ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 9(41), DOI: 10.17485/ijst/2016/v9i41/104099, November 2016*

Trajectory Planning for a Six Axis Manipulator for SFF-Inspired Depth Estimation

C. R. Srinivasan[1]*, R. Senthilnathan[2], P. Subhasree[3], R. Sivaramakrishnan[3], R. Srividya[1] and P. Karthikeyan[3]

[1]Faculty of Manipal Institute of Technology, Manipal University, Manipal - 576104, Karnataka, India; cr.srinivasan@manipal.edu,
[2]Department of Mechatronics Engineering, SRM University, Kattankulathur - 603203, Tamil Nadu, India; senthilnathan.r@ktr.srmuniv.ac.in
[3]Division of Mechatronics, Department of Production Technology, MIT Campus, Anna University, Chennai-600025, Tamil Nadu, India; subhasreesenthilnathan@gmail.com, srk@mitindia.edu, pkarthikeyan@annauniv.edu

Abstract

Objectives: Trajectory planning is the most vital procedure in every continuous path control based application of robotic arm type manipulators. This paper presents the methodology adopted for planning a trajectory required to be followed by the eye-in-hand camera mounted near the end-effector of a six axis manipulator. The basic purpose is to acquire a sequence of images by translating the camera along the optical axis of the camera which is meant to be used by an algorithm inspired by Shape from Focus (SFF). **Methods/Statistical Analysis:** The movement of the camera (the end effector) is controlled in the task space of the robot a Cartesian space approach is presented where-in the inverse kinematics computations needs to be performed at run time. All the trajectory planning and simulation of the robot's movements are demonstrated in the simulation where the cues for camera motion are obtained from the SFF-inspired algorithm. Since the SFF-inspired algorithm requires a linear trajectory to be followed by the camera and manipulator is an all-revolute joint based one, careful understanding of the workspace of the robot for singularities is of prime importance. The ability of the manipulator to orient in three axes at the various reachable positions of the workspace is estimated by a manipulability measure. The linear trajectory to be followed by the end-effector is planned in the region of the workspace which has the highest manipulability. **Findings**: The paper describes the various steps involved in the process of understanding the manipulability of the workspace and planning of the trajectory in accordance with the manipulability index. The details of the trajectory planning to meet the requirements are clearly illustrated. **Application/Improvements:** Though the intended purpose of planning the trajectory is specific to a computer vision task, the methodology demonstrated is applicable for any application involving linear trajectories. The scope for future work in continuation of the current work is in the direction of dynamics affected trajectory planning.

Keywords: Linear Trajectory, Manipulability, SFF, Trajectory Planning

1. Introduction

Recent trends in robotics have focused on converting the robots into an information-intensive mechanism, with the aid of advanced sensing technologies, especially for the external state of the robots by means of the category of sensors, called extrinsic sensors. Vision is one of the most powerful external state sensing technologies available, the field of study of which is computer vision. The attributes of perception from vision include color, texture, shapes,

*Author for correspondence

depth, etc. perceive the environment. The attributes of perception from vision include color, texture, shapes, depth, etc. Scene reconstruction in the field of computer vision has been the topic of interest for over three decades, where a 3-D model of the scene is generated from computations made on one or more images of the scene. SFF is one of the popular methods for scene reconstruction which generally requires more number of images along the different focal stacks of the lens. The SFF techniques have been successfully used in industrial inspection, medical diagnostics involving microscopic imaging or imaging of small objects[1]. The SFF techniques demand images of the scene with different focus levels which can be obtained by translating the camera or the object or by changing the focus setting of the lens. One of the inherent limitations of the SFF method of reconstruction of the scene is that they are highly sensitive to parallax. Many authors in the recent in years have attempted to extend the applicability of SFF techniques by means of novel image processing procedures[2]. The attempts made for increasing the applicability of the SFF technique is purely for complete scene reconstruction rather than using some approximate information from SFF with an imaging system prone to parallax errors. In the research work carried out by the authors an algorithm inspired by SFF is developed wherein in the lines of conventional SFF a sequence of images needs to be acquired from various focal distances. This is achieved using a robotic manipulator which is served by a vision system to impart understanding of the geometry of the object to be manipulated by the robotic arm. This paper reports a small part of the research work which presents the procedural information of the trajectory planning to achieve the camera motion as demanded by the SFF-inspired algorithm.

