frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 04 May 2021
doi: 10.3389/frobt.2021.638849

Autonomous Robots for Space: Trajectory Learning and Adaptation Using Imitation

R. B. Ashith Shyam[1]*, *Zhou Hao*[1], *Umberto Montanaro*[2], *Shilp Dixit*[2], *Arunkumar Rathinam*[1], *Yang Gao*[1], *Gerhard Neumann*[3] and *Saber Fallah*[1,2]

[1]*Department of Electrical and Electronic Engineering, Surrey Space Center, University of Surrey, Guildford, United Kingdom,* [2]*Department of Mechanical Engineering, University of Surrey, Guildford, United Kingdom,* [3]*Karlsruhe Institute of Technology, Karlsruhe, Germany*

This paper adds on to the on-going efforts to provide more autonomy to space robots and introduces the concept of programming by demonstration or imitation learning for trajectory planning of manipulators on free-floating spacecraft. A redundant 7-DoF robotic arm is mounted on small spacecraft dedicated for debris removal, on-orbit servicing and assembly, autonomous and rendezvous docking. The motion of robot (or manipulator) arm induces reaction forces on the spacecraft and hence its attitude changes prompting the Attitude Determination and Control System (ADCS) to take large corrective action. The method introduced here is capable of finding the trajectory that minimizes the attitudinal changes thereby reducing the load on ADCS. One of the critical elements in spacecraft trajectory planning and control is the power consumption. The approach introduced in this work carry out trajectory learning offline by collecting data from demonstrations and encoding it as a probabilistic distribution of trajectories. The learned trajectory distribution can be used for planning in previously unseen situations by conditioning the probabilistic distribution. Hence almost no power is required for computations after deployment. Sampling from a conditioned distribution provides several possible trajectories from the same start to goal state. To determine the trajectory that minimizes attitudinal changes, a cost term is defined and the trajectory which minimizes this cost is considered the optimal one.

Keywords: motion planning, probabilistic movement primitives, robot manipulation, learning from demonstrations, trajectory adaptation

OPEN ACCESS

Edited by:
Evangelos G. Papadopoulos,
National Technical University of Athens, Greece

Reviewed by:
Karol Seweryn,
Space Research Center (PAN), Poland
Shital Chiddarwar,
Visvesvaraya National Institute of Technology, India

****Correspondence:***
R. B. Ashith Shyam
a.rajendrababu@surrey.ac.uk
shyamashi@gmail.com

Specialty section:
This article was submitted to Space Robotics, a section of the journal Frontiers in Robotics and AI

Received: *07 December 2020*
Accepted: *16 April 2021*
Published: *04 May 2021*

Citation:
Ashith Shyam RB, Hao Z, Montanaro U, Dixit S, Rathinam A, Gao Y, Neumann G and Fallah S (2021) Autonomous Robots for Space: Trajectory Learning and Adaptation Using Imitation. Front. Robot. AI 8:638849. doi: 10.3389/frobt.2021.638849

1 INTRODUCTION

Robots that operate in space are very much limited due to the unique challenges encountered like communication latency, lack of power sources and extreme safety requirements. Current robots operating in space are either controlled from ground stations or tele-operated. As a result, there is large scale research going on to make space robots more autonomous.

One of the critical issues that require immediate attention is the ever increasing space junk, especially in the past decade as it poses a huge threat to the functioning spacecraft (satellites, International Space Station etc.). There are more than half a million debris in Low Earth Orbit (LEO) NASA, 2013 and it is estimated that the space environment can be stabilised when on the order of 5–10 objects are removed from LEO per year ESA, 2018. Although several methods for space debris removal has been proposed like harpoons, nets, tentacles SPACE.COM, 2018, using robotic arms to capture still remain the preferred choice as it can be extended to various other application areas like on-orbit servicing and assembly and autonomous rendezvous and docking.

Spacecrafts require flying at a nominal attitude to charge battery, communicate with the ground station and determine its attitude and position. However orbital environment being micro-gravitational poses a difficult challenge since the spacecraft bus to which the robot-arm is attached is floating and any motion of the robot-arm would induce an attitudinal disturbance to the spacecraft. For free-flying spacecraft, the attitude determination and control system (ADCS) continuously compensate for the disturbances from the operation of the manipulator to maintain the nominal attitude of the spacecraft and hence a lot of energy is consumed.

Free-floating is a conceptual operating state of the spacecraft (when ADCS is switched off) installed with robotic arm. This type of spacecraft leave the attitude uncontrolled during the operation of the robotic arm. However, leaving the attitude of the spacecraft tumbled is unsafe and not ideal for the power system and the sensors used for determining its attitude. For both cases, we want the trajectory planner of the robotic arm to consider two important aspects, viz.,

1. minimal attitudinal disturbances of the spacecraft bus due to manipulator operation
2. computationally inexpensive for minimal power consumption.

Inthis work, we demonstrate how programming by demonstrations or imitation learning Shyam et al., 2019; Paraschos et al., 2018 can be used to plan trajectory of a 7-DoF robot arm attached to small spacecrafts. The learned trajectories are efficiently encoded as a probabilistic distribution (PD) from which we can sample out trajectories for reproduction. This method is computationally efficient and is capable of minimizing the attitude disturbances (as shown in **Section 5.3**).

Optimal control methods Rybus et al., 2016; Rybus et al., 2017; Camacho and Alba, 2013 are well developed but it often get stuck at local minima due to poor initial guess and if successful, produces only a single trajectory. However, modeling as a PD captures the mean as well as the variance of the trajectories. The variance information could be used for sampling initial guess values form the PD for optimization based local planners to perturb the trajectory to avoid obstacles. It is known that the quality of initial guess determines the computational load and avoidance of local minima Shyam et al., 2019.

