Image Space based Path Planning for Reactionless Manipulation of Redundant Space Robot

Rachit Bhargava[1], P. Mithun[1], V. V. Anurag[1], A.H. Abdul Hafez[2] and S. V. Shah[3]

***Abstract*— This work addresses path planning for reactionless visual servoing of a redundant dual-arm space robot through exploration in the image space. The planner explores the image moment based feature space, impends acceleration to the image features and extends the feature tree. A reactionless visual servoing control law is integrated to extend the tree in the configuration space simultaneously. The proposed algorithm is able to incorporate the necessary coupling between the motions of the the dual arms and the base of the robot to ensure zero base reactions. Additionally, it also gives the flexibility to apply multiple constraints in both the image space and the configuration space. The effectiveness of the proposed framework is exhibited by implementing the algorithm on a numerical model of a 14-DoF dual arm space robot.**

I. INTRODUCTION

The use of robots in space, especially in future On Orbit Services (OOS)[1], is inevitable due to the efficiency, reliability, and ease of operation of robotic manipulators [2]. Redundant manipulators are especially useful for applications in deep space since they are capable of performing tasks that require high dexterity. But path planning for redundant manipulators involves computations in high dimensional state spaces. Tree based random sampling techniques like Rapidly exploring Random Trees (RRT) [3], [4] are effective in exploring high dimensional spaces and are capable of performing goal based exploration. The main challenge in such methods is that for most manipulators, the goal is defined in the task space. The inverse kinematics that map the task space onto the configuration space is not trivial for a redundant system. For a point in the task space, there is an infinite number of solutions in the configuration space of a redundant robot. This challenge was addressed in [5] where the node in the exploration tree closest to the task space goal was chosen using a task space based distance metric and extended in a random direction in the configuration space. This eliminated the need for inverse kinematics. The work presented in [6] improves the convergence of the tree further by directing the extension towards the goal using an error function defined in the task space.

There is extensive literature dedicated to visual servoing such as [7], [8], that utilises an error function in the image space to control the motion of the robot. In these works, the main motive was to generate a feasible image feature trajectory that guides the manipulator towards the goal in the task space while abiding to certain constraints.

For space manipulators, ensuring zero base reactions during manipulation is preferred [9]. Reactionless visual servoing [9] was proposed, which utilised a task functional approach to minimize base reactions to zero. However, upon occurrence of algorithmic singularity in presence of a task conflict, this framework sacrifices the secondary task. This problem was addressed to some extent in [10], where a switching-based path planning framework was proposed for avoiding algorithmic singularity due to such task conflicts. The planning was done by switching between two phases, a Servoing to Goal phase and a Singularity Avoidance phase, depending on whether or not singularity was encountered. Nevertheless, the planner can possibly take many cycles to complete the task successfully and even get infinitely stuck in a switching loop between the two phases.

This motivated the proposal of a random sampling technique, RRT in this work. Planning through RRT inherently allows taking into account various constraints as obstacles. An RRT based approach for path planning for vision-based control of earth-based robot was demonstrated in [11]. The outcome was discrete feature trajectories. In such approaches where planning and execution are decoupled, the underlying dynamic system may not be able to follow the trajectories even after fitting kinematic paths that are optimized to fit the dynamic constraints. The work presented in [12], [13] for earth-based robots addressed this issue by implementing randomized kinodynamic path planning through exploration in the camera state space. The algorithm presented in the aforementioned works however, cannot be applied directly for reactionless visual servoing, mainly due to the nonholonomic constraints associated with reactionless manipulation. More importantly, the work done in [12], [13] does not take into account the dynamic coupling between the arms and base of the robot.

In this paper, we propose path planning in image space for reactionless visual servoing of a satellite mounted dual-arm redundant robot. The proposed algorithm uses tree based sampling in the image feature space, where tracking the end goal is easy and straightforward. By utilising a sampling based planning algorithm, the framework is able to incorporate several tasks or constraints in the image space and configuration space such as field of view limits, joint angle limits and singularity avoidance. The algorithm uses certain feature based criteria and a local planner in the image space to ensure smooth extension of the tree in the feature space. Additionally, a vision based control law is developed

* This research was supported by INSPIRE research Grant (IFA-13 ENG-52) by Department of Science and Technology, India.

[1] R. Bhargava, P. Mithun, and V. V. Anurag are with RRC, KCIS International Institute of Information Technology, Hyderabad - 500032, India

[2] A. H. Abdul Hafez is with Hasan Kalyoncu University, Sahinbey 27410, Gaziantep, Turkey

[3] S. V. Shah is with Indian Institute of Technology Jodhpur, Jodhpur - 342011, India

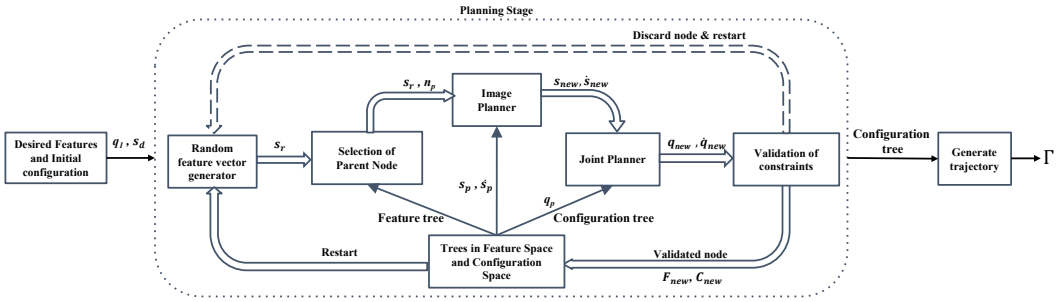


Fig. 1: Block diagram for image-based path planning algorithm for reactionless visual servoing of a space robot

to provide a unique mapping from the feature velocities to the joint velocities that ensure reactionless manipulation. These are the main contributions of this work.

