosomes with the highest fitness values are selected using the *selection* process and undergo *crossover* or recombination to produce a new population. These offspring are then mutated to maintain diversity in the population. Furthermore, *elitism* ensures that the best solutions are retained during each iteration. This process of iterative *selection*, *crossover*, and *mutation* produces a population of chromosomes with improved quality of genes (a vector of variables that are to be optimized).[35] Since the process is dynamically iterative, GA can produce multiple near-optimal solutions. Furthermore, it has better global search capability compared to other optimization techniques as it offers the advantage of avoiding local minima.[36] For this reason, GA is an often-used approach to optimize the MFs and rules in a given FIS.[22,37] This method is usually advantageous since the optimization process is offline and the optimized FISs can be easily operated in real time.

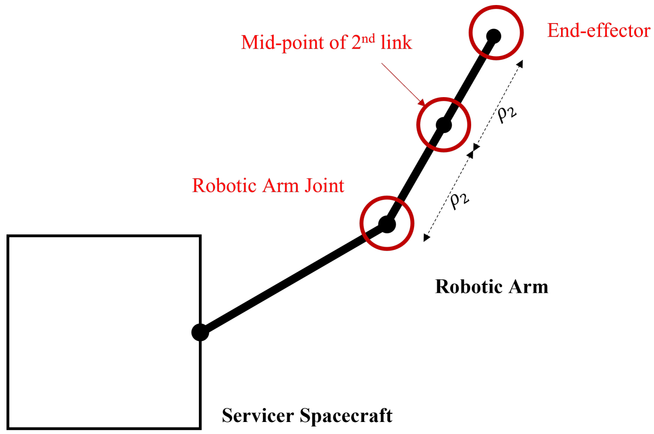
PROPOSED METHODOLOGY

This study proposes a fuzzy-aided CLIK control technique that adaptively controls the 5 DOF SMS to aim the end-effector at the capture point of the target satellite. The conventional CLIK control is dependent on the objective function that is defined using certain physical constraints such as joint limits and potential collisions. However, the change in the joint accelerations may not be necessary at all times. For instance, the robotic arm doesn't need to be constrained if it is not approaching the joint limits or if the distance from the obstacles is higher than a given safe distance. Traditionally, the change in joint accelerations ($\dot{\boldsymbol{\lambda}}$) obtained from these constraints is weighted to give a higher priority to a certain task over another. However, based on the changing environment and conditions, these weights need to be updated. Using an optimization technique often results in an increased computational time which is not desirable for space systems. Therefore, this study proposes the use of FIS to obtain the weighting factor for the joint acceleration change generated by the collision avoidance constraint.

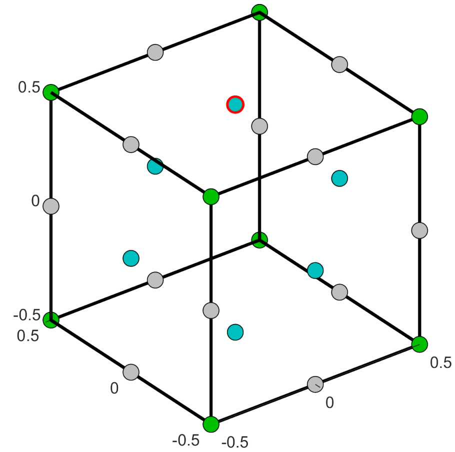
Controller Design

The objective function for secondary tasks in CLIK control is aimed at ensuring that the joints stay within the specified physical limits and that the robotic arm avoids collision with the body of the target satellite. Based on this, the scalar objective function, γ, can be defined as follows:

$$\gamma = \gamma_1 + \gamma_2. \tag{31}$$



(a) Critical Points on the SMS



(b) Obstacle Points on the Target Satellite

Figure 5. Critical and Obstacle Point Definitions

Here, γ_1 focuses on ensuring that the joints remain within the physical limits and can be defined as[29,38]

$$\gamma_1 = \frac{1}{5}\sum_{i=1}^{5} p_i \left(\frac{q_i - \bar{q}_i}{\max(q_i) - \min(q_i)}\right)^2, \tag{32}$$

where $\bar{q}_i$ is the center value of the full range of the coordinate q_i. Here, p_i is a positive weight that can be understood in terms of the freedom of movement allocated to each DOF within the specified physical limits ($\min(q_i)$ and $\max(q_i)$).

To ensure that the robotic arm of the SMS avoids collision with the target satellite while aiming to reach the capture point, the objective function γ_2 is represented as[19]

$$\gamma_2 = \sum_{j=1}^{3}\sum_{k=1}^{2} w_{jk}\frac{1}{d_{jk}^2}, \tag{33}$$

where d_{jk} denotes the distance between the j^{th} critical point (CP) on the robotic arm and the k^{th} nearest obstacle points (OPs) on the target satellite. Here, w_{jk} is the weight corresponding to each CP-OP pair. As given in the equation, this study considers three CPs, and each CP is considered to be affected by only the two nearest OPs. This is to ensure that the controller is not affected by OPs distant from the CPs.