Future space robots is expected to have human arm like dexterity. Hence it is proposed to use a 7-DoF redundant robot arm. This has the added advantage that the planning and control can still be carried out effectively even if a joint encoder or sensor fails.

2 RELATED WORK

The analysis of the kinematic and dynamic of spacecraft with manipulator is well established. Exploiting the non-holonomic behavior of the orbital manipulator system for spacecraft attitude and end-effector trajectory control have been studied extensively. It usually involves joint space techniques to control both the motion of the arm and sometimes the spacecraft attitude Yoshida and Nakanishi, 2003; Hirano et al., 2018. Early research used mapping methods to correlate the end-effector position with the induced disturbances on the spacecraft to minimize the attitude disturbances Torres and Dubowsky, 1992; Vafa and Dubowsky, 1993. However, the mapping methods are computationally inefficient and furthermore, higher DoF manipulators will significantly increase the mapping difficulty and are challenging to find optimised paths.

The work by Nenchev et al. Nenchev et al. (1999) proves that for certain manipulator motions, no reaction forces are induced on the spacecraft. As mentioned in their work, such solutions exists only for some special cases where integrability of the reaction null space velocity exists. This work then inspired many following research to exploit and optimise the control method for spacecraft with manipulator Dimitrov and Yoshida, 2006; Piersigilli et al., 2010; Nguyen-Huynh and Sharf, 2013.

More recently, researchers have attempted to solve the problem of trajectory planning by minimizing a cost functional which satisfies certain criteria. For example Rybus et al. (2016); Seweryn and Banaszkiewicz (2008) minimizes the power consumption. Non-linear Model Predictive Control (NMPC) have been used for control of free-floating spacecrafts Rybus et al., 2017 but it remains to be seen how such heavy computations can be carried out by an on-board spacecraft computer. The other focus is post-panning impedance control of the orbital manipulator to free-motion targets. These researches aim to solve the kinodynamics in order to finely control the impact force for safe and accurate manipulation in the micro-gravity environment Papadopoulos, 1992.

2.1 Contributions

The main contributions of this work are

1. Imitation learning based trajectory planning:

- First the trajectories are learned from demonstrations and encoded as a probabilistic distribution (PD). Planning to an unseen target only requires sampling and conditioning of the PD. This avoids computationally expensive optimization methods (which usually have a cost function to minimize) to run on on-board computer.

2. Minimize attitude disturbances during capture:

- Sampling from a PD for our redundant manipulator arm can produce infinite possible trajectories theoretically. Attitude disturbances for each trajectory can be easily computed and is possible to choose the trajectory with the least disturbance.

This paper is organized as follows. **Section 3** gives briefly kinematic and dynamic formulations of the spacecraft manipulator system. In **Section 4**, we discuss the method used for generating trajectory data for learning. **Section 5** provides the equations by which trajectories can be compactly encoded as a probabilistic distribution which can be used further for reproduction to unseen situations. **Section 6** presents the simulation results and **Section 7** gives the conclusions and future directions.

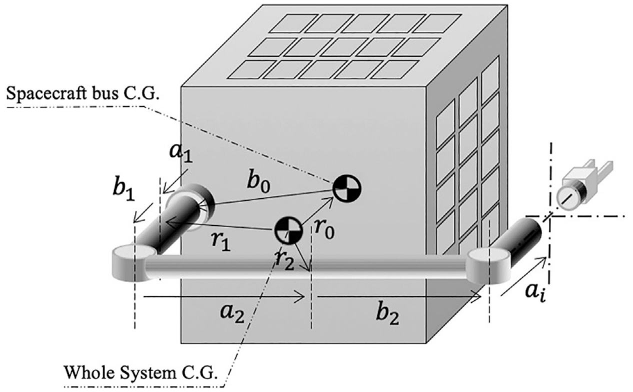

FIGURE 1 | Schematic diagram of a spacecraft-manipulator arm.

TABLE 1 | DH parameters of the robot arm.

Joint	α (rad)	*a* (m)	*d* (m)	θ (rad)
1	$-\frac{\pi}{2}$	0.0	0.5	θ_1
2	$\frac{\pi}{2}$	0.0	0.0	θ_2
3	$\frac{\pi}{2}$	0.9	0.0	θ_3
4	$-\frac{\pi}{2}$	0.9	0.0	θ_4
5	$\frac{\pi}{2}$	0.8	0.0	θ_5
6	$-\frac{\pi}{2}$	0.8	0.0	θ_6
7	$\frac{\pi}{2}$	0.0	0.8	θ_7

TABLE 2 | Simulation Parameters.

	Satellite L 0	L 1	L 2	L 3	L 4	L 5	L 6	L 7
Mass (kg)	200.0	20.0	30.0	30.0	20.0	20.0	20.0	20.0
I_x	1400.0	0.10	0.25	0.25	0.25	0.25	0.25	0.25
I_y	1400.0	0.10	25.0	25.0	25.0	25.0	25.0	25.0
I_z	2040.0	0.10	25.0	25.0	25.0	25.0	25.0	25.0

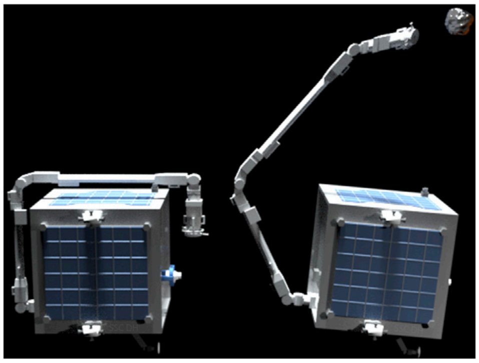

FIGURE 2 | Left: Home position of the Future Space Debris Removal Orbital Manipulator (FSDROM) (a 7-DoF redundant robot arm attached to the spacecraft); **Right:** Position of the Future Space Debris Removal Orbital Manipulator (FSDROM) with Manipulator capturing an orbital debris.