The following section provides a brief outline of the proposed algorithm. Sections III and IV describe the planning in the image space and configuration space respectively. Results obtained from simulation of the proposed algorithm on a numerical model of a 14-DoF dual arm space robot are summarized in section V. This is followed by the conclusions in Section VI. The video accompanying the paper discusses the motivation behind the algorithm and its implementation on the aforementioned space robot model.

II. Outline of the Proposed Planning Algorithm

The algorithm presented in this work has been developed for a dual-arm free-floating robot with eye-in-hand systems. The algorithm is initiated by feeding the desired features (in both image planes) $\boldsymbol{s}_d$ and the initial configuration $\boldsymbol{q}_1$ to the planning stage. These inputs are used to initialize the trees in the feature space and the configuration space. It is important to note here, that configuration collectively refers to the robot base pose and the joint angles in both the arms. As shown in Fig.1, each iteration of the planning stage begins with the generation of a random goal feature vector $\boldsymbol{s}_r$ in the image space. For the generated $\boldsymbol{s}_r$, the parent feature node with index $\boldsymbol{n}_p$ is selected from the feature tree using the Direction criteria and Distance criteria. Given the index, the parent feature state and parent configuration state information can be extracted from the respective trees. The image planner takes in the random goal, the parent features $\boldsymbol{s}_p$ and parent feature velocities $\dot{\boldsymbol{s}}_p$ as the inputs and impends accelerations to the features. The image planner, thus gives the feature velocities of the new node, $\dot{\boldsymbol{s}}_{new}$ as the output. The new feature vector, $\boldsymbol{s}_{new}$ is obtained by updating $\boldsymbol{s}_p$ by integrating $\dot{\boldsymbol{s}}_{new}$ over a small time interval.

The configuration velocity $\dot{\boldsymbol{q}}_{new}$, which contains the base velocity $\boldsymbol{t}_{b_{new}}$ and joint velocity $\dot{\boldsymbol{\theta}}_{new}$ (for both arms) is generated from $\dot{\boldsymbol{s}}_{new}$ using the joint planner. By numerical integration of $\dot{\boldsymbol{q}}_{new}$ over a stipulated time, the change in the configuration is attained. This change is used to update the parent configuration, $\boldsymbol{q}_p$ to attain the configuration of the new node $\boldsymbol{q}_{new}$, which contains the new pose of the base $\boldsymbol{b}_{new}$ and the joint angles of both arms $\boldsymbol{\theta}_{new}$. The newly generated feature state and configuration state are validated for constraints in order to ensure an unobstructed extension of both the trees. Upon validation, the newly generated feature state is encoded as the feature node F_{new} and the configuration state is encoded as the configuration node C_{new}. These nodes are added to their respective trees. Otherwise, the node is discarded. After either case, a new random goal is generated and the planning stage loop is restarted. The planning stage loop terminates when the L2 norm distance of the new feature vector $\boldsymbol{s}_{new}$ from the desired features, $\boldsymbol{s}_d$ is less than a threshold r.

When the loop terminates, a trajectory is generated in the configuration state space by backtracking from the current node to the root node. The resultant configuration state trajectory $\boldsymbol{\Gamma}$, will ideally move the robot from the initial configuration to the goal configuration. This trajectory ensures reactionless visual servoing to the goal while avoiding constraint violations in the image space and joint space. The following two sections give a detailed description of the proposed algorithm.

III. Planning in Image Space

This section summarizes the steps involved in generating the new feature state [$\boldsymbol{s}_{new}$, $\dot{\boldsymbol{s}}_{new}$] by exploration in the image feature space.

A. Feature Selection and Initialization

The planning is initiated by exploring the image space for a random goal. A common issue faced during exploration is the choice of features. In the presented framework, image moment based features [14] are being used. Use of image moment based features over conventional point features provides many practical advantages. Since image moments are recalculated in each frame, it not only eliminates the need for feature tracking but also removes the problem of correspondence in the two cameras. Additionally since it is dependent on the sum of all pixels rather than individual pixels, it is also less affected by image noise in comparison to point features. As shown in [14], given a visual feedback, the image moments of order $i+j$, m_{ij} can be calculated very easily. Since there are two cameras being used for visual servoing (one for each arm), the 6 dimensional feature vector takes the form

$$s = \begin{bmatrix} x_{c_1},\ y_{c_1},\ a_1,\ x_{c_2},\ y_{c_2},\ a_2 \end{bmatrix}^T,$$

Here, $[x_{c_1}, y_{c_1}]$ and $[x_{c_2}, y_{c_2}]$ represents pixel coordinates of the centroid (calculated as $[m_{10}/m_{00},\ m_{01}/m_{00}]$) and a_1

