Switching method to Avoid Algorithmic Singularity in Vision-based Control of a Space Robot

Suril V. Shah[1], V. V. Anurag [1], A. H. Abdul Hafez[2], and K Madhava Krishna[1]

***Abstract*—This paper presents a novel approach for algorithmic singularity avoidance for reactionless visual servoing of a satellite mounted space robot. Task priority approach is used to perform visual servoing while reactionless manipulation of the space robot. Algorithmic singularity is prominent in such cases of prioritizing two tasks. The algorithmic singularity is different from the kinematic and dynamic singularities as the former is an artefact of the tasks at hand, and difficult to predict. In this paper, we present a geometric interpretation of its occurrence, and propose a method to avoid it. The method involves path planning in image space, and generates a sequence of images that guides the robot towards goal avoiding algorithmic singularity. The method is illustrated through numerical studies on a 6-DOF planar dual-arm robot mounted on a service satellite.**

I. Introduction

On Orbit Services (OSS) [1], such as orbital detritus management, refurbishment and refuelling of orbiting satellites, construction in space, etc., are going to be an important part of the future's space missions. These OOS will be carried out autonomously with the help of robots to boosts reliability, safety and ease of execution of operations.

The above mentioned OOS would involve closing over manoeuvre of the robot. It is highly desired that the closing over manoeuvre is carried out autonomously due to communication time delay between the service satellite and ground station [2]. Therefore, the use of control technique which uses the on-board vision for successfully completing OOS is inevitable. Vision-based control is one such technique very popular for control of the earth-based robots, and have made many recent progresses [3], [4], [5]. Situation in outer space is further worsen due to satellite drift, variable light conditions, absence of marker, and limited computing power. These make visual servoing problem very challenging [6]. In this work, we attempt to solve problem of algorithmic singularity while doing visual servoing of a dual-arm space robot (Fig. 1) in a reactionlees manner.

Visual servoing was shown for autonomous satellite capture in [2] and later under limited availability of computing power and severe lighting conditions [6]. Robotic capture of a helium airship using visual servoing was demonstrated in [7] using an earth-based experimental set up. It is worth noting that the above works emphasized on capture without considering the reactions transferred on the base satellite due to robot's motion. This would be important in future's space mission as servicing satellites are meant to be lighter in weight and comparable in mass with that of the robot. Hence, robot motion would disturb the attitude of the satellite base if it is not controlled. Use of attitude controller will consume fuel which is reserved mainly for orbital manoeuvres.

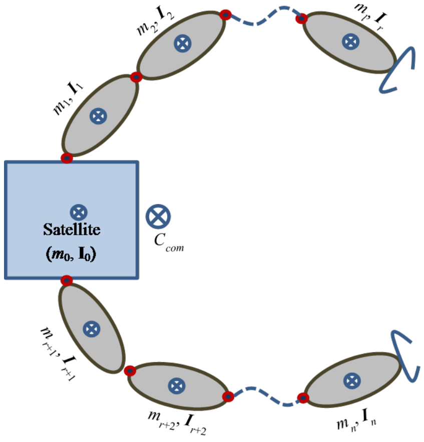


Fig. 1: Schematic of an n-DOF dual-arm space robot. m_i and $\boldsymbol{I}_i$ are mass and inertia tensor of the i^{th} link, and C_{COM} is system's Center-of-Mass.

On the other hand, robot's operation in the free-floating mode using the Generalized Jacobian Matrix (GJM) [8] was proven to be robust [9] and helps in reducing fuel consumption, however, at the cost of change in the orientation of the base satellite. This may destabilise the satellite, and require its re-orientation. Several researchers [9], [10], [11], [12], [13] have also focused on reactionless manipulation of robotic arm. This not only helps in keeping attitude disturbances minimum but also results into fuel saving. These works, however, stressed on reactionless path planning without emphasizing on vision-based control.