3 DYNAMIC FORMULATION

The kinematic and dynamic formulation of free-floating spacecrafts have been studied previously Umetani and Yoshida, 1989; Wilde et al., 2018; Nanos and Papadopoulos, 2017. Here we give a abridged version of the same for completeness. This formulation makes it easier to compute various matrices especially the coriolis and centrifugal which requires symbolic differentiation of the mass matrix. The whole formulation is carried out using Python's symbolic library called 'sympy' Meurer et al., 2017 which supports 'C' code generation as well for faster execution.

3.1 Nomenclature

- m_i: mass of the i^{th} link, the first being the spacecraft
- r_i: position vector of the centre of mass of i^{th} link with respect to the inertial co-ordinate system
- $\dot{r}_i$: linear velocity of the centre of mass of i^{th} link with respect to the inertial co-ordinate system
- I_i: moment of inertia of i^{th} link with respect to the inertial co-ordinate system
- ω_i: angular velocity of the i^{th} link with respect to the inertial co-ordinate system
- a_i: vector pointing from the joint i to the centre of mass of link i
- b_i: vector pointing from the centre of mass of link i to joint $i + 1$
- l_i: length of i^{th} link
- ϕ_s: vector of attitude angles (yaw, pitch and roll) of the spacecraft
- ϕ_m: vector of manipulator joint angles

3.2 Assumptions

1. Momenta is conserved and is zero at the beginning
2. Gravity is negligible
3. The Centre of Mass of the system coincides with the origin of the inertial co-ordinate system
4. The motion planning is carried out when the satellite-manipulator system at a safe state and is sufficiently close to the target.

The mass centre of the spacecraft-manipulator arm can be described as

$$\sum_{i=0}^{n} m_i r_i = 0 \tag{1}$$

The linear and angular momentum conservation equations become

$$\sum_{i=0}^{n} m_i \dot{r}_i = 0 \tag{2}$$

$$\sum_{i=0}^{n} I_i \omega_i = 0 \tag{3}$$

From **Figure 1**, the geometrical relationship between the various vectors can be written as

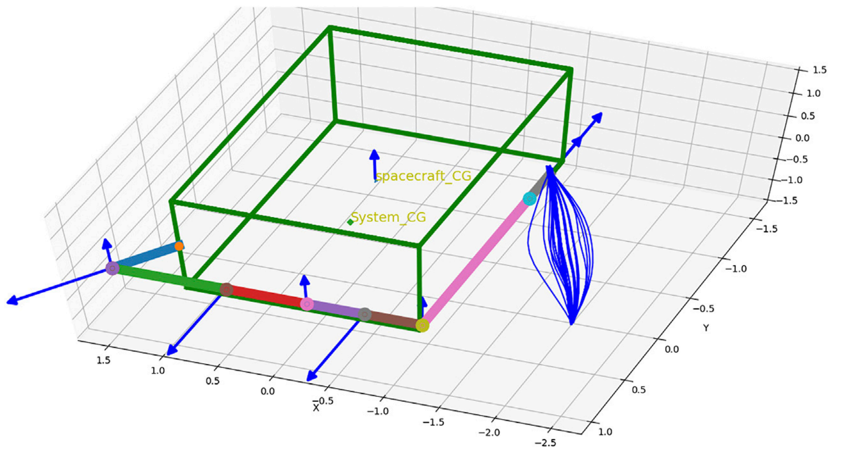

FIGURE 3 | Home position and several possible trajectories of the end-effector to reach the same target.

$$r_i = r_{i-1} + a_i + b_{i-1} \tag{4}$$

Equations. 1, 4 can be solved simultaneously to obtain the centre of mass of the spacecraft and can be expressed as in **Eq. 5**

$$\begin{aligned} r_s = r_0 &= -\sum_{i=0}^{n-1} Kij\left(b_i + a_{i+1}\right) \\ K_{ij} &= 1 - \sum_{j=0}^{i} \frac{m_j}{W} \end{aligned} \tag{5}$$

$$v_s = \frac{d}{dt} r_s \tag{6}$$

where W is the total mass of the system, r_s and v_s are the position vector and linear velocity of the spacecraft with all vectors expressed with respect to the inertial co-ordinate system. The position vector and velocity of the rest of the links can be found using the recursive relation given by **Eq. 4**. The differential kinematics of the satellite-manipulator arm system gives the jacobian matrix of the system which consists of the manipulator part (J_m) and the satellite part (J_s). Thus the end-effector velocity, v_{eef}, and the momentum conservation can be expressed as in Umetani and Yoshida, 1989.

$$v_{eef} = J_{s\dot{\phi}s} + J_{m\dot{\phi}m} \tag{7a}$$

$$0 = I_s \dot{\phi}_s + I_m \dot{\phi}_m \tag{7b}$$

From **Eq. 7**, the end-effector velocity can be solved as a function of the manipulator joint rates and generalized jacobian, J^* given by $(J_m - J_s I_s^{-1} I_m)$

$$\begin{aligned} v_{eef} &= \left(J_m - J_s I_s^{-1} I_m\right) \dot{\phi}_m \\ &= J^* \dot{\phi}_m \end{aligned} \tag{8}$$

where I_s and I_m are respectively the satellite and manipulator inertia matrices expressed in inertial co-ordinate system Umetani and Yoshida, 1989.