For this study, three CPs on the SMS are defined as shown in Figure 5(a). Since only the second link of the robotic arm on the SMS interacts with the target satellite, this choice of CPs ensures that the robotic arm can successfully avoid any collisions with the target satellite's body. Since the target satellite is modeled as a rigid cube, it contains several points where collisions can occur. These OPs are chosen to cover the majority of the target satellite's surfaces, edges, and vertices depicted in Figure 5(b), where the blue points denote the face centers, grey points denote the edge centers, and the green points denote the vertices. Note that one of the face centers is marked by a red outline, highlighting the capture point, and therefore is not considered an obstacle for the end-effector. Furthermore, to reduce the computational burden during numerical simulations, only the two nearest obstacles are considered for each CP.

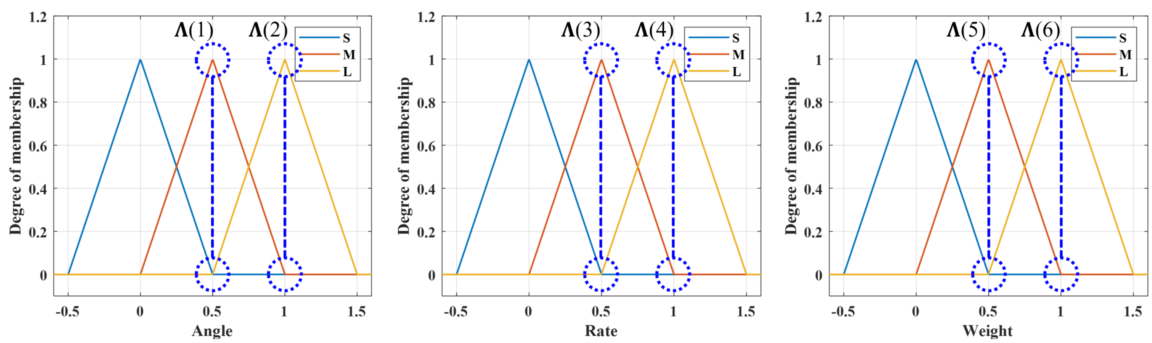
Modeling the FISs

As mentioned in earlier sections, the weights chosen in the CLIK controller design are not adaptive in nature and therefore are replaced by appropriately designed FISs in this study. In its limited context, this study focuses on adapting the weights provided to the collision avoidance constraints for each CP. These include the weights p_i in Eq. (32) and w_{jk} in Eq. (33). This requires the design of two FISs, one aimed at obtaining the joint limit constraint weights p_i (FIS_1) and the other to obtain the collision avoidance constraint weights w_{jk} (FIS_2). Each FIS is modeled as a 2-inputs, 1-output system with three triangular MFs for each input and output. Correspondingly, there are nine rules defined for each FIS.

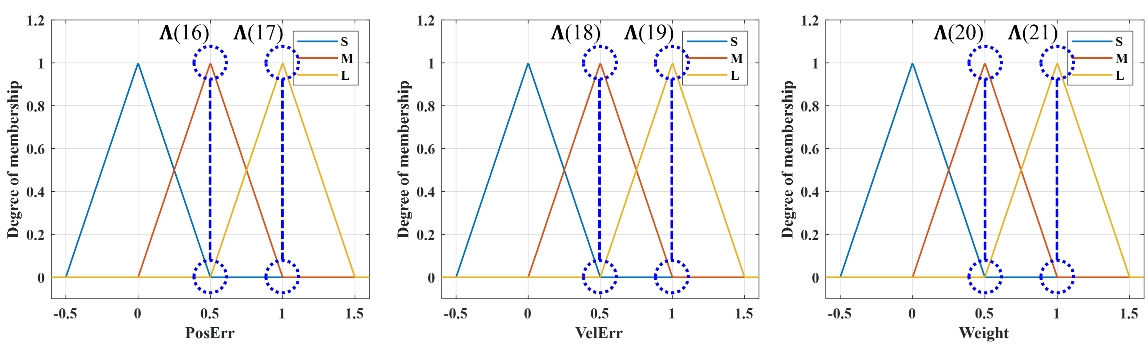
FIS_1 is designed with the joint states (q_i) and joint velocities ($\dot{q}_i$) as inputs and p_i as the output as shown in Figure 6(a). Since the relationships between each joint's states and the corresponding weights are common, the same FIS can be used for all five weights. The corresponding input and output MFs are modeled using two unique parameters each as shown in Figure 7(a). The range of

(a) FIS_1 (b) FIS_2

Figure 6. Structure of the FISs



(a) FIS_1 (Left: q_i, Center: $\dot{q}_i$, Right: p_i)



(b) FIS_2 (Left: d_{jk}, Center: $\dot{d}_{jk}$, Right: w_{jk})

Figure 7. MFs of the FISs

Table 1. FIS_1 Rules

		Angle		
		S	**M**	**L**
Angular Rate	**S**	$\mathbf{\Lambda}(7)$	$\mathbf{\Lambda}(10)$	$\mathbf{\Lambda}(13)$
	M	$\mathbf{\Lambda}(8)$	$\mathbf{\Lambda}(11)$	$\mathbf{\Lambda}(14)$
	L	$\mathbf{\Lambda}(9)$	$\mathbf{\Lambda}(12)$	$\mathbf{\Lambda}(15)$