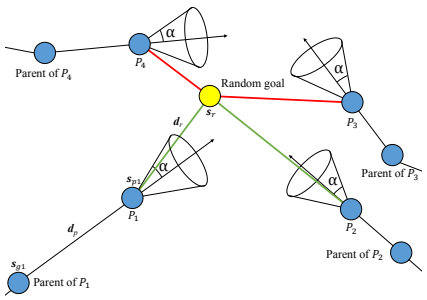


Fig. 2: Selection of candidates for parent node using direction criteria. Each of the right circular cones have a half angle $\boldsymbol{\alpha}$ and axis equal to the direction vector between the candidate node features $\boldsymbol{s}_p$ and its own parents features $\boldsymbol{s}_g$. The cones serve as the boundary for the heading of the direction vector between the random goal features $\boldsymbol{s}_r$ and the candidate node features $\boldsymbol{s}_p$. In this figure the feature vector of nodes P_1 and P_2 validate this criteria and are added to list of candidate parent nodes

and a_2 represents the area of object (calculated as m_{00}) in the images formed by the two cameras respectively. With respect to the 14 DoF space robot, using such a low dimensional feature vector to control a high dimensional robotic system provides Degree-of-Redundancies (DoR) of order 8. This DoR can be used to incorporate other tasks besides the reactionless visual servoing task, such as field of view limit avoidance, joint limit avoidance and singularity avoidance. Once the features are selected, the algorithm begins by initializing two different trees. The first tree is the feature tree $Tree_1$ which stores data about $[\boldsymbol{s},\dot{\boldsymbol{s}}]$, the image moment based features and the respective feature velocities at that instant. The second, the configuration tree $Tree_2$ stores data about $[\boldsymbol{q},\dot{\boldsymbol{q}}]$, the state of the configuration (state of the joints in both the arms and state of the floating base) of the dual arm space robot. $Tree_1$ is initialized by calculating the features $\boldsymbol{s}_1$ from the images captured from the two cameras in their initial states. Since the system starts from rest, the initial feature velocity $\dot{\boldsymbol{s}}_1$ is set to zero. This data is encoded to a feature node F_1 and added to $Tree_1$. $Tree_2$ is directly initialized using the input initial configuration $\boldsymbol{q}_1$. Since the robot starts from rest, the initial configuration velocity vector $\dot{\boldsymbol{q}}_1$ is also set to zero. This data is encoded to $Tree_2$ as the configuration node C_1 . As nodes are being simultaneously added to each of these trees, node i in $Tree_2$ stores the configuration state of the robot that gives the feature state represented by node i in $Tree_1$.

Additionally a bias k is set which represents the frequency of greedy extensions (once for every k iterations) during image space exploration. Finally, a radius r is also set. The radius serves as a threshold for the euclidean distance between the desired features and the latest feature vector $\boldsymbol{s}_{new}$. After initialization, the planning stage loop begins.

B. Random Node Generation and Parent Selection

Each loop begins with the generation of a 6 dimensional random goal feature vector $\boldsymbol{s}_r$. The generated random goal should lie within the limits of the image space, so that the tree does not expand beyond the image space and the object stays completely in view. Additionally, the area of the object in the random goal image must not be too small or too large in comparison to the size of the object in the initial or desired image. These constraints can be expressed as,

$$\begin{aligned} x_{c_{min}} \leq x_c \leq x_{c_{max}}, \\ y_{c_{min}} \leq y_c \leq y_{c_{max}}, \\ min(a_{ini}, a_d) \leq a \leq max(a_{ini}, a_d), \end{aligned}$$

where $x_{c_{max}}$, $x_{c_{min}}$, $x_{c_{min}}$ and $y_{c_{max}}$ are defined by the limits of the image space and a_{ini} and a_d are the intial and desired areas of the object in the image. The random goal $\boldsymbol{s}_r$ is generated by randomly sampling a 6 dimensional space (3 features, 2 camera views) that is bounded by the conditions above. This ensures that the features do not violate the field of view limits. Additionally, after regular intervals of time (depending on bias k), the random goal is set equal to the actual goal $\boldsymbol{s}_d$. This ensures a greedy extension in the image feature space which raises the rate of convergence of the planning stage. Once the random goal has been generated, the parent for the random goal needs to be selected from the nodes present in the feature tree, $Tree_1$. The parent node is selected by applying certain feature based criteria.

1) Direction Criteria: When generating a new branch in the feature tree, it would be favourable that there is a smooth transition between the parent features and the new features. From the nodes already existing in $Tree_1$, the direction criteria generates a list of the candidate parent nodes that will lead to such a smooth transition. This criterion is explained in Fig. 2 where the feature node P_1 has feature vector $\boldsymbol{s}_{p1}$. Let $\boldsymbol{d}_r = \boldsymbol{s}_r - \boldsymbol{s}_{p1}$ represent the direction feature vector between $\boldsymbol{s}_r$ and $\boldsymbol{s}_{p1}$. Additionally, let $\boldsymbol{d}_p = \boldsymbol{s}_{p1} - \boldsymbol{s}_{g1}$ be the direction feature vector between $\boldsymbol{s}_{p1}$ and the features $\boldsymbol{s}_{g1}$ of the parent of P_1. Note that $\boldsymbol{d}_p$ also represents the direction of the feature velocity $\dot{\boldsymbol{s}}_{p1}$ of the parent P_1. The direction criteria simply limits the angle between the two feature direction vectors by some manually defined $\boldsymbol{\alpha}$. It can be mathematically written as,

$$\arccos \frac{\boldsymbol{d}_r . \boldsymbol{d}_p}{|\boldsymbol{d}_r||\boldsymbol{d}_p|} \leq \boldsymbol{\alpha}, \tag{1}$$

This constraint can be visually expressed as setting a right angled circular cone with half angle $\boldsymbol{\alpha}$ and axis $\boldsymbol{d}_p$ as an upper bound on the the heading of $\boldsymbol{d}_r$. In other words, for each node in $Tree_1$, this criteria sets a constraint on the direction of the feature error represented by $\boldsymbol{d}_r$ with respect to the direction of the parent velocity represented by $\boldsymbol{d}_p$. In Fig. 2, for P_1 and P_2, the direction feature vectors (in green) lie within the respective cones, and hence satisfy the criteria. In the same figure, direction feature vectors originating from P_3 and P_4(in red), lie outside their respective cone boundaries. Thus P_3 and P_4 violate the criteria. Each node in $Tree_1$ that satisfies this criteria is added to the candidate parent nodes list mentioned above.