The Kinetic energy, T, can then be expressed as

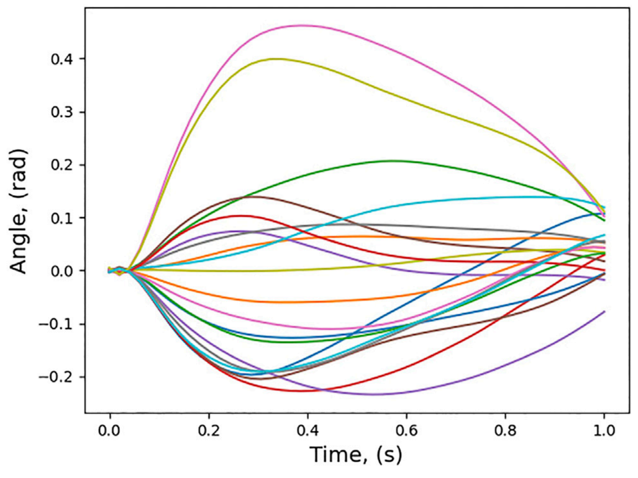

FIGURE 4 | Learned and normalized trajectory distribution for joint one.

$$T = \sum_{i=0}^{n} mi\left(vi \cdot vi\right) = \frac{1}{2} \dot{\phi}^T M(\phi) \dot{\phi} \tag{9}$$

where $M(\phi)$ is the mass matrix and $\phi = [\phi_s^T \phi_m^T]^T$. The centripetal and coriolis vector, C, is given by

$$C\left(\phi, \dot{\phi}\right) = \dot{M}(\phi)\,\dot{\phi} - \frac{1}{2} \begin{bmatrix} \dot{\phi}^T \dfrac{\partial M(\phi)}{\partial \phi_1} \dot{\phi} \\ \dot{\phi}^T \dfrac{\partial M(\phi)}{\partial \phi_2} \dot{\phi} \\ \cdot \\ \cdot \\ \dot{\phi}^T \dfrac{\partial M(\phi)}{\partial \phi_n} \dot{\phi} \end{bmatrix} \tag{10}$$

The equation of motion of the free-floating spacecraft manipulator system can be written as

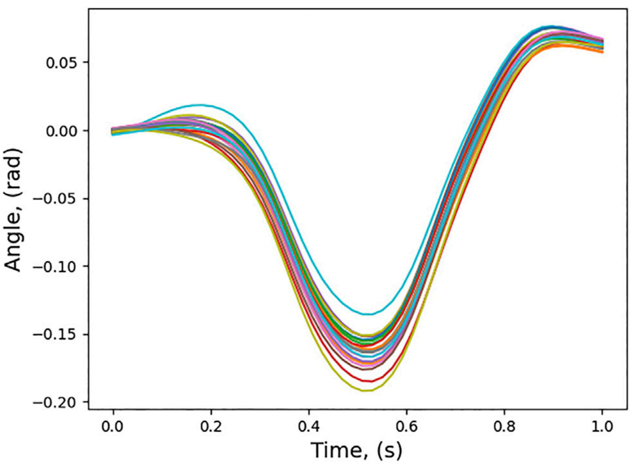

FIGURE 5 | Conditioned trajectory distribution for joint one.

$$M(\phi)\ddot{\phi} + C\left(\phi, \dot{\phi}\right) = \begin{bmatrix} 0 \\ \tau \end{bmatrix} \tag{11}$$

where τ is the control torque to be applied at the manipulator joints.

4 DATA GENERATION FOR TRAJECTORY LEARNING

The method introduced here requires data samples for trajectory learning. A trajectory, ζ, is a mapping of all the robot configuration (x) from start to goal with time. Mathematically it can be represented as $\zeta : [0,1] \rightarrow x$ where $x \in \mathbb{R}^d$ and d corresponds to the number of joints with $\zeta(0)$ and $\zeta(1)$ being the start and goal configurations respectively. These trajectories could be generated by a human expert by demonstrations Zhu and Hu, 2018; Havoutis and Calinon, 2019. As real hardware orbital simulation of micro-gravity environment being extremely expensive, we demonstrate the concept by generating trajectories using an optimal control algorithm Kirk, 2004. We make use of the redundancy of the chosen 7-DoF manipulator arm to generate several trajectories which starts at the home position (**Figure 2**) and go to a particular goal state given by the vision system. Once enough trajectories are generated, the goal state is changed and the process is repeated until the entire workspace is covered.

The cost function for trajectory generation is given as

$$J = x_T^T P_t x_T + \int_{t_0}^{T-1} x_t^T Q_t x_t + u_t^T R_t u_t \, dt \tag{12}$$

subject to the constraints

$$\begin{aligned} \dot{x}_t &= A_t x_t + B_t u_t \\ x_t(0) &= x_0 \end{aligned}$$

where x_t represents the state (position and velocity in task space) of the manipulator joints at time t. For a space-manipulator, once the manipulator states have been found out, the satellite states can be determined from **eq. 7** and integration. Here Q_t and R_t are respectively the time varying state and control cost matrices, P_t is the stabilizing matrix obtained by the solution of algebraic Ricatti equation at every time instant, t_0, T are the initial and final time respectively. The trajectory data samples in simulation are obtained by the following methods.

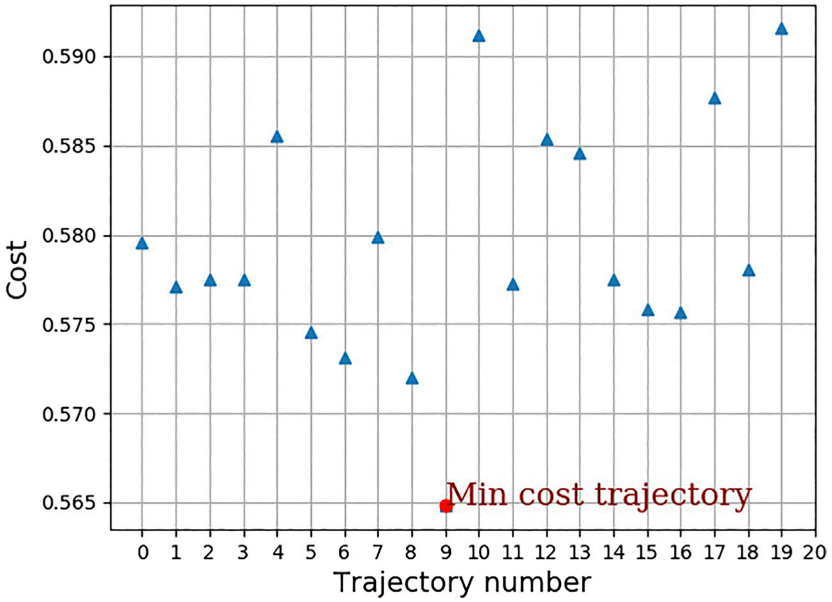

FIGURE 6 | Cost associated with each of the sampled trajectory.