Table 2. FIS_2 Rules

		Relative position		
		S	**M**	**L**
Relative velocity	**S**	$\mathbf{\Lambda}(22)$	$\mathbf{\Lambda}(25)$	$\mathbf{\Lambda}(28)$
	M	$\mathbf{\Lambda}(23)$	$\mathbf{\Lambda}(26)$	$\mathbf{\Lambda}(29)$
	L	$\mathbf{\Lambda}(24)$	$\mathbf{\Lambda}(27)$	$\mathbf{\Lambda}(30)$

the input and output fuzzy variables is defined as (-0.5, 1.5) by normalizing all variables within this range. The corresponding rule-base for the FIS is specified in Table 1. Here, **S**, **M**, and **L** refer to **Small**, **Medium**, and **Large** MFs. An example rule based on this can be written as: "*If* Angle is **S** *AND* Angular Rate is **M**, *then* Output is $\mathbf{\Lambda}(\mathbf{8})$." As shown, the MF parameters and rules are defined as the elements of vector $\mathbf{\Lambda}$ which is optimized by the GA. Similar to FIS_1, the FIS_2 is defined using the relative distance (d_{jk}) and velocity ($\dot{d}_{jk}$) between a given CP and OP as inputs with the weight w_{jk} as the obtained output as shown in Fig 6(b). The corresponding MFs are shown in Figure 7(b) and the rule-base is shown in Table 2. The parameters that define the MFs and rules of FIS_2 complete the optimization vector $\mathbf{\Lambda}$ for the GA.

Optimization

The parameters that need to be optimized by the GA are referenced as the elements of $\mathbf{\Lambda}$. This vector is referred to as the *gene* which is optimized by creating a population of potential solutions. The best solutions are passed on to the next generation via a series of operations, such as *crossover*, *mutation*, and *elitism*. However, the selection process depends on the fitness function which is used to define the fitness of a given solution. This study considers the following fitness function:

$$\mathcal{J} = \int_{t=0}^{t_F} \boldsymbol{\tau}_{\mathrm{s}}^{\mathrm{T}} \boldsymbol{\tau}_{\mathrm{s}} \, \mathrm{d}t + \Delta, \tag{34}$$

where t_F is the total simulation time, $\boldsymbol{\tau}_{\mathrm{s}}$ is the control effort applied at a given time step to make the end-effector reach the capture point, and Δ is a penalty given to the solutions in case a joint limit is not satisfied or a collision happens. This cost function ensures that the secondary motion generated by the CLIK controller doesn't increase the control effort by huge values thereby improving the efficiency of the CLIK controller. The penalty is defined by a large value and ensures that solutions which cause the physical constraints to be violated are not selected for future generations. Therefore, the choice of penalty is to ensure that $\Delta \gg \int \boldsymbol{\tau}_{\mathrm{s}}^{\mathrm{T}} \boldsymbol{\tau}_{\mathrm{s}} \, \mathrm{d}t$.

Table 3. Physical Parameters

Body No. (i)	Body	ρ_i (m)	m_i (kg)	I_i (kg-m^2)
0	Servicer Spacecraft	0.5	250	25
1	Manipulator Link 1	0.5	25	2.5
2	Manipulator Link 2	0.5	25	2.5
3	Target Satellite	0.5	180	18

SIMULATION STUDY

A simulation study is conducted to validate the proposed approach against the traditional CLIK control and highlight the advantages of using FISs in combination with CLIK control for improving the controller. For this purpose, the SMS and the target satellite are modeled based on similar research work[39], and the corresponding mass and inertial parameters are defined in Table 3.

As shown in Figure 8, the simulation environment is defined with the SMS located at the origin in the inertial frame, i.e., $\boldsymbol{r}_S = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^{\mathrm{T}}$ m and the target satellite located at a distance of 2.5 m from the SMS, i.e., $\boldsymbol{r}_T = \begin{bmatrix}2.5 & 0 & 0\end{bmatrix}^{\mathrm{T}}$ m. Note that the physical parameters of the SMS also include the physical constraints on the robotic arm. These constraints are given as follows:

$$-\frac{\pi}{2} < \theta_1 < \frac{\pi}{2}, \quad -\pi < \theta_2 < \pi. \tag{35}$$

The total simulation time is chosen to be 5 s to allow for enough time for the SMS to accurately track the capture point for at least one second. This allows for enough time to perform the capture operation and initiate the detumbling process. However, this study only focuses on the first part