2) Distance Criteria: From the list of candidate parents generated by the direction criteria, the parent node is selected as the node closest to the random goal. This is done by

calculating a distance metric,

$$dist_i = ||\boldsymbol{s}_r - \boldsymbol{s}_i||, \tag{2}$$

where $dist_i$ is the euclidean distance of the feature of the i^{th} node $\boldsymbol{s}_i$ in the list from the random goal. The parent feature node F_p with features $\boldsymbol{s}_p$, feature velocity $\dot{\boldsymbol{s}}_p$ and index $\boldsymbol{n}_p$ in $Tree_1$ is the one which has the minimum value for this distance metric. This results in minimal feature error between the chosen parent node and the random goal. This minimal error ensures that the acceleration impended on the parent feature velocities by the local planner (discussed in next subsection) is minimal, thus allowing for a smooth transition between the feature nodes. Hence, by setting constraints on the direction (using Direction Criteria) and the magnitude (using Distance Criteria) of the feature error, these criteria select a parent node that ensures smooth extension of the feature tree for a random goal $\boldsymbol{s}_r$. Also note that these criteria do not sacrifice the integrity of the RRT since extension is still towards a random goal. The extension of $Tree_1$ will still be in the less explored regions of the feature space.

C. Local Planner

Once the parent feature node is attained, the error in features, $\boldsymbol{e}$ = $\boldsymbol{s}_p$ - $\boldsymbol{s}_r$ can be calculated. Using this error and the parent feature velocity, the local planner calculates an appropriate feature velocity of the new node $\dot{\boldsymbol{s}}_{new}$. It does so by impending acceleration to the parent features as shown in the following equation,

$$\dot{\boldsymbol{s}}_{new} = \dot{\boldsymbol{s}}_p + \beta \frac{(-\lambda \boldsymbol{e} - \dot{\boldsymbol{s}}_p)}{|(-\lambda \boldsymbol{e} - \dot{\boldsymbol{s}}_p)|}\delta. \tag{3}$$

Here $-\lambda \boldsymbol{e}$ is the feature velocity obtained from the exponentially decreasing error control law where λ is the convergence rate. The convergence of the difference between the two velocities $-\lambda \boldsymbol{e}$ and parent feature velocity $\dot{\boldsymbol{s}}_p$ is controlled by a constant β. A smaller value of this constant allows for a more smooth transition in velocity with an increased time of convergence. The acceleration to be impended on the features, δ can be calculated as,

$$\delta = \min(|(-\lambda \boldsymbol{e} - \dot{\boldsymbol{s}}_p)|, a_{bound}),$$

where a_{bound} is an upper bound on the acceleration which is limited by the capabilities of the motors of the joints. A salient feature of this local planner is that it preserves the convergence of the exponentially decreasing error control law, while incorporating acceleration constraints, thus resulting in a smooth transition in the feature space. $\boldsymbol{s}_p$ can now be updated over a small time step dt as,

$$\boldsymbol{s}_{new} = \boldsymbol{s}_p + \dot{\boldsymbol{s}}_{new}.dt.$$

The newly generated feature state [$\boldsymbol{s}_{new}$,$\dot{\boldsymbol{s}}_{new}$] is used to generate the new configuration state using the joint planner.

IV. Joint Space Planning and Reactionless Manipulation

This section focuses on the extension of the configuration tree $Tree_2$ once $\dot{\boldsymbol{s}}_{new}$ is attained. It highlights the steps involved in calculating the instantaneous configuration state [$\boldsymbol{q}_{new}$,$\dot{\boldsymbol{q}}_{new}$] that corresponds to the newly generated feature state [$\boldsymbol{s}_{new}$,$\dot{\boldsymbol{s}}_{new}$]. Additionally, this section also examines the capability of the proposed framework to ensure zero base reactions and handle multiple constraints in the configuration space simultaneously during path-planning of the two arms.

A. Calculation of Reactionless Joint motions

While extending the tree in the configuration space, it is essential that the features move towards the random goal while ensuring negligible base reaction. Assuming that the new joint rate for both arms is collectively represented by $\dot{\boldsymbol{\theta}}_{new}$, the primary task, the visual servoing task is mathematically represented as

$$\boldsymbol{J}_1 \dot{\boldsymbol{\theta}}_{new} = \dot{\boldsymbol{s}}_{new} \tag{4}$$

where $\boldsymbol{J}_1 = \boldsymbol{L}_m \boldsymbol{J}_g$ is called the Modified Image Jacobian. Here, $\boldsymbol{L}_m$ is the interaction matrix [14] which contains combined information from both the camera systems and $\boldsymbol{J}_g$ is the the Generalized Jacobian Matrix (GJM) for the space robot [15]. Given $\tilde{\boldsymbol{I}}_{bm}$, the coupling inertia matrix between the two arms and the base of the system, the secondary task, the reactionless manipulation task can be mathematically represented as [9]

$$\tilde{\boldsymbol{I}}_{bm} \dot{\boldsymbol{\theta}}_{new} = 0, \tag{5}$$

where $\tilde{\boldsymbol{I}}_{bm} \dot{\boldsymbol{\theta}}_{new}$ represents the coupling angular momentum.