1. varying the cost matrices thus encouraging certain joint motions and discouraging certain other joint motions.
2. introducing artificial obstacles (elastic bands Quinlan and Khatib (1993)) between the start and goal point so as to force the redundant robot to follow a different trajectory to the same goal point
3. introduction of noise into the system given by **Eq. 12**.

5 TRAJECTORY ENCODING AND REPRODUCTION

The generated trajectory data samples need to be represented in an efficient manner for future planning and control. It has to be mentioned that the trajectory planning is carried out when the spacecraft-robot arm is sufficiently close to the target and is safe to operate. Here we demonstrate the core idea of this work by representing the generated trajectories as a probabilistic distribution. We find that Gaussian distributions fit all the essential criteria for efficiently representing trajectories as it depends only on two parameters i.e. mean and covariance. For reproduction of trajectories to unseen situations, we use the conditioning property of the Gaussian as explained in **Section 5.2**.

5.1 Gaussian Trajectory Encoding

The encoding of the trajectories can be expressed as a linear basis function model as in **Eq. 13** where$\psi_t = [\psi_t \; \dot{\psi}_t]^T$ is the basis function (see APPENDIX A), w a parameter vector, plus some error ε. Such a representation reduces the number of parameters and facilitates learning. Assuming trajectories to be independent and identically distributed, the probability of observing a trajectory, ζ, given the parameter vector w can be written as in **Eq. 14** Paraschos et al., 2018.

$$x_t = \begin{bmatrix} q_t \\ \dot{q}_t \end{bmatrix} = \begin{bmatrix} \psi_t \\ \dot{\psi}_t \end{bmatrix} w + \varepsilon_x \tag{13}$$

$$p(\zeta|w) = \prod_t \mathcal{N}\left(x_t \middle| \psi_t w, \Sigma_x\right) \tag{14}$$

where $\epsilon_x \sim \mathcal{N}(0, \Sigma_x)$ represents the zero-mean Gaussian noise associated with each observation and x_t is the state. There are several possible choices for the basis function. Here a radial basis function (or squared exponential) is used for representing stroke based movements which is ideal for motion planning Paraschos et al. (2018). The parameter vector w is modeled as another Gaussian distribution with parameter $\theta = \{\mu_w, \Sigma_w\}$ to capture the variance of the trajectories. (Here μ_w and Σ_w are respectively the mean and covariance of the Gaussian). Using the linear transformation property of the Gaussian distribution (see APPENDIX B), the state can be represented asFor the generated trajectory samples, the parameter vector w can be estimated as a ridge regression and is given in **Eq. 16**

$$\begin{aligned} p(x_t; \theta) &= \int \mathcal{N}\left(x_t \middle| \psi_t w, \Sigma_x\right) \mathcal{N}\left(w \middle| \mu_w, \Sigma_w\right) dw \\ &= \mathcal{N}\left(x_t \middle| \mu_w, \psi_t \Sigma_w \psi_t^T + \Sigma_x\right) \end{aligned} \tag{15}$$

$$w_i = \left(\Psi^T \Psi + \lambda I\right)^{-1} \Psi^T X_i \tag{16}$$

where X_i is a 1-D concatenated vector (see APPENDIX C) of all joint values during all time steps from the i^{th} trajectory sample and Ψ is a block diagonal matrix with each block diagonal being $\boldsymbol{\psi_t}$. The mean and variance of the parameter vector, w, are estimated as in **Eq. 17**

$$\begin{aligned} \mu_w &= \frac{1}{N} \sum_{i=1}^{N} w_i \\ \Sigma_w &= \frac{1}{N} \sum_{i=1}^{N} \left(w_i - \mu_w\right)\left(w_i - \mu_w\right)^T \end{aligned} \tag{17}$$

where N is the number of demonstrations.

All the above computations could be carried out once the spacecraft manipulator design is complete and the whole trajectory planning problem could then be stated as taking the robot from $\zeta(0)$ (home position) to $\zeta(1)$ which is the pose of the target estimated by the vision system.

5.2 Trajectory Planning to Unseen Situations

The data generation described in **Section 4** needs several trajectory samples to accurately represent the workspace of the spacecraft-manipulator. However, workspace can have infinite possible locations of the target theoretically and it is impossible to do data generation for all possible goal poses. The Gaussian distribution introduced above can solve this problem by using the conditional distribution property. A probabilistic trajectory distribution can be conditioned to follow not only the desired start and goal state but also the via-points Paraschos et al., 2018. For example, if our trajectory has to pass through a desired state $x^{\star}{}_t$ the new mean and variance of the conditioned trajectory will be

$$\begin{aligned} \mu_w^{[new]} &= \mu_w + L\left(x_t^* - \psi_t^T \mu_w\right) \\ \Sigma_w^{[new]} &= \Sigma_w - L \psi_t^T \Sigma_w \end{aligned} \tag{18}$$

where L is

$$L = \Sigma_w \psi_t \left(\Sigma_x^* + \psi_t^T \Sigma_w \psi_t\right)^{-1} \tag{19}$$

and Σ_x^* is the desired accuracy to which the state (x_t^*) is to be reached.