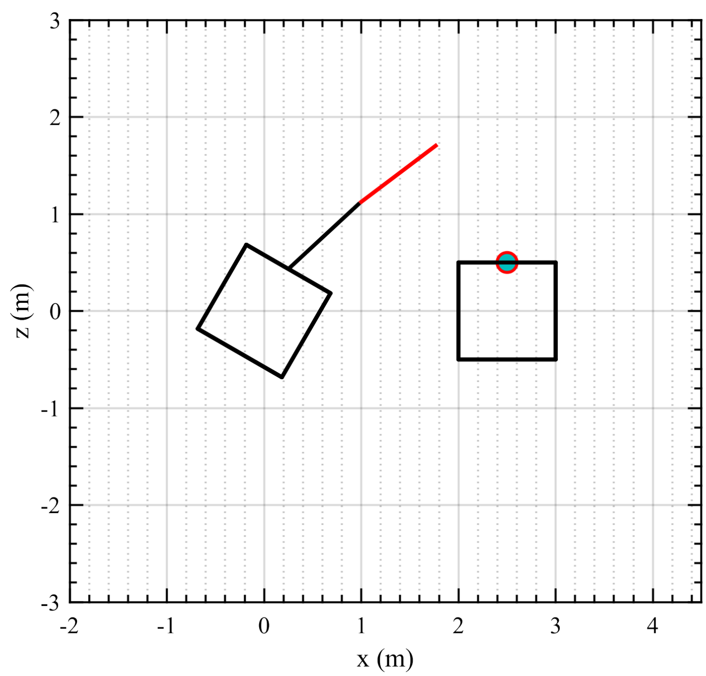


Figure 8. Simulation Environment

Table 4. Training Scenario Parameters

Description	Value
Probability of Crossover	0.80
Mutation Rate	0.10
Elitism Ratio	0.10
Population Size	20
Number of Generations	100
Selection Algorithm	Tournament selection
Tournament size	4
Crossover Algorithm	Scattered crossover
States of the SMS ($\boldsymbol{q}_s(0)$)	$\begin{bmatrix} \frac{\pi}{2} & -\frac{\pi}{3} & 0 & -0.3 & -0.1 \end{bmatrix}^{\mathrm{T}}$ (rad)
State velocities of the SMS ($\dot{\boldsymbol{q}}_s(0)$)	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$ (rad/s)
Euler angles of the target satellite ($\boldsymbol{\theta}_T(0)$)	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$ (rad)
Angular Velocity of the target satellite ($\boldsymbol{\omega}_T(0)$)	$\begin{bmatrix} 0 & -0.2 & 0.2 \end{bmatrix}^{\mathrm{T}}$ (rad/s)

Table 5. Testing Scenario Parameters

Description	Value
Euler angles of the target satellite ($\boldsymbol{\theta}_T(0)$)	$\begin{bmatrix} 0 & -0.2 & 0 \end{bmatrix}^{\mathrm{T}}$ (rad)
Angular Velocity of the target satellite ($\boldsymbol{\omega}_T(0)$)	$\begin{bmatrix} 0.1 & -0.3 & 0.25 \end{bmatrix}^{\mathrm{T}}$ (rad/s)
External disturbance on the SMS ($\mathbf{d}$)	$0.002 \times \begin{bmatrix} \sin(0.025t) \\ \cos(0.02t) \\ \cos(0.03t + \pi/3) \\ 0 \\ 0 \end{bmatrix}$ (N-m)

where the SMS end-effector tracks the capture point. Furthermore, the integration time-step is selected as 0.1 s. To optimize the FISs as defined in the previous section, an appropriate training scenario for the GA needs to be defined. This training scenario aims to present a simple case of a tumbling satellite for the SMS to attempt to capture. As mentioned, the SMS is considered to be in an open position with the robotic arm extended outward towards the target satellite. This marks the start of the capture operation. The training scenario considers certain initial states for the SMS and the target satellite to optimize the FISs. These parameters are listed in Table 4 alongside certain parameters required to initiate the GA. The GA applies a penalty of $\Delta = 10^9$ for avoiding bad solutions. This value was chosen after a few simulations to test the scale of the cost function.

For testing the scalability of the proposed approach, the initial conditions are modified to bring a change in the scenario conditions. A simple testing scenario, similar to the training scenario, is defined to test the proposed approach with an increased tumbling velocity of the target satellite starting from a different initial orientation. Furthermore, certain external disturbances are applied to the servicer spacecraft.[40] These conditions are summarized in Table 5.

Results and Discussion

Figure 9 shows the motion of the SMS to reach the capture point on the target satellite. It is