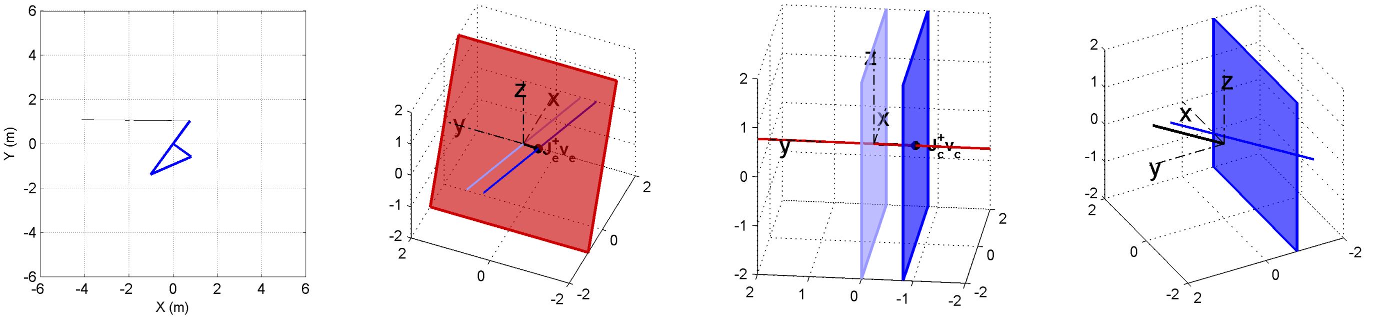
Recently, a novel reactionless visual servoing was presented in [14]. In [14], a task function approach was used, where visual servoing was designed as the primary task whereas reactonless manipulation was designed as the secondary task. However, the task function approach suffers from the potential problem like encountering the algorithmic singularity. At the occurrence of algorithmic singularity, the joint rates increase significant leading to ill-conditioned system. The methods shown in [15], [16], [17] helped in taking care of the algorithmic singularity, however, at the cost of compromise on the secondary task. However, in the present work the secondary task of reactionless manipulation is required to be ensured throughout. The proposed work builds on [14] and the limitations of the work done [15].

In this paper, we present a switching-based path planning framework for algorithmic singularity avoidance for reaction-

[1] S. V. Shah, V. V. Anurag, and K. Madhava Krishna are with International Institute of Information Technology, Hyderabad - 500032, India

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

978-1-4673-7509-2/15/$31.00 2015 IEEE



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

Fig. 3: Geometric demonstration in presence of algorithmic singularity (a-d); (a) shows configuration of the robot at the occurrence of the algorithmic singularity; (b) plots the plane representing least square solution (LNS) space of $\boldsymbol{J}_e$ (in red), its nullspace using a line (in light blue), a vector (in black line) showing LNS of $\boldsymbol{v}_e$, and a line (in dark blue) representing generalized solution obtained by shifting nullspace to the LNS of $\boldsymbol{v}_e$; (c) shows a line (in red) representing LNS space of ($\boldsymbol{J}_c$), a plane (in light blue) representing its nullspace, a vector (in black line) showing LNS of $\boldsymbol{v}_c = \dot{\theta}_2$, and a plane (in dark blue) presenting generalised solution; (d) shows generalized solution of $\boldsymbol{v_e}$ and $\dot{\theta}_2$, and intersection of the generalized solution, by a vector (black line).

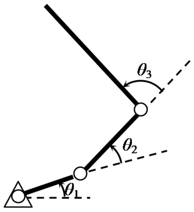


Fig. 2: A 3-link planar arm

less visual servoing of a space robot. The framework performs path planning in the image space and provides sequence of images as an output which guides the robot towards the goal while avoiding algorithmic singularity.

The rest of the paper is organized as follows: Description of algorithmic singularity and its geometric interpretations are provided in Section II, whereas the proposed framework of reactionless visual servoing with algorithmic singularity avoidance is discussed in Section III, Section IV presents results and discussion. Finally, conclusions are given in Section V.

II. Algorithmic Singularity and its Geometric Interpretation

Algorithmic singularity [17] is a common problem for a redundant system executing multiple tasks. It arises when there is a conflict between two or more tasks. Therefore, at the occurrence of algorithmic singularity, it is difficult to satisfy all the task constraints. Given the prominence of algorithmic singularity [15], this section presents its geometric interpretation, which was lacking in the previous literature, for the sack of better understanding of the problem at hand.

For this, let us consider a planar 3-link robot connected by 3 revolute joints, as shown in Fig. 2. The lengths of the links 1, 2 and 3 are taken as 1 m, 2 m, and 3 m, respectively. Here, three independent joint velocities can be chosen as control variables. Initial configuration of the robot is shown in Fig. 3a. Next, primary and secondary tasks are identified as control of end-effector's linear velocity and control of the second joint velocity, respectively. The resulting equations representing primary and secondary tasks are given below

$$\boldsymbol{J}_e\dot{\boldsymbol{\theta}} = \boldsymbol{v}_e \tag{1}$$

$$\boldsymbol{J}_c\dot{\boldsymbol{\theta}} = \dot{\theta}_2 \tag{2}$$

where $\boldsymbol{J}_e \in R^{2\times3}$ the robot Jacobian, $\boldsymbol{v}_e \in R^2$ is end-effector's velocity, $\boldsymbol{J}_c \in R^{1\times3} = [0 \;\; 1 \;\; 0]$, and $\dot{\theta}_2 \in R^1$ is velocity of the second joint. $\dot{\theta}_2$ was chosen as a constant $-0.8 rad/sec$. The solution of (1) and (2) is obtained using task priority framework [15] as