5.3 Cost of a Trajectory

The cost of a trajectory is a scalar which estimates how much the attitude of the spacecraft changes when the robot arm follows a particular trajectory. It is defined as follows.

$$Q = c^2 \, \Sigma \, \dot{\phi}_s^T \, \dot{\phi}_s + \Sigma \, v_s^T v_s \tag{20}$$

where $\dot{\phi}_s$ (from **Eq. 7b**) and v_s (from **Eq. 6**) are respectively the rate of change of Euler angles[1] and linear velocities of the spacecraft calculated at each discrete time step from initial to

[1] It is to be noted that angular velocity and rate of change of Euler angles are not the same. Interested readers may please refer to chapter 3 in Siciliano et al., 2010 for a detailed discussion

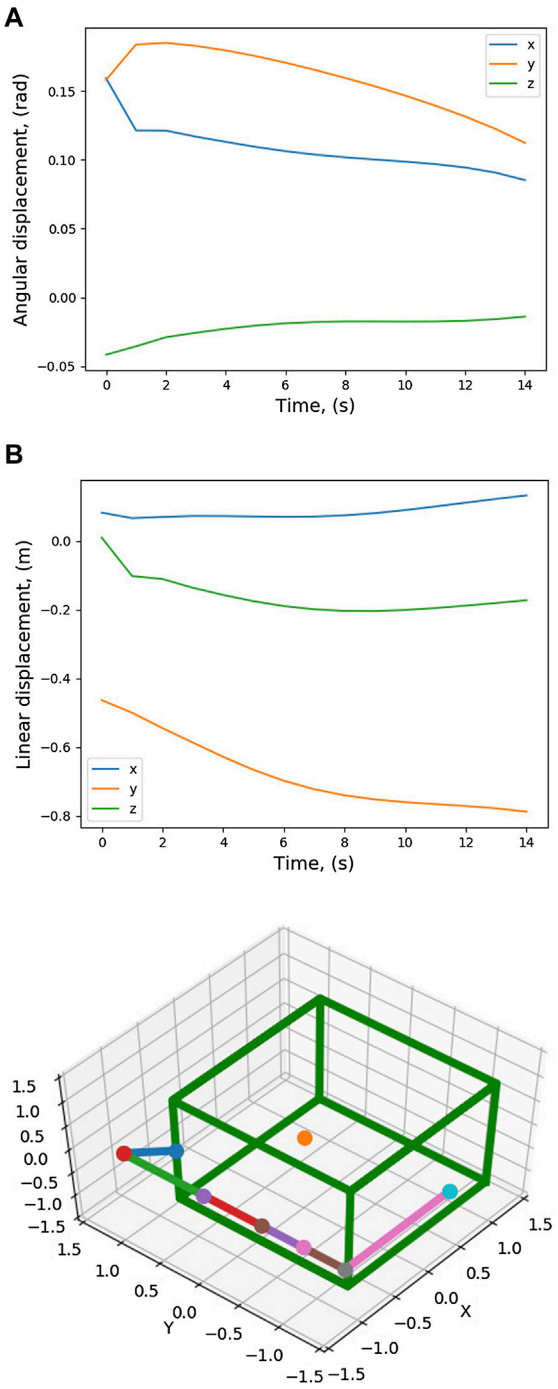

FIGURE 7 | Pose variation of the spacecraft CG during trajectory tracking.

final pose, c is a angular to linear conversion coefficient which allows to combine an angular value with a linear value and Σ is the summation symbol. The minimum cost corresponds to the trajectory having minimal disturbances.

5.4 Algorithm

The algorithm can be summarised as given below.

Algorithm 1: Algorithm for finding optimal trajectory using imitation learning

6 SIMULATION RESULTS

The Denavit-Hartenberg Hartenberg and Denavit (1955) parameters of the robot arm is shown in **Table 1** and the parameters used for the simulation are shown in **Table 2**. For this particular result presented here, the end-effector of the robot is commanded to a target position of [−2, 0, 0] m in Cartesian frame.

Figure 3 shows the home position of the robot arm attached to the spacecraft corresponding to the joint values, $(0.0, 5\frac{\pi}{4}, 0.0, 0.0, \frac{\pi}{2}, -\frac{\pi}{2}, 0.0)$. The spacecraft's Euler angles are (0, 0, 0) at the start and is commanded to any Cartesian position in the world which is considered safe to carry out the manipulation. **Equation 6** can be used to find the distance of centre of mass from inertial coordinate system.

Algorithm 1: Algorithm for finding optimal trajectory using imitation learning

```
Input: Start $\zeta(0)$, DH table of the robot
Output: Trajectory with optimal cost
Data: Trajectory data from the optimal control (joint parameters)
Precompute:
  – Compute the basis function, $\Psi$
  – Compute the mean, $\mu_w$ and covariance, $\Sigma_w$ of the parameter vector from trajectory data.
for all debris do
    – Estimate target pose from vision system, $\zeta(1)$
    – Compute $\mu_w^{new}$ and $\Sigma_w^{new}$
    – Sample trajectories (say 100) from the conditioned distribution using $\mu_w^{new}$ and $\Sigma_w^{new}$
    – For each trajectory, find the cost
    – Select the trajectory having minimum cost
```

The trajectories obtained from the optimal control algorithm are normalized in the time interval 0–1. The learned trajectory distribution for joint 1 is shown in **Figure 4**.

The conditioned trajectory for joint 1 is also shown in **Figure 5**.