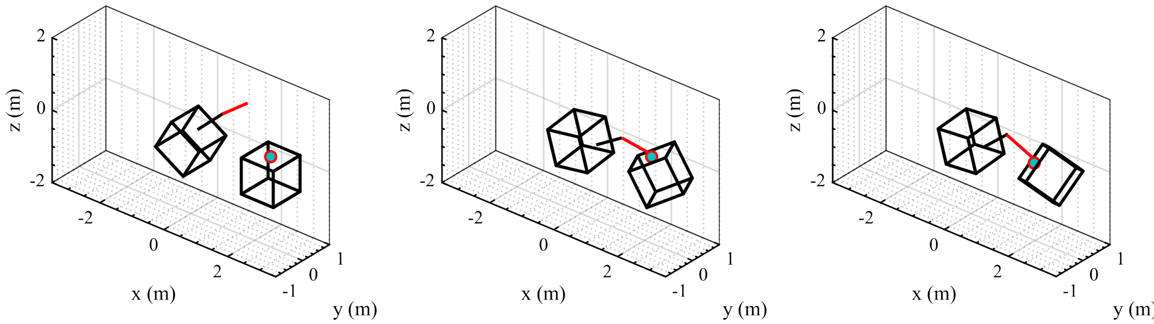


Figure 9. SMS Motion in Training Scenario using FIS+CLIK Approach

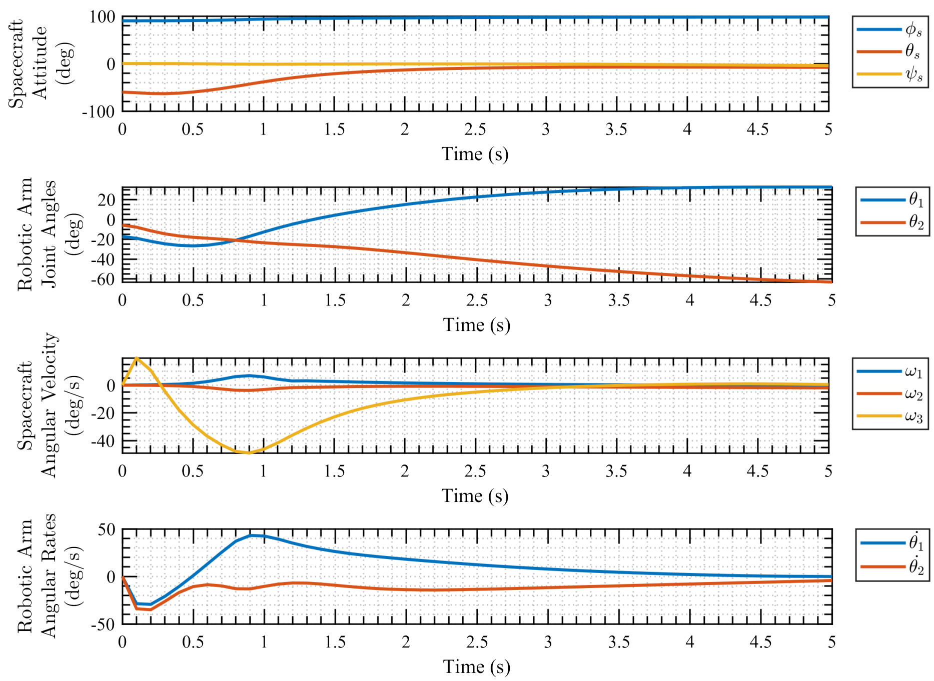
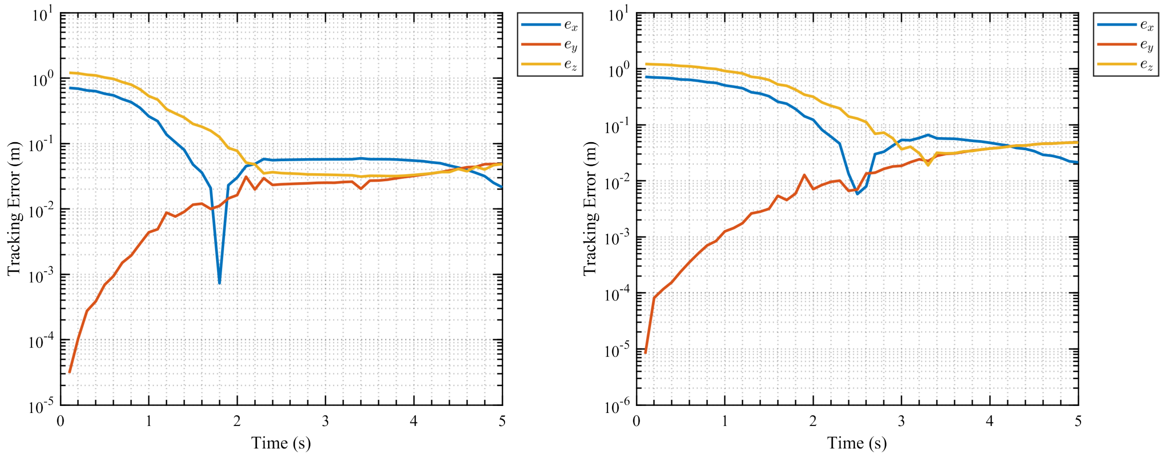


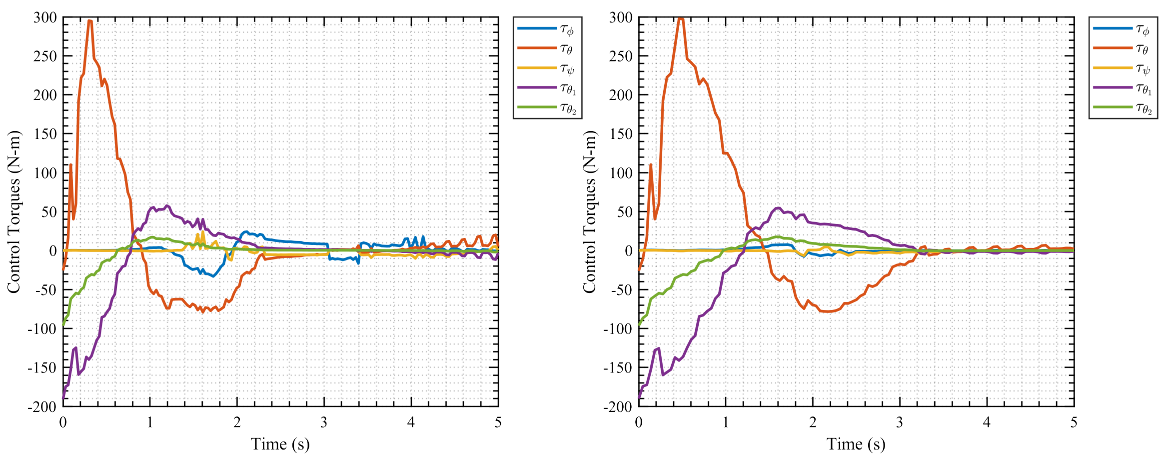
Figure 10. SMS State History in Training Scenario using FIS+CLIK Approach