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_e^+\boldsymbol{v}_e + (\boldsymbol{J}_c(\boldsymbol{\mathcal{I}}_n - \boldsymbol{J}_e^+\boldsymbol{J}_e))^+(\dot{\theta}_2 - \boldsymbol{J}_c\boldsymbol{J}_e^+\boldsymbol{v}_e) \tag{3}$$

It is worth noting that the use of (3) can lead to algorithmic singularity when both the tasks are conflicting. This is illustrated next.

Figs. 3b-3d present geometric representation of primary task, secondary task and their common solution at the situation of algorithmic singularity. For primary task, the least norm solution of (1), i.e., $\dot{\boldsymbol{\theta}} = \boldsymbol{J}_e^+\boldsymbol{v}_e$, maps $\boldsymbol{v}_e$ onto a 2D plane (in red) in R^3 and referred to as the Least Norm Solution (LNS) space. Null space of $\boldsymbol{J}_e$ is 1D line, perpendicular to LNS space, as depicted on Fig 3b by a solid line in light blue. For a particular value of $\boldsymbol{v}_e$, the generalised solution is obtained by shifting the nullspace from origin to the point representing least norm solution, i.e., $\boldsymbol{J}_e^+\boldsymbol{v}_e$. This is depicted in Fig 3b by the 1D solid dark blue line parallel to the line representing nullspace (light blue) in Fig 3b. Similarly, for the secondary task $\dot{\theta}_2$, Fig. 3c depicts the LNS space ($\dot{\boldsymbol{\theta}} = \boldsymbol{J}_c^+\dot{\theta}_2$) by a 1D line (in red) in R^3 which coincides with Y-axis. Null space of $\boldsymbol{J}_c$ is a 2D plane in R^3 and also shown in Fig. 3c by a light blue plane passing through origin and perpendicular to Y-axis. The generalized solution to secondary task is depicted by dark blue plane parallel to null space and passing through particular solution, $\boldsymbol{J}_c^+\dot{\theta}_2$.

It is worth noting that the LNS space of the primary task (Fig. 3b) passes through Y-axis, and is aligned with the LNS space of the secondary task (Fig. 3c). This makes the null spaces of the two tasks, and hence their generalized solutions parallel to each other. In fact, nullspace of the primary task is a subspace of nullspace of secondary task. Fig 3d depicts this clearly by showing that the vector representing the intersection of both extends to infinity. This gives rise to large and fluctuating joint rates. Computationally, the common solution represented by (3) is ill conditioned at algorithmic singularity. Such a situation arises only when the LNS spaces are parallel. Mathematically, the algorithmic singularity is encountered when the term $\boldsymbol{J}_c(\boldsymbol{\mathcal{I}}_n - \boldsymbol{J}_e^+\boldsymbol{J}_e)$ is close to being singular or looses rank even though $\boldsymbol{J}_e$ and $\boldsymbol{J}_c$ are full ranked matrices.

III. Reactionless Visual Servoing with Algorithmic Singularity Avoidance

This section presents switching based path planning approach for avoiding algorithmic singularities while performing reactionless visual servoing [14] of a dual-arm space robot shown in Fig. 1. Here, the dual-arm space robot is required to perform the primary task of visual servoing and secondary task of reactionless manipulation. The aim of the proposed method is to generate sequence of image trajectories *Queue* that takes the manipulator to the goal while avoiding algorithmic singularity. This forms the fundamental contribution of this work. It may be noted that the entire path planning is carried out off-line through numerical simulations.

The path planning method involves two phases, *Servoing to Goal Phase* and *Singularity Avoidance Phase*. The planner starts with the former phase executing task priority visual servoing, e.g., the one we proposed in [14], with the final goal image as the desired image. During this phase, images are generated and added to the sequence *Queue* at small intervals of time. A measure μ that aids in detecting algorithmic singularity is tracked. When the measure drops below a certain lower limit μ_{lower}, this indicates reaching the vicinity of algorithmic singularity.