Twenty trajectories in joint space are sampled out from the conditioned distribution. For the twenty joint space trajectories, the corresponding end-effector trajectories in task space are found out and are also shown in **Figure 3**. **Figure 6** shows the cost of each of the trajectories and the trajectory which has the minimum cost. For this trajectory, the induced motion on the spacecraft is shown in **Figure 7**.

7 CONCLUSION

To the authors' knowledge, this is the first time imitation learning is used in trajectory planning of robot arms for free floating spacecraft. This work addresses the issue of minimizing attitude disturbance spacecraft bus when the arm reaches out to capture a debris. The learning is carried out offline and is computationally very efficient for finding new trajectories after deployment.

The trajectory learning algorithm presented in this paper will be potentially tested on a Future Space Debris Removal Orbital Manipulator which has a similar micro-satellite spacecraft bus as RemoveDEBRIS Forshaw et al., 2017 but with a 7-DoF redundant robot arm attached, as shown in **Figure 2**. This is the next step towards space autonomy for on-orbit operations that will be demonstrated by a potential mission concept that goes beyond RemoveDEBRIS spacecraft. The overall mission objectives would be to execute pose estimation, trajectory and motion planning of the robotic arm, and capture a sample debris in order.

DATA AVAILABILITY STATEMENT

The work undertaken comes under the purview of University of Surrey's policy. The codes used for generating data can only be made available with prior permission from the university. Please contact Dr. Ashith Shyam Babu at shyamashi@gmail.com for any queries.

AUTHOR CONTRIBUTIONS

AS: Conceived the idea of using imitation learning for space robots as a computationally inexpensive way to carry out trajectory planning. Also carried out the implementation and manuscript writing ZH: Conceived the idea of FSDROM mission and the spacecraft design and contributed in the literature survey on space manipulation control. AR: Have contributed in literature survey and some very useful discussions SD and UM: Have contributed in MPC implementation YG: Supervision GN: Implementation of the code and idea of using programming by demonstration in robotics SF: Supervision and project lead.

FUNDING

This work is supported by grant EP/R026092 (FAIR-SPACE Hub) through UKRI under the Industry Strategic Challenge Fund (ISCF) for Robotics and AI Hubs in Extreme and Hazardous Environments.

REFERENCES

Camacho, E. F., and Alba, C. B. (2013). *Model Predictive Control*. London, United Kingdom: Springer-Verlag.

Dimitrov, D., and Yoshida, K. (2006). "Utilization of Holonomic Distribution Control for Reactionless Path Planning," in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, January 09–January 15 2006 (IEEE), 3387–3392.

ESA (2018). Active space debris removal. Available at: https://www.esa.int/Safety_Security/Space_Debris/Active_debris_removal (Accessed November 5, 2020).

Forshaw, J. L., Aglietti, G. S., Salmon, T., Retat, I., Roe, M., Burgess, C., et al. (2017). Final Payload Test Results for the RemoveDebris Active Debris Removal Mission. *Acta Astronautica* 138, 326–342. doi:10.1016/j.actaastro.2017.06.003

Hartenberg, R. S., and Denavit, J. (1955). A Kinematic Notation for Lower Pair Mechanisms Based on Matrices. *J. Appl. Mech.* 77, 215–221.

Havoutis, I., and Calinon, S. (2019). Learning from Demonstration for Semi-autonomous Teleoperation. *Auton. Robot* 43, 713–726. doi:10.1007/s10514-018-9745-2

Hirano, D., Kato, H., and Saito, T. (2018). "Online Path Planning and Compliance Control of Space Robot for Capturing Tumbling Large Object," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, October 1–October 5 2018 (IEEE), 2909–2916.

Kirk, D. E. (2004). *Optimal Control Theory: An Introduction*. Mineola, New York: Courier Corporation.

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., et al. (2017). Sympy: Symbolic Computing in python. *Peerj Comp. Sci.* 3, e103. doi:10.7717/peerj-cs.103

Nanos, K., and Papadopoulos, E. G. (2017). On the Dynamics and Control of Free-Floating Space Manipulator Systems in the Presence of Angular Momentum. *Front. Robotics AI* 4, 26. doi:10.3389/frobt.2017.00026

NASA (2013). Space debris and human spacecraft. Available at: https://www.nasa.gov/mission_pages/station/news/orbital_debris.html (Accessed November 3, 2020).

Nenchev, D. N., Yoshida, K., Vichitkulsawat, P., and Uchiyama, M. (1999). Reaction Null-Space Control of Flexible Structure Mounted Manipulator Systems. *IEEE Trans. Robot. Automat.* 15, 1011–1023. doi:10.1109/70.817666

Nguyen-Huynh, T. C., and Sharf, I. (2013). Adaptive Reactionless Motion and Parameter Identification in Postcapture of Space Debris. *J. Guidance, Control Dyn.* 36, 404–414. doi:10.2514/1.57856

Papadopoulos, E. G. (1992). "Path Planning for Space Manipulators Exhibiting Nonholonomic Behavior," in IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC, United States, July 7–July 10 1992 (IEEE), 669–675.

Paraschos, A., Daniel, C., Peters, J., and Neumann, G. (2018). Using Probabilistic Movement Primitives in Robotics. *Auton. Robot* 42, 529–551. doi:10.1007/s10514-017-9648-7

Piersigilli, P., Sharf, I., and Misra, A. K. (2010). Reactionless Capture of a Satellite by a Two Degree-Of-Freedom Manipulator. *Acta Astronautica* 66, 183–192. doi:10.1016/j.actaastro.2009.05.015

Quinlan, S., and Khatib, O. (1993). "Elastic Bands: Connecting Path Planning and Control,". in Proceedings IEEE International Conference on Robotics and Automation, Atlanta, GA, May 2-6, 1993 (IEEE), 802–807.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2021.638849/full#supplementary-material

Rybus, T., Seweryn, K., and Sasiadek, J. Z. (2016). "Trajectory optimization of space manipulator with non-zero angular momentum during orbital capture maneuver," in AIAA Guidance, Navigation, and Control Conference, San Diego, CA, January 4-8, 2016. 0885.