observed that the SMS is able to reach the capture point while satisfying the physical constraints applicable to the robotic arm. Moreover, it can be seen that the motion is collision-free as all the CPs on the SMS stay away from the OPs on the target satellite. Figure 10 shows the state history of the SMS during the operation, highlighting the low angular rates for the servicer spacecraft. This is desirable as it depicts a realistic motion of the SMS. Figures 11(a) and 11(b) depict the position errors during tracking of the capture point by the SMS end-effector for both cases: with FIS and without FIS. It is shown that the FIS+CLIK approach is able to maintain the same level of accuracy as the conventional CLIK method. Also, to compare the control effort for both the control methods, the cost function defined in Eq. (34) is computed. For the conventional CLIK control technique, the control effort is calculated as $\mathcal{J}_{\text{CLIK}} = 29106.5190$ whereas for the proposed FIS+CLIK method, the value is obtained as $\mathcal{J}_{\text{FIS+CLIK}} = 27317.0636$. From this, it is evident that the proposed method is able to achieve the same level of performance as CLIK control by having the same level of accuracy while being able to satisfy the secondary tasks with much less control



(a) CLIK (b) FIS+CLIK

Figure 11. Tracking Error for the Conventional and Proposed Controller in Training Scenario



(a) CLIK (b) FIS+CLIK

Figure 12. Control Torque for the Conventional and Proposed Controller in Training Scenario

effort. This highlights the advantage of the proposed approach over the traditional CLIK control as by just adaptively updating the weights, the proposed controller requires much less control effort for the given simulation scenario. Figures 12(a) and 12(b) depict the control torque of both control techniques, further highlighting the smoothness in the applied control by implementing the proposed approach over traditional control. The proposed method is validated in another scenario based on conditions listed in Table 5. In this scenario, as shown in Figure 13, the proposed controller is able to successfully control the SMS for performing the capture operation similar to the training scenario. The corresponding tracking error is shown in Figure 14(a) and the applied control effort is shown in Figure 14(b). More fluctuations compared to the training scenario can be observed due to the change in environment and the additional disturbance torques. However, the proposed FIS+CLIK approach again outperforms the CLIK control technique in terms of the total control effort as $\mathcal{J}_{\text{FIS+CLIK}} = 23599.8533$ and $\mathcal{J}_{\text{CLIK}} = 23912.2731$.

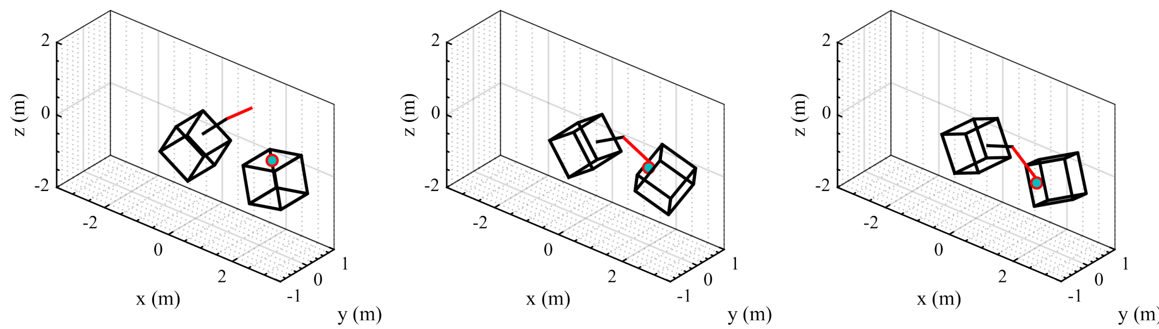
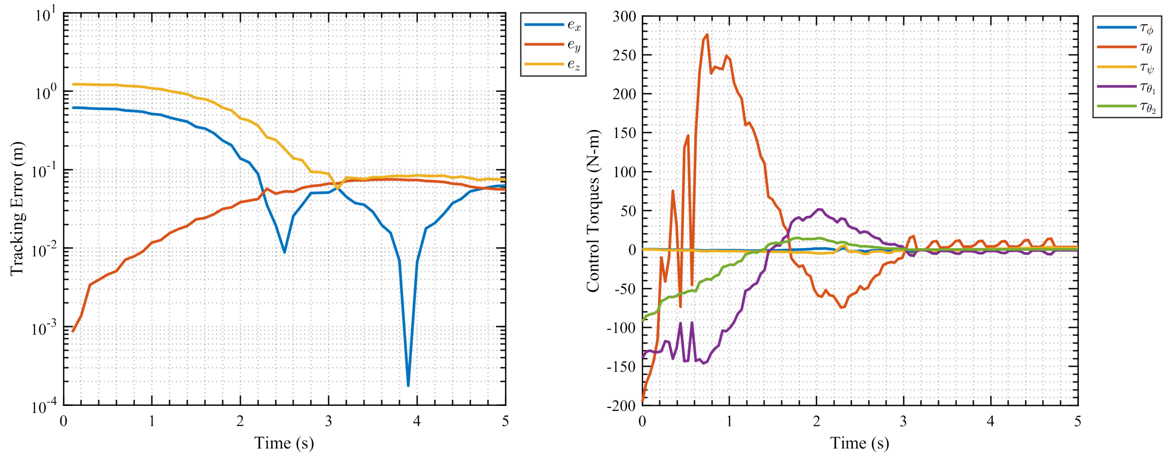


Figure 13. SMS Motion in Testing Scenario using FIS+CLIK Approach



(a) Tracking Error

(b) Control Torque

Figure 14. Tracking Error and Control Torque in Testing Scenario using FIS+CLIK Approach

From the above results, it can be observed that FISs can successfully add value to conventional control techniques by improving the efficiency of the approach. It must be noted that the CLIK control isn't highly accurate or optimal in any sense. However, the modification of CLIK by employing FIS-based weighting of secondary accelerations can allow the satisfaction of physical constraints in a near-optimal manner. This highlights the opportunity to employ FIS+CLIK methodology in the future by further optimizing the CLIK component to achieve higher accuracy in tracking a moving target while aiming to minimize the overall control effort.