Given the primary task $\boldsymbol{J}_p \dot{\boldsymbol{\theta}} = \boldsymbol{v}_p$ and secondary task $\boldsymbol{J}_s \dot{\boldsymbol{\theta}} = \boldsymbol{v}_s$, the solution $\dot{\boldsymbol{\theta}}$ can be directly calculated using the task priority framework [16] as

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_p^+ \boldsymbol{v}_p + (\boldsymbol{J}_s(\boldsymbol{\mathcal{I}}_n - \boldsymbol{J}_p^+ \boldsymbol{J}_p))^+ (\boldsymbol{v}_s - \boldsymbol{J}_s \boldsymbol{J}_p^+ \boldsymbol{v}_p) \tag{6}$$

Upon substituting the 2 tasks represented by (4) and (5) in (6), $\dot{\boldsymbol{\theta}}_{new}$ is obtained as

$$\dot{\boldsymbol{\theta}}_{new} = \boldsymbol{J}_r \dot{\boldsymbol{s}}_{new}, \tag{7}$$

where $\boldsymbol{J}_r = \boldsymbol{J}_1^+ + (\tilde{\boldsymbol{I}}_{bm}(\boldsymbol{\mathcal{I}}_n - \boldsymbol{J}_1^+ \boldsymbol{J}_1))^+ \tilde{\boldsymbol{I}}_{bm} \boldsymbol{J}_1^+$. It is worth noting that (7) represents a novel servoing framework. This control equation represents the reactionless visual servoing control law. The above equation provides a unique optimal solution for the joint velocities of the 2 redundant manipulators of the space robot. It provides a unique mapping from the feature velocity $\dot{\boldsymbol{s}}_{new}$ to the joint velocity $\dot{\boldsymbol{\theta}}_{new}$ that enables reactionless manipulation of the redundant space robot system. It is worthwhile to note that since the joint velocity is attained by imposing nonholonomic constraints that incorporate the dynamics of the system, the path planning framework is kinodynamic in nature. As shown in [9], the generated joint rates can be used to directly calculate the new base velocity vector $\boldsymbol{t}_{b_{new}}$ using the relation,

$$\boldsymbol{t}_{b_{new}} = \boldsymbol{J}_{be}^{-1}(\boldsymbol{t}_e - \boldsymbol{J}_{me} \dot{\boldsymbol{\theta}}_{new}), \tag{8}$$

where $\boldsymbol{t}_e$ is the end effector velocity vector, $\boldsymbol{J}_{be}$ is the Base Jacobian and $\boldsymbol{J}_{me}$ is the Manipulator Jacobian. The above calculated base velocity along with the joint rates form the rate of change of configuration of the robot $\dot{\boldsymbol{q}}_{new}$. This $\dot{\boldsymbol{q}}_{new}$ is integrated over the time step dt, to generate the change

in configuration. This change is used to update the parent configuration $\boldsymbol{q}_p = [\boldsymbol{b}_p^T, \boldsymbol{\theta}_p^T]^T$, where $\boldsymbol{b}_p$ is the base pose and $\boldsymbol{\theta}_p$ is the joint angles of both arms in the parent configuration. Note that the parent configuration has been extracted from node C_p (with index $\boldsymbol{n}_p$) in the configuration tree, the node corresponding to F_p having index $\boldsymbol{n}_p$ in the feature tree. The above steps can be summarized by the following equations,

$$\dot{\boldsymbol{q}}_{new} = [\boldsymbol{t}_{b_{new}}^T, \dot{\boldsymbol{\theta}}_{new}^T]^T$$
$$\boldsymbol{q}_{new} = \boldsymbol{q}_p + \dot{\boldsymbol{q}}_{new}.dt = [\boldsymbol{b}_{new}^T, \boldsymbol{\theta}_{new}^T]^T.$$

B. Joint limits and singularity avoidance

After generating the new configuration state it needs to be validated for several constraints in order to ensure an unobstructed extension in the configuration tree. For joint limit avoidance, each joint angle is simply checked to lie between an upper and lower bound, which are decided by the limitations of the system. Ensuring algorithmic and dynamic singularity avoidance, on the other hand, requires the calculation of a singularity index,

$$\mu = \sqrt{det|\boldsymbol{J}_s {\boldsymbol{J}_s}^T|} \tag{9}$$

where $\boldsymbol{J}_s = (\tilde{\boldsymbol{I}}_{bm}(\boldsymbol{\mathcal{I}}_n - \boldsymbol{J}_1^+ \boldsymbol{J}_1))$. Note that the parameter $\boldsymbol{J}_s$ is dependent on the configuration of the space robot. A lower bound is set for this index. A configuration is considered not singular if its index lies above this lower bound. If the configuration is validated for both the constraints, the new feature state and robot configuration state are encoded as nodes F_{new} and C_{new} respectively and added to their respective trees. If either constraint is violated, the newly generated states are discarded. After either case, a new random goal is generated and the entire path planning loop is restarted.

As mentioned earlier, before starting each loop of the planning process the L2 norm distance between the new feature vector $\boldsymbol{s}_{new}$ and $\boldsymbol{s}_d$ is calculated. When this value is less than a threshold r, the planning stage terminates. Following this, backtracking is done in $Tree2$ from the newest added node (the goal node) until the root node is reached. The result of this random sampling in image space based planning procedure is the configuration state $[\boldsymbol{q}, \dot{\boldsymbol{q}}]$ trajectory $\boldsymbol{\Gamma}$ that ensures reactionless visual servoing of a redundant space robot system, while avoiding various constraints set in both the image and configuration state spaces. Algorithmic implementation of the proposed algorithm is shown in Algorithm 1.