The planner switches to the second phase running an optimization framework to generate further sequence of images. The generated image trajectories during this phase ensures the increment of the measure μ. The second phase continues till the measure of singularity is above a threshold value μ_{upper}. Implementation details of each phase are provided in the following subsections.

A. Servoing to Goal Phase

In this phase, the robot performs visual servo using goal image as desired image and corresponding image error is given by

$$\boldsymbol{e} = (\boldsymbol{s}_{current} - \boldsymbol{s}_{goal}). \tag{4}$$

As the robot requires to servo in a reactionless manner, the primary task of servoing [14] is defined as

$$\boldsymbol{J}_1\dot{\boldsymbol{\theta}} = -\lambda\boldsymbol{e} \tag{5}$$

where $\boldsymbol{J}_1 = \boldsymbol{L}_S\boldsymbol{J}_g$ is the modified Jacobian, $\boldsymbol{L}_S$ is the image Jacobian, and $\boldsymbol{J}_g$ is the Generalized Jacobian Matrix [8]. Next, the secondary task of reactionless manipulation [14] is defined as

$$\tilde{\boldsymbol{I}}_{bm}\dot{\boldsymbol{\theta}} = \boldsymbol{0}. \tag{6}$$

were $\tilde{\boldsymbol{I}}_{bm}$ is the coupling inertia matrix, and $\tilde{\boldsymbol{I}}_{bm}\dot{\boldsymbol{\theta}}$ represents coupling angular momentum. Next, the joint rates are calculated by using (3) as

$$\dot{\boldsymbol{\theta}} = -\lambda\boldsymbol{J}_1^+\boldsymbol{e} + (\tilde{\boldsymbol{I}}_{bm}(\boldsymbol{\mathcal{I}}_n - \boldsymbol{J}_1^+\boldsymbol{J}_1))^+(\boldsymbol{0} - \tilde{\boldsymbol{I}}_{bm}\boldsymbol{J}_1^+(-\lambda\boldsymbol{e})) \tag{7}$$

Interestingly, the above equation can also be represented in the form

$$\dot{\boldsymbol{\theta}} = -\lambda\boldsymbol{J}_r\boldsymbol{e} \tag{8}$$

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 (8) represents a novel servoing framework which takes into account both primary an secondary tasks. After having solution of $\dot{\boldsymbol{\theta}}$, twist of the base ($\boldsymbol{t}_b$) [14], containing its linear velocity and Euler angle rates, is obtained. Now, state of the robot ($\boldsymbol{\theta}$) and base ($\boldsymbol{x}$) is updated for a small time interval δt, i.e.,

$$\boldsymbol{\theta}_{new} = \boldsymbol{\theta}_{current} + \dot{\boldsymbol{\theta}}\delta t \tag{9}$$

$$\boldsymbol{x}_{new} = \boldsymbol{x}_{current} + \boldsymbol{t}_b\delta t \tag{10}$$

Once $\boldsymbol{\theta}_{new}$ and $\boldsymbol{x}_{new}$ is obtained, an image from the current state, i.e.,

$$\boldsymbol{s}_{new} = \mathcal{F}(\boldsymbol{\theta}_{new}, \boldsymbol{x}_{new}), \tag{11}$$

is added to the sequence of images. It is worth noting that $\mathcal{F}$ is the transformation from the state space to the cameras image space. Throughout this phase, a singularity measure is tracked, which is defined as [15]

$$\mu = \sqrt{\boldsymbol{J}_s\boldsymbol{J}_s^T}; \tag{12}$$

where $\boldsymbol{J}_s = \tilde{\boldsymbol{I}}_{bm}(\boldsymbol{I} - \boldsymbol{J}_1^+\boldsymbol{J}_1)$. This phase continues till μ drops below the lower limit μ_{lower}. Note that $\boldsymbol{J}_s$ is function of system's states, $\boldsymbol{\theta}$ and $\boldsymbol{x}$.

B. Singularity Avoidance Phase

When μ drops below the lower limit μ_{lower} the planner switches to *Singularity Avoidance Phase*. This phase essentially creates an intermediate goal image ($\boldsymbol{s}_{int}$) such a way so that the robot moves away for the algorithmic singularity. An optimization framework is used to find $\boldsymbol{s}_{int}$ that maximizes increase in μ. For this, the objective function is defined as

$$h = \max_{\boldsymbol{s}_{int}}(\mu^2) = \max_{\boldsymbol{s}_{int}}(\boldsymbol{J}_s\boldsymbol{J}_s^T). \tag{13}$$

where $\boldsymbol{s}_{int}$ is design variable for the optimization.