CONCLUSION

This study proposes a fuzzy-aided closed-loop inverse kinematics (CLIK) control technique for the safe capture of a tumbling target satellite by ensuring the end-effector's alignment with the capture point on the target satellite. The proposed methodology uses fuzzy inference systems (FISs) to adaptively update the weights for the secondary task accelerations. These secondary tasks include the satisfaction of physical constraints and obstacle avoidance. In this study, a genetic algorithm (GA) is employed to optimize the membership functions and rules of the FISs. Since the GA-based FIS optimization is an offline process, the proposed methodology is quick and therefore computationally efficient. The proposed methodology is validated against CLIK control in a simulation study. The FISs are trained in a simple environment with certain initial conditions and then tested with different initial conditions and added disturbances. The results show that the proposed methodology ensures a collision-free trajectory of the end-effector to the capture point with less control effort than the conventional CLIK control technique. The integration of FISs with CLIK enables increased efficiency in achieving the secondary tasks without largely affecting the required control effort. The proposed method, however, isn't highly accurate, and therefore future work will be carried out to further improve the performance of the proposed method.

ACKNOWLEDGMENT

This research effort was funded through a fellowship award from the University of Cincinnati's Space Research Institute for Discovery and Exploration, Office of Research.

REFERENCES

[1] E. Turan, S. Speretta, and E. Gill, "Autonomous navigation for deep space small satellites: Scientific and technological advances," *Acta Astronautica*, Vol. 193, 2022, pp. 56–74, https://doi.org/10.1016/j.actaastro.2021.12.030.

[2] M. W. Rhodes, J. J. Bennie, A. Spalding, R. H. f. Constant, and I. M. D. Maclean, "Recent advances in the remote sensing of insects," *Biological Reviews*, Vol. 97, No. 1, 2022, pp. 343–360, https://doi.org/10.1111/brv.12802.

[3] K. V. Khlopenkov, K. M. Bedka, J. W. Cooney, and K. Itterly, "Recent Advances in Detection of Overshooting Cloud Tops From Longwave Infrared Satellite Imagery," *Journal of Geophysical Research: Atmospheres*, Vol. 126, No. 14, 2021, p. e2020JD034359. e2020JD034359 2020JD034359, https://doi.org/10.1029/2020JD034359.

[4] N. Saeed, A. Elzanaty, H. Almorad, H. Dahrouj, T. Y. Al-Naffouri, and M.-S. Alouini, "CubeSat Communications: Recent Advances and Future Challenges," *IEEE Communications Surveys Tutorials*, Vol. 22, No. 3, 2020, pp. 1839–1862, 10.1109/COMST.2020.2990499.

[5] V. Demyanov and J. Becedas, *Satellites Missions and Technologies for Geosciences*. IntechOpen, 2020.

[6] Z. Xue, J. Liu, C. Wu, and Y. Tong, "Review of in-space assembly technologies," *Chinese Journal of Aeronautics*, Vol. 34, No. 11, 2021, pp. 21–47, https://doi.org/10.1016/j.cja.2020.09.043.

[7] H. D. Mohr, "Real-Time On-Orbit Momentum Wheel Health Monitoring for Robust Satellite Attitude Control," *2021 IEEE Aerospace Conference (50100)*, 2021, pp. 1–7, 10.1109/AERO50100.2021.9438164.

[8] Goddard Space Flight Center, "On-Orbit Satellite Servicing Study," tech. rep., NASA, 2010.

[9] K. Yoshida, D. Dimitrov, and H. Nakanishi, "On the Capture of Tumbling Satellite by a Space Robot," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 4127–4132, 10.1109/IROS.2006.281900.

[10] N. Inaba and M. Oda, "Autonomous satellite capture by a space robot: world first on-orbit experiment on a Japanese robot satellite ETS-VII," *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, Vol. 2, 2000, pp. 1169–1174 vol.2, 10.1109/ROBOT.2000.844757.

[11] E. Papadopoulos and S. Moosavian, "Dynamics and control of multi-arm space robots during chase and capture operations," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, Vol. 3, 1994, pp. 1554–1561, 10.1109/IROS.1994.407648.

[12] S. Nakasuka and T. Fujiwara, "New method of capturing tumbling object in space and its control aspects," *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No.99CH36328)*, Vol. 2, 1999, pp. 973–978 vol. 2, 10.1109/CCA.1999.800940.

[13] W. Xu, Y. Liu, and Y. Xu, "The coordinated motion planning of a dual-arm space robot for target capturing," *Robotica*, Vol. 30, 09 2012, pp. 755–771.