Algorithm 1 Image based path planning for reactionless visual servoing of a dual arm space robot

Input: Initial configuration ($\boldsymbol{q}_1$) and desired image ($\boldsymbol{s}_d$)
Output: Desired robot configuration state trajectory ($\boldsymbol{\Gamma}$)

procedure PLANNING
  $F_1 \leftarrow$ *Feature_node(*$\boldsymbol{s}_1$,$\dot{\boldsymbol{s}}_1$*)*
  $Tree_1 \leftarrow F_1$
  $C_1 \leftarrow$ *Configuration_node(*$\boldsymbol{q}_1$,$\dot{\boldsymbol{q}}_1$*)*
  $Tree_2 \leftarrow C_1$
  $k \leftarrow$ *bias*
  $r \leftarrow$ *radius*
  while $\| \boldsymbol{s}_d - \boldsymbol{s}_{new} \| > r$ **do**
    if modulo(count,k)=0 **then**
      $\boldsymbol{s}_r \leftarrow \boldsymbol{s}_d$
    else
      $\boldsymbol{s}_r \leftarrow$ *RandNode(*$\boldsymbol{s}_1$, $\boldsymbol{s}_d$*)*
    $\boldsymbol{n}_p \leftarrow$*Get_Parent_Index(*$\boldsymbol{s}_r$,$Tree_1$*)*
    $[F_p, C_p] \leftarrow$ *Get_Parent_Nodes(*$\boldsymbol{n}_p$,$Tree_1$,$Tree_2$*)*
    $[\boldsymbol{s}_p, \dot{\boldsymbol{s}}_p] \leftarrow$ *Get_Parent_Feature(*F_p*)*
    $[\boldsymbol{q}_p, \dot{\boldsymbol{q}}_p] \leftarrow$ *Get_Parent_Config(*C_p*)*
    $\dot{\boldsymbol{s}}_{new} \leftarrow$ *Local_Planner(*$\boldsymbol{s}_p, \dot{\boldsymbol{s}}_p, \boldsymbol{s}_r$*)*
    $\dot{\boldsymbol{q}}_{new} \leftarrow$ *Joint_Planner(*$\dot{\boldsymbol{s}}_{new}$*)*
    $[\boldsymbol{s}_{new}, \boldsymbol{q}_{new}] \leftarrow [\boldsymbol{s}_p, \boldsymbol{q}_p] + [\dot{\boldsymbol{s}}_{new}, \dot{\boldsymbol{q}}_{new}] \times dt$
    if *Constraints_Validate(*$\boldsymbol{s}_{new}, \boldsymbol{q}_{new}$*)* **then**
      count $\leftarrow$ *count* + 1
      $F_{new} \leftarrow$ *Feature_node(*$\boldsymbol{s}_{new}$,$\dot{\boldsymbol{s}}_{new}$*)*
      $C_{new} \leftarrow$ *Configuration_node(*$\boldsymbol{q}_{new}$,$\dot{\boldsymbol{q}}_{new}$*)*
      $Tree_1.add\_node[F_{new}]$
      $Tree_2.add\_node[C_{new}]$
      restart while loop
    else
      count $\leftarrow$ *count* + 1
      restart while loop
  $\boldsymbol{\Gamma} \leftarrow$*Get_Trajectory(*$Tree_2$*)*
return $\boldsymbol{\Gamma}$

V. Results and Discussion

In order to validate the presented framework, a 14-DOF dual arm space robot is considered as shown in Fig. 3. The model parameters of the dual arm space robot are shown in Table I. Each of the manipulators in the robot has 7 DoF. Each pair of links in the system are connected by revolute joints. The dual-arm robot is placed in 3D space, and the end-effector of each arm is mounted with a camera (eye-in-hand system). To prove the validity of the proposed algorithm a performance evaluation is done for different goal positions. Additionally, the results also discuss the effects of using the direction criteria (section III B) in the image space and the singularity avoidance constraint (section IV B) in the joint space during the path planning process.

A. Path planning of a 14-DOF dual-arm space robot

In order to validate the proposed algorithm, path planning was carried out on the mathematical model of the 14 DoF dual arm space robot. The object considered is a rectangular plane described by a set of four points in the image space as shown in Fig. 4a and Fig. 4b. In each test case, the camera parameters and the model of the goal was assumed to be known a priori. The initial configuration and the desired moment features were fed to the planning stage. The robot was initially considered to be at rest. The resultant trajectory generated the set of images shown in Fig. 4a and 4b for camera 1 and 2 respectively. The convergence of feature error in Fig. 4c and of joint velocities in Fig 5a to zero highlights the successful completion of the visual servoing task. The negligible base angular velocities in Fig. 4d proves that the

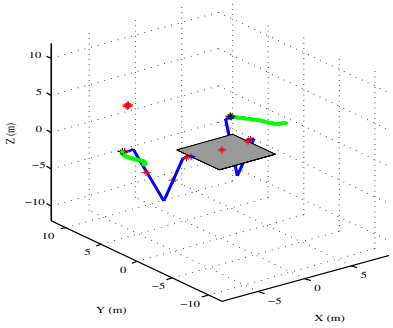


Fig. 3: Model of 14 DoF Dual Arm Space Robot on which the proposed algorithm was implemented (Symmetric case in table II). The robot is in its final state. The green trajectory shows the path followed by the end effector.

TABLE I: Physical parameters for the model of the 14 DoF dual arm space robot system

Parameters	Satellite	Arms 1 and 2						
		Link 1	Link 2	Link 3	Link 4	Link 5	Link 6	Link 7
Mass (kg)	500	3	0.63	0.756	6.85	7.606	0.653	0
Length (m)	0.5 x 0.5	1	1	1	1	1	1	1
I_{xx}	1283.6	0.1	0.1	0	0.1	0.1	0	0
I_{yy}	1283.6	0	0	80.4	115.8	0	0	0
I_{zz}	1283.6	0	0	0	115.8	0	0	0

TABLE II: Numerical results for application of the proposed algorithm on the model of the 14 DoF dual arm space robot. The table highlights the efficacy of the algorithm with changes in the nature of the goals for both the arms and the number of features.