The corresponding new error is defined as

$$\boldsymbol{e}_{int} = (\boldsymbol{s}_{current} - \boldsymbol{s}_{int}). \tag{14}$$

which forms control input to (8). It may be noted that finding $\boldsymbol{s}_{int}$ is rather challenging due the fact that (8) is nonholonomic, and hence, μ^2 can not be directly expressed in terms of design variable $\boldsymbol{s}_{int}$. In order to find $\boldsymbol{s}_{int}$ which maximizes h, μ^2 is evaluated using the steps shown below:

$$\begin{aligned}
\boldsymbol{e}_{int} &= (\boldsymbol{s}_{current} - \boldsymbol{s}_{int}) \\
\dot{\boldsymbol{\theta}} &= -\lambda \boldsymbol{J}_r \boldsymbol{e}_{int} \\
\boldsymbol{\theta}_{int} &= \boldsymbol{\theta}_{current} + \dot{\boldsymbol{\theta}} \delta t \\
\boldsymbol{x}_{int} &= \boldsymbol{x}_{current} + \boldsymbol{t}_b \delta t \\
\mu^2 &= \boldsymbol{J}_s \boldsymbol{J}_s^T
\end{aligned} \tag{15}$$

It is also important to ensure that the intermediate goal image is not far away from the current image though in the required direction. So an inequality constraint is formulated as

$$\|\boldsymbol{e}\| < e_T \tag{16}$$

where e_T is chosen in such that it reduces state error while numerical integration, i.e., change in $\boldsymbol{\theta}$ is minimized.

The output of optimization is an intermediate desired image ($\boldsymbol{s}_{int}$). For the optimized $\boldsymbol{s}_{int}$, the $\boldsymbol{\theta}_{int}$ and $\boldsymbol{x}_{int}$ are the new states of the system, i.e., $\boldsymbol{\theta}_{current}$ and $\boldsymbol{x}_{current}$. Moreover, the image from this state, $\boldsymbol{s}_{current}$ is updated as $\mathcal{F}(\boldsymbol{\theta}_{current}, \boldsymbol{x}_{current})$ and is added to sequence of images. The optimization procedure is repeated till μ is above the upper limit, i.e., μ_{upper}, and then, the planner is again switched to *Servo to Goal Phase*. Note that optimization is performed using Matlabs fminsearch function, which is minimum of unconstrained multivariable function using derivative-free method.

The planning completes when the image error in *Servoing to Goal Phase* reduces to the specified magnitude $\boldsymbol{e}_{max}$. The sequence of images obtained during the planning is used by the robot to visual servo through to move from initial position to goal position while avoiding algorithmic singularities. The detailed algorithm is give in Algorithm 1.

IV. Results and Discussions

We implemented the proposed method on a 6-DOF planar dual-arm robot mounted on a service satellite as shown in Fig. 4. Each arm comprises of 3 links connected by revolute joints. Mass of each link is $10kg$ where as that of satellite is taken as $500Kg$. Base is assumed in the free-floating mode, i.e., free to move along X and Y directions and rotate about Z-axis. Initially, the Center-of-Mass (COM) of the base is at origin the (0, 0, 0) with zero orientation, whereas the joints angles are taken as $[0 \; 0.51 \; 2.3 \; 3.1 \; 0.26 \; -2.8]^T rad$. Each arm is provided with a camera on its end-effector, which provides visual feedback for visual servoing. The object to be tracked is a point placed at $[0.1 \; 3 \; -0.1]^T m$, and gives rise to 2 features per camera when projected on to the image plane. It is worth noting that the modified image Jacobian $\boldsymbol{J}_1 \in R^{4\times 6}$ has maximum rank equals to 4 which is less than the number of actuated joints, i.e., 6. Hence, there exits a nullspace, dimension of which is 2. This is utilized in designing the secondary task of reactionless manipulation. Note that the assumption of one point feature per camera is justified as the robot moves in a plane. Next, systematic numerical experiments were carried out to demonstrate efficacy of the proposed approach.