Rybus, T., Seweryn, K., and Sasiadek, J. Z. (2017). Control System for Free-Floating Space Manipulator Based on Nonlinear Model Predictive Control (Nmpc). *J. Intell. Robot Syst.* 85, 491–509. doi:10.1007/s10846-016-0396-2

Seweryn, K., and Banaszkiewicz, M. (2008). "Optimization of the Trajectory of a General Free-Flying Manipulator during the Rendezvous Maneuver," in AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii, August 18-21, 2008, 7273.

Shyam, R. A., Lightbody, P., Das, G., Liu, P., Gomez-Gonzalez, S., and Neumann, G. (2019). "Improving Local Trajectory Optimisation Using Probabilistic Movement Primitives," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, November 3–November 8 2019, 2666–2671.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2010). *Robotics: Modelling, Planning and Control*. London, United Kingdom: Springer-Verlag.

SPACE COM (2018). Space Junk Clean Up: 7 Wild Ways to Destroy Orbital Debris. Available at: https://www.space.com/24895-space-junk-wild-clean-up-concepts.html (Accessed November 10, 2020).

Torres, M. A., and Dubowsky, S. (1992). Minimizing Spacecraft Attitude Disturbances in Space Manipulator Systems. *J. Guidance, Control Dyn.* 15, 1010–1017. doi:10.2514/3.20936

Umetani, Y., and Yoshida, K. (1989). Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix. *IEEE Trans. Robot. Automat.* 5, 303–314. doi:10.1109/70.34766

Vafa, Z., and Dubowsky, S. (1993). On the Dynamics of Space Manipulators Using the Virtual Manipulator, with Applications to Path Planning. *Space Robotics: Dyn. Cont.*, 45–76. doi:10.1007/978-1-4615-3588-1–310.1007/978-1-4615-3588-1_3

Wilde, M., Kwok Choon, S., Grompone, A., and Romano, M. (2018). Equations of Motion of Free-Floating Spacecraft-Manipulator Systems: An Engineer's Tutorial. *Front. Robotics AI* 5, 41. doi:10.3389/frobt.2018.00041

Yoshida, K., and Nakanishi, H. (2003). "Impedance Matching in Capturing a Satellite by a Space Robot," in Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, United States, October 27–October 31, 2003 (IEEE), 3059–3064.

Zhu, Z., and Hu, H. (2018). Robot Learning from Demonstration in Robotic Assembly: A Survey. *Robotics* 7, 17. doi:10.3390/robotics7020017

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

APPENDIX

A Basis Function

The basis function used in this work is given by

$$\psi_i(z) = exp^{\left(-\frac{(z-c_i)^2}{h^2}\right)}$$

here z_t is the phase, c_i is the center and h is the bandwidth factor. Interested readers can refer to Paraschos et al. (2018) for details.

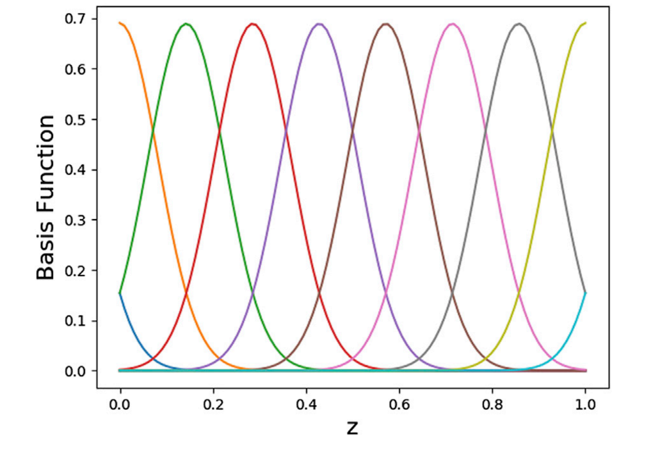

FIGURE A1 | Radial basis function.

Figure A1 gives a plot of ten basis function centered at [−0.142, 0., 0.142, 0.285, 0.428, 0.571, 0.714, 0.857, 1., 1.142].

B Linear Transformation Property of Gaussian Distribution

If a random variable x is normally distributed ($\mathcal{N}(\mu, \Sigma)$), the linear transformation $Ax + c$ follows the distribution

$$Ax + c \sim \mathcal{N}\left(A\mu + c, A\Sigma A^T\right)$$

C Explanation of the Learning Scheme

To illustrate the learning process, a simple example of 2-DoF planar robot is considered here. Let θ_1 and θ_2 represent the joint angles. The complete trajectory for a single demonstration, ζ, concatenated as a 1-D vector is represented as

$$\lambda_1 = \left[\theta_{1_{t1}}, \theta_{1_{t2}}, \ldots, \theta_{1_{tn}}\right]$$
$$\lambda_2 = \left[\theta_{2_{t1}}, \theta_{2_{t2}}, \ldots, \theta_{2_{tn}}\right]$$

$$\zeta = [\lambda_1 \lambda_2, \dot{\lambda}_1, \dot{\lambda}_2]^T_{tn\times 2nDoF}$$

Here t_i are the time points, tn and $nDoF$ are the total time points and number of degrees of freedom respectively. Let the number of basis functions be nBf. The matrices ψ_t and $\dot{\psi}_t$ are of dimension $2nDoF \times (nBf \times 2nDoF)$ each and the matrix Ψ is of dimension $(tn \times 2nDoF) \times (nBf \times 2nDoF)$. The learning parameter, w_i for one demonstration is then calculated by formulating as a regression problem and reducing the loss, $(\zeta - \Psi w)^2$