Exp No.	Nature of desired goal	Number of nodes	Number of features	Total Base Reaction (rad.)		Norm of Final Joint Velocity (rad/s)	Norm of Final Error (Pixels)
1	Symmetric	1515	6	1.71E-17	Arm1	9.96E-06	0.8815
					Arm2	1.09E-05	0.4791
		1572	10	1.21E-13	Arm1	1.88E-05	0.8739
					Arm2	1.80E-05	0.4904
2	Moderately unsymmetric	1549	6	5.92E-17	Arm1	1.29E-05	0.8473
					Arm2	9.41E-06	0.5391
		1576	10	6.42E-15	Arm1	2.05E-05	0.8534
					Arm2	1.57E-05	0.5241
3	Extremely Unsymmetric	1694	6	9.48E-17	Arm1	9.83E-06	0.9203
					Arm2	7.22E-06	0.3923

task was completed in a reactionless manner. Additionally it can be seen that the joint angles shown in Fig. 6 lie within the designated limit of -π to π throughout the planned path.

B. Performance Evaluation of the Proposed Algorithm

To prove the efficacy of the proposed algorithm, it was implemented on the space robot model for a variety of constraints and parameters as discussed below.

1) Effect of variation in Desired Goal position and Number of Features : The algorithm was tested for changes in the nature of goals for both the arms and the number of features, as shown in Table II. In terms of number of features, the algorithm was tested for the 6 feature case discussed in section III A. In addition, a 10 feature case, which also takes into account the coordinates of one of the corner points of the rectangle (2 coordinates in 2 cameras) as features, was also used to test the robustness of the algorithm. As is evident from the negligible norm values for both the final joint velocities and feature errors, the path successfully converges for all the cases. Note that the threshold r was set as 1 and bias was set as 1 for every 20 iterations during implementation. Additionally, it can be seen for all the test cases that the total base reaction is almost zero. The different test cases prove that the algorithm is able to conduct visual servoing in a reactionless manner and is robust to the nature of the desired goal and the number of features.

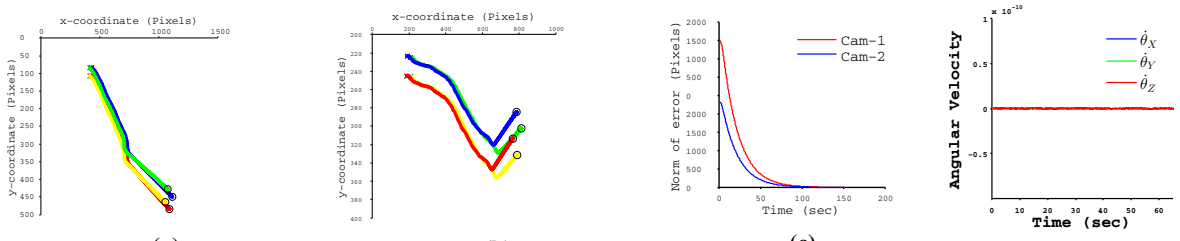
2) Singularity Avoidance: Upon observing the results obtained in Fig. 7, it can be seen that as soon as μ (red in Fig. 7b) falls towards the zero value, the norm of joint velocity vector $\dot{\boldsymbol{\theta}}$ begins to vary erratically (red in Fig. 7a).

To avoid this singularity, a lower bound of 1e+5 was applied on the singularity index μ of the latest configuration. It is only when the index value for the new configuration lies above the lower bound that the new node is validated for the singularity avoidance constraint. The result of applying the constraint during path planning of the space robot was the μ profile in blue in Fig. 7b and a converging norm of joint velocity profile (blue in Fig. 7a).

3) Effect of Direction Criteria: Finally, results were also generated for path planning for the setup used in section V A, but without implementing the Direction Criteria. This resulted in the joint and feature velocity profiles shown in Fig. 5b and Fig. 5d respectively. The velocity profiles for both the joints (Fig. 5a) and features (Fig. 5c) while implementing Direction Criteria were found to be observably smoother than their counter parts. It is owing to the Local Planner in the image space and the Direction Criteria, that $Tree_1$ is able to extend smoothly in the feature space.

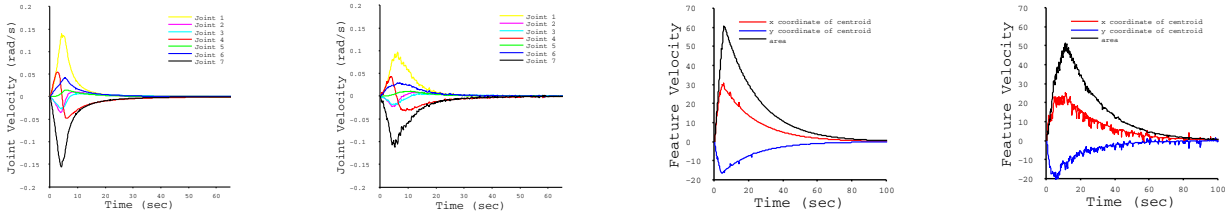
VI. Conclusions and Future Work

In this paper, an image space based random sampling path planning technique for a dual arm space robot is presented. The paper introduces the notion of tree based exploration in the image space using moments based features. Random goals generated in the image space are directly used for planning in the feature and configuration state spaces. The algorithm is able to include the effects of the motion of the manipulators on the base using the reactionless visual servo control law. It is also able to incorporate several constraints in the image space and configuration space by utilising an RRT-based framework. By doing so it is able to achieve reactionless visual servoing without violating constraints such as field of view limits and joint limits and also avoid singularity. To prove its efficacy, the proposed algorithm was also implemented on a numerical model of a 14-DoF dual arm space robot. While the proposed system works well on the numerical model, execution on the real-world setup is not very simple. Setting up a test-bed for reactionless manipulation of a spatial space robot is a challenging task. Execution of the algorithm in our in-house setup [17] will be handled in our future work.