Initially, the robot starts in *Servo to Goal phase* and continues till a point where singularity measure μ drops below μ_{lower}. Lower limit in our experiment is chosen to be $\mu_{lower} = 100$. Fig. 5 shows variation of features in image plane and μ over the time. It can be seen that after $1.9s$, the μ drops below μ_{lower}. As μ is below μ_{lower}, the *Singularity Avoidance phase* is entered to improve μ back to safety above upper threshold. Fig. 6 shows the images obtained along this phase (shown in red) and the plot of μ when the robot servo through those images. The improvement in the value of μ is shown in red.

```
Initialization;
State ← Initial state;
Image ← Initial Image;
Create empty queue of images Queue ;
$e$ ← (Current Image − Goal Image);
Add [First image] to Queue;
while $\|e\| > e_{max}$ do
    $\dot{\theta}$ ← Visual servoing($e$) using (08);
    State ← Current state using (09-10);
    Image ← Current Image using (11) ;
    Add Image to Queue ;
    Find $\mu$ using (12);
    if $\mu < \mu_{lower}$ then
        while $\mu < \mu_{upper}$ do
            Intermed. image ← Run Optimiz. in (13);
            Find $e_{int}$ using (14);
            $\dot{\theta}$ ← Visual servoing($e_{int}$);
            State ← Intermed. state using (15);
            Image ← Intermed. Image using (15);
            Add Image to Queue ;
            Find $\mu$ using (15);
        end
        Current Image ← Intermediate image ;
    end
    $e$ ← (Current Image − Goal Image);
end
Desired path in Image Space ← Image stack;
```

Algorithm 1: Proposed Switching Path Planning

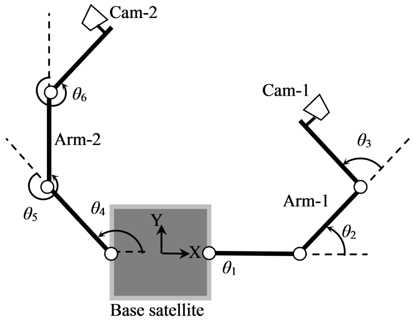
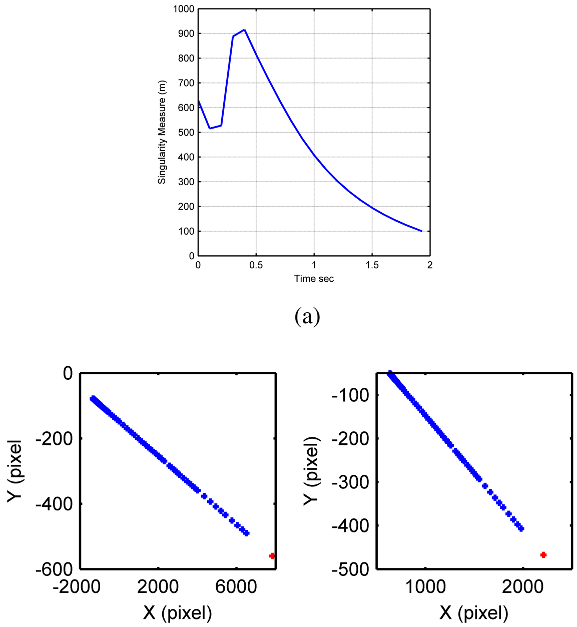


Fig. 4: Schematic of a dual-arm robotic system

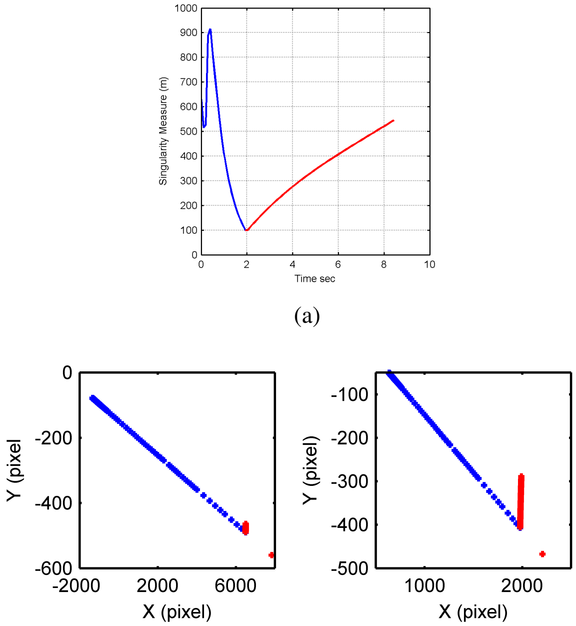
Once μ is over the upper threshold, the simulation resumes to visual servoing towards the goal. In this particular case, μ drops below the lower limit one more time calling for *Singularity Avoidance phase*. Finally, the current image converges to the desired image. The plots showing μ, and images taken by camera 1 and 2 for the entire simulation period is presented in Fig. 7. Fig. 8 shows initial and final states of the system. It may be seen that there is no significant change in the attitude of the