[14] F. Aghili, "Optimal control for robotic capturing and passivation of a tumbling satellite with unknown dynamics," *AIAA guidance, navigation and control conference and exhibit*, 2008, p. 7274.

[15] C. Jiao, L. Zhang, X. Su, F. Xue, and B. Liang, "Predictive motion control for autonomous capture of a tumbling target with a space manipulator," *Journal of the Franklin Institute*, Vol. 359, No. 15, 2022, pp. 7913–7935.

[16] X. Cyril, G. J. Jaar, and A. K. Misra, "Dynamical modelling and control of a spacecraft-mounted manipulator capturing a spinning satellite," *Acta Astronautica*, Vol. 35, No. 2, 1995, pp. 167–174. Astrodynamics, https://doi.org/10.1016/0094-5765(94)00262-K.

[17] P.-c. Cong and Z.-w. Sun, "Pre-impact configuration planning for capture object of space manipulator," *2008 2nd International Symposium on Systems and Control in Aerospace and Astronautics*, 2008, pp. 1–6, 10.1109/ISSCAA.2008.4776179.

[18] K. Hovell and S. Ulrich, "Laboratory Experimentation of Spacecraft Robotic Capture Using Deep-Reinforcement-Learning–Based Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 11, 2022, pp. 2138–2146.

[19] P. Hsu, J. Mauser, and S. Sastry, "Dynamic control of redundant manipulators," *Journal of Robotic Systems*, Vol. 6, No. 2, 1989, pp. 133–148, https://doi.org/10.1002/rob.4620060203.

[20] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of Intelligent and Robotic Systems*, Vol. 3, Sep 1990, pp. 201–212, 10.1007/BF00126069.

[21] A. Chhabra and D. Kim, "Ground Robotic Platform for Simulation of On-Orbit Servicing Missions," *Journal of Aerospace Information Systems*, Vol. 19, No. 7, 2022, pp. 480–493, 10.2514/1.I011008.

[22] D. Choi, A. Chhabra, and D. Kim, "Intelligent cooperative collision avoidance via fuzzy potential fields," *Robotica*, Vol. 40, No. 6, 2022, p. 1919–1938, 10.1017/S0263574721001454.

[23] E. H. Mamdani, "Applications of fuzzy algorithms for control of a simple dynamic plant," *Proceedings of the IEEE*, 1974.

[24] D. E. Goldberg, "Genetic Algorithms in Search Optimization and Machine Learning," Addison-Wesley Longman Publishing Co., Inc., 1988.

[25] H. Schaub and J. Junkins, *Analytical Mechanics of Space Systems*. AIAA education series, American Institute of Aeronautics and Astronautics, Incorporated, 2014.

[26] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing, Springer London, 2008.

[27] L. Zhang, W. Pan, and X. Ma, "Real-time docking ring detection based on the geometrical shape for an on-orbit spacecraft," *Sensors*, Vol. 19, No. 23, 2019, p. 5243.

[28] A. Colomé and C. Torras, "Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements," *IEEE/ASME Transactions on Mechatronics*, Vol. 20, No. 2, 2014, pp. 944–955.

[29] J. Wang, Y. Li, and X. Zhao, "Inverse kinematics and control of a 7-DOF redundant manipulator based on the closed-loop algorithm," *International Journal of Advanced Robotic Systems*, Vol. 7, No. 4, 2010, p. 37.

[30] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE Journal on Robotics and Automation*, Vol. 4, No. 4, 1988, pp. 403–410.

[31] G. Song, S. Su, Y. Li, X. Zhao, H. Du, J. Han, and Y. Zhao, "A closed-loop framework for the inverse kinematics of the 7 degrees of freedom manipulator," *Robotica*, Vol. 39, No. 4, 2021, pp. 572–581.

[32] R. Penrose, "A generalized inverse for matrices," *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 51, No. 3, 1955, p. 406–413, 10.1017/S0305004100030401.

[33] O. Cord *et al.*, *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*, Vol. 19. World Scientific, 2001.

[34] V. Cherkassky, "Fuzzy inference systems: a critical review," *Computational intelligence: soft computing and fuzzy-neuro integration with applications*, 1998, pp. 177–197.

[35] S. Mirjalili, "Evolutionary algorithms and neural networks," *Studies in computational intelligence*, Vol. 780, Springer, 2019.

[36] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, Vol. 80, No. 5, 2021, pp. 8091–8126.

[37] D. Choi, D. Kim, and K. Lee, "Collision Avoidance of Unmanned Aerial Vehicles In an Urban Environment," *NAECON 2021 - IEEE National Aerospace and Electronics Conference*, 2021, pp. 25–32, 10.1109/NAECON49338.2021.9696377.

[38] C. d. Wit, B. Siciliano, and G. Bastin, *Theory of Robot Control*. Communications and Control Engineering, Springer London, 2012.

[39] A. Flores-Abad, L. Zhang, Z. Wei, and O. Ma, "Optimal capture of a tumbling object in orbit using a space manipulator," *Journal of Intelligent & Robotic Systems*, Vol. 86, No. 2, 2017, pp. 199–211.

[40] D. Choi, A. Chhabra, and D. Kim, *Genetic Algorithm-Aided Fuzzy Controller for Spacecraft Attitude Maneuver with Uncertainties*, 10.2514/6.2021-4174.