(a) **(b)** **(c)** **(d)**

Fig. 4: Results exhibiting the completion of the reactionless visual servoing task on a 14-DoF free floating dual arm space robot (a) Set of images obtained in camera 1. (b) Set of images obtained in camera 2 (c) Convergence of feature error in both the cameras. (d) Angular velocity profiles of the base.



(a) **(b)** **(c)** **(d)**

Fig. 5: Results exhibiting the effect of the Direction Criteria (D.C.) on the feature and the joint velocities (a) Joint velocity profile for arm 1 with D.C. (b) Joint velocity profile for arm 1 without D.C. (c) Feature velocity profile for cam 1 with D.C. (d) Feature velocity profile for cam 1 without D.C.

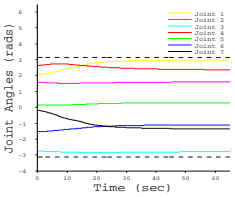
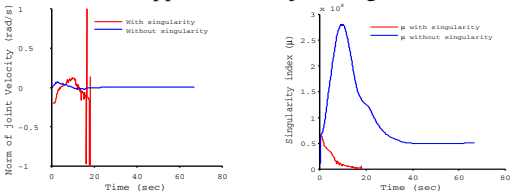


Fig. 6: Joint angle trajectories in arm 1 while following the path generated by the proposed algorithm. The dotted lines represent the limits of -π and π applied on the joint angles.



(a) **(b)**

Fig. 7: Results highlighting the effect of occurrence and avoidance of singularity (a) Behaviour of norm of joint velocity vector $\dot{\boldsymbol{\theta}}$ (b) Behaviour of μ (Singularity index)

ACKNOWLEDGMENT

We would like to thank Dr. K Madhava Krishna for his encouragement and continuous support throughout the duration of the project. Mithun P. is supported by the TCS PhD Research Scholar Program 2014.

REFERENCES

[1] A. Long, M. Richards, and D. E. Hastings, "On-orbit servicing: a new value proposition for satellite design and operation", in *Journal of Spacecraft and Rockets 2007*, vol. 44, no. 4, pp. 964–976.

[2] *On-Orbit Satellite Servicing Study*. National Aeronautics and Space Administration, Goddard Space Flight Center, 2010.

[3] S.M. LaValle and J.J. Kuffner, "Randomized kinodynamic planning", in *Int. Journal of Robotics Research 2001*, vol. 20, no. 5, pp. 378–400.

[4] J. Yakey, S.M. LaValle, and L E. Kavraki, "Randomized path planning for linkages with closed kinematic chains", in *IEEE conf. ICRA 2001*, vol. 17, no. 6, pp. 951–958.

[5] D. Bertram, J. Kuffner, R. Dillmann, and T. Asfour, "An integrated approach to inverse kinematics and path planning for redundant manipulators", in *IEEE conf. ICRA 2006*, pp. 1874–1879.

[6] M. Vande Weghe, D. Ferguson, and S.S. Srinivasa, "Randomized path planning for redundant manipulators without inverse kinematics", in *IEEE-RAS conf. on Humanoid Robots 2007*, pp. 477–482.

[7] Y. Mezouar and F. Chaumette, "Path planning in image space for robust visual servoing", in *IEEE conf. ICRA 2000*,vol. 3, pp. 2759–2764.

[8] A.H. Abdul Hafez, A. K. Nelakanti, and C. V. Jawahar, "Path planning for visual servoing and navigation using convex optimization", in *Int. Journal of Robotics and Automation 2015*,vol. 30(3).

[9] A.H. Abdul Hafez, V.V. Anurag, S.V. Shah, K. Madhava Krishna, and C. V. Jawahar, "Reactionless visual servoing of a dual-arm space robot", in *IEEE conf. ICRA 2014*.

[10] Suril V. Shah, V.V. Anurag, A.H. Abdul Hafez, and K. Madhava Krishna, "Switching method to avoid algorithmic singularity in vision-based control of a space robot", in *IEEE conf. ICAR 2015*, pp. 271–276.

[11] M. Kazemi, K. Gupta, and M. Mehrandezh, "Global path planning for robust visual servoing in complex environments", in *IEEE conf. ICRA 2009*, pp. 326–332.

[12] M. Kazemi,K. Gupta and M. Mehrandezh, "Path planning for image-based control of wheeled mobile manipulators", in *IEEE/RSJ conf. IROS 2012*, pp. 5306–5312.

[13] M. Kazemi, K.K. Gupta, and M. Mehrandezh, "Randomized kinodynamic planning for robust visual servoing", in *IEEE Transactions on Robotics 2013*, vol. 29, no. 5, pp. 1197–1211.

[14] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects", in *IEEE Transactions on Robotics 2005*, vol. 21, no. 6, pp. 1116–1127.

[15] Y. Umetani and K. Yoshida, "Resolved motion rate control of space manipulators with generalized jacobian matrix", in *IEEE conf. ICRA 1989*, vol. 5, no. 3, pp. 303–314.

[16] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators", in *IEEE conf. ICRA 1997*, vol. 13, no. 3, pp. 398–410.

[17] F. James, S. Vyas, P. Bandikatla, P. Mithun P and S.V. Shah , "An Experimental Setup for Testing and Verifcation of Planar Space Robot Simulations", in *Advances in Robotics (AIR), Goa 2015*.