(a)

(b)

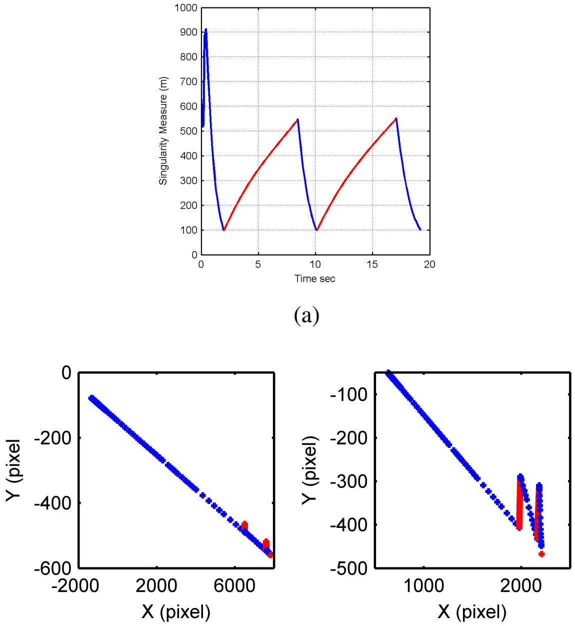
Fig. 5: Initial phase of Visual serving: (a) Singularity measure μ; (b) Variation of features for arm-1 and arm-2



(a)

(b)

Fig. 6: Optimization stage 1: (a) Singularity measure μ (b) Variation of features for arm-1 and arm-2



(a)

(b)

Fig. 7: Optimization stage 2 and visual servoing to the goal: (a) Singularity measure μ (b) Variation of features for arm-1 and arm-2

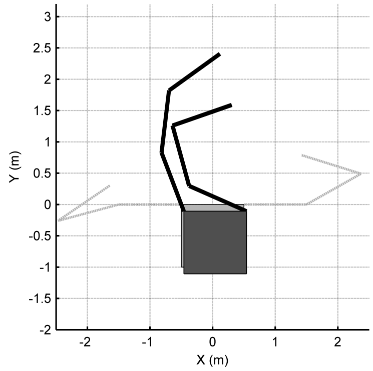


Fig. 8: Initial (grey base and grey links) and final(dark base and dark links) configurations of the manipulator.

base, which prove reactionless manipulation of the dual-arm robot. It may be noted that when this experiment was carried out without the proposed path planing framework, simply by using the standard task priority approach, the simulation failed after $3s$ at the occurrence of algorithmic singularity.

Another simulation was run, this time with lower limit and upper limits decreasing after every cycle. The results are shown in Fig. 9. After every cycle, the lower and upper limits of μ were divided with the number of cycles performed. This is done in order to incorporate goal that actually requires lower singularity measure to approach. In such a case, putting fixed limits may take many iterations to reach the goal. But with

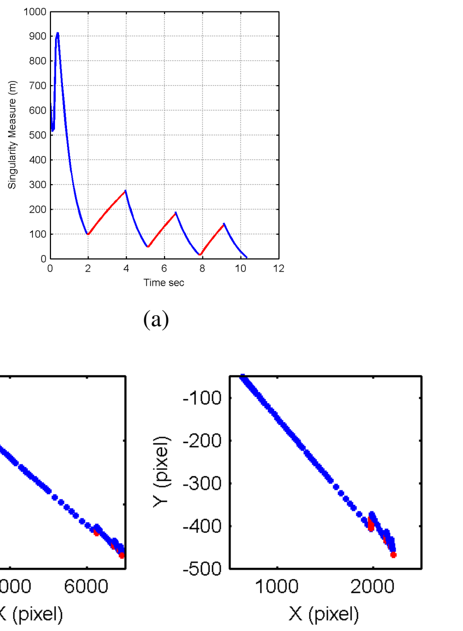


Fig. 9: Optimization stage 2 and visual servoing to the goal: (a) Singularity measure μ (b) Variation of features for arm-1 and arm-2

decreasing limits, it can be ensured that the robot reaches the goal quickly after only a few iterations. Several cases in which manipulator running into algorithmic singularity are simulated and the proposed method successfully helped in bypassing the algorithmic singularity and complete visual servoing task. This proves efficacy of the proposed framework.

V. Conclusions

We have proposed a novel approach of path planning in image space targeting the problem of algorithmic singularity for reactionless visual servoing of the redundant space manipulators. The method generates sequence of images that a robot can use to servo from the initial position to the final position. The planning is done in two phases, *Servoing to Goal phase* in which the robot advances towards the goal image, and *Singularity Avoidance phase* in which an optimization framework is used to move the robot towards improving its singularity measure. The planner switches between these two phases according to the tracked singularity measure, and completes with *Servoing to Goal phase*. Finally, the generated sequence of images is used to servo the robot. The method has been successfully implemented on a 6-DOF planar dual-arm space robot to perform reactionless visual servoing with algorithmic singularity avoidance. The proposed framework is generic, and applicable to any redundant system that uses task priority approach.

It may be noted that the two phases in the path planner work like a switching algorithm which highlights one of the limitations that the planner can take many cycles to complete the task successfully. This can happen when the direction of advancement in *Servo to Goal phase* and direction of improvement in *Singularity Avoidance phase* are closely aligned and in opposite directions. Another limitation is that completion of the tasks also depends on how the singularity measure limits are set. Future work would be in lines of improving the over all performance of the method by improving switching conditions, and modifying the constraints such that the images towards the goal are given more weightage in *Singularity Avoidance phase.*

Acknowledgment

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

References

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

[2] N. Inaba and M. Oda, “Autonomous satellite capture by a space robot: world first on-orbit experiment on a japanese robot satellite ets-vii,” in *IEEE ICRA*, vol. 2, 2000, pp. 1169–1174.

[3] A. H. Abdul Hafez and C. V. Jawahar, “Visual servoing by optimization of a 2d/3d hybrid objective function,” in *IEEE ICRA*, 2007, pp. 1691–1696.

[4] R. Ozawa and F. Chaumette, “Dynamic visual servoing with image moments for a quadrotor using a virtual spring approach,” in *IEEE ICRA*, 2011, pp. 5670–5676.

[5] A. Cherubini, F. Spindler, and F. Chaumette, “A new tentacles-based technique for avoiding obstacles during visual navigation,” in *IEEE ICRA*, 2012, pp. 4850–4855.

[6] N. Inaba, M. Oda, and M. Hayashi, “Visual servoing of space robot for autonomous satellite capture,” *Tran. of the Japan Society for Aeronautical and Space Sciences*, vol. 46, no. 153, pp. 173–179, 2003.

[7] D. Crymble and I. Sharf, “Experiments on robotic capture of objects in space,” in *55th Int. Astronautical Congress*, 2004.

[8] S. K. Saha, “A unified approach to space robot kinematics,” *Tran. on Robotics and Automation*, vol. 12, no. 3, pp. 401–405, 1996.

[9] K. Yoshida, K. Hashizume, and S. Abiko, “Zero reaction maneuver: flight validation with ets-vii space robot and extension to kinematically redundant arm,” in *IEEE ICRA*, vol. 1. IEEE, 2001, pp. 441–446.

[10] P. Piersigilli, I. Sharf, and A. Misra, “Reactionless capture of a satellite by a two degree-of-freedom manipulator,” *Acta Astronautica*, vol. 66, no. 1, pp. 183–192, 2010.

[11] S. V. Shah, I. Sharf, and A. K. Misra, “Reactionless path planning strategies for capture of tumbling objects in space using a dual-arm robotic system,” in *AIAA Guid., Navigation and Cont. Conf.*, 2013.

[12] S. V. Shah, A. Gattupalli, and A. Misra, “Energy optimum reactionless path planning for capture of tumbling orbiting objects using a dual-arm robot,” in *Proc. of the 1st Int. and 16th Nat. Conf. on Machines and Mechanisms, India*, 2013.

[13] A. Gattupalli, S. V. Shah, K. M. Krishna, and A. Misra, “Control strategies for reactionless capture of an orbiting object using a satellite mounted robot,” in *Proc. of Advances In Robotics*, 2013, pp. 1–6.

[14] 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 ICRA*, 2014.

[15] S. Chiaverini, “Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators,” *IEEE Tran. on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.

[16] C. A. Klein, C. Chu-Jenq, and S. Ahmed, “A new formulation of the extended jacobian method and its use in mapping algorithmic singularities for kinematically redundant manipulators,” *Tran. on Robotics and Automation*, vol. 11, no. 1, pp. 50–55, 1995.

[17] G. Marani, J. Kim, J. Yuh, and W. K. Chung, “Algorithmic singularities avoidance in task-priority based controller for redundant manipulators,” in *IEEE/RSJ IROS*, vol. 4. IEEE, 2003, pp. 3570–3574.