2. Robotic Manipulator

The robot used in the simulation study specifically fabricated, to perform real time visual servoing on the actual scale. The robot based visual servoing was carried out at the simulation level. The manipulator considered in the study is a serial manipulator, with six revolute joints and a spherical wrist which is specifically chosen to have all orientations possible. Each joint variable is defined with a single variable and the number of joints equals the number of degrees of freedom, which amounts to 6. The CAD Model of the manipulator is shown in Figure 1.The simulated model is created by analyzing the kinematic model of the robot, and its representation in the Robotics Toolbox for MATLAB[1]. The conventions adopted in the toolbox, are based on the general method of representing kinematics and dynamics of serial-link manipulators. The robotic manipulator modelled using the Robotics Toolbox is shown in Figure-2.

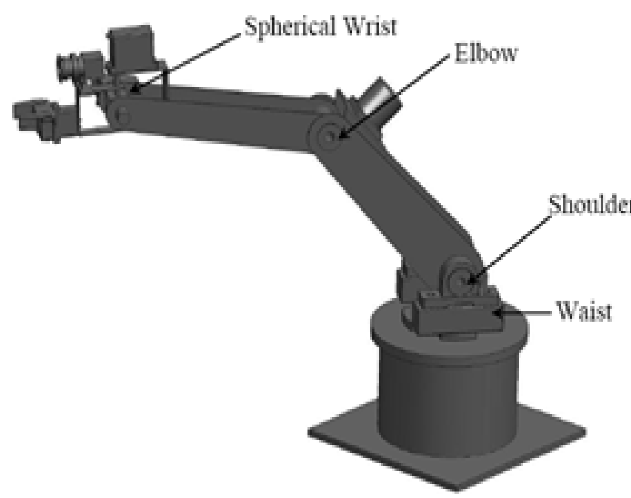

Figure 1. CAD Model of the Robotic Manipulator.

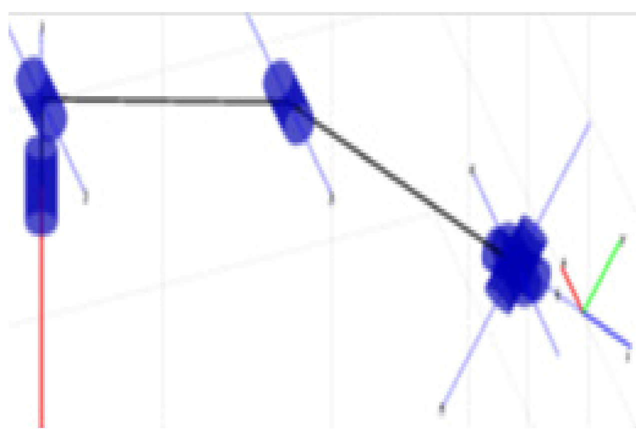

Figure 2. CAD Model of the Robotic Manipulator.

3. Forward Kinematics

This section of the paper deals with the application of the spatial descriptions of a rigid body presented in the third chapter in robotics, particularly to the manipulator configuration considered. Robot kinematics is the study of the motion (position, velocity and acceleration) of the various links of the robot, without considering the forces causing the motion. A basic problem to be solved is to relate the robot's joint configuration to the position and orientation of its end effector. The configuration of a 6-DoF robot is a 6-vector space ($q_1, q_2, \ldots, q_6$), where each q_i is the rotational joint angle. This is known as the forward kinematics of the robot. The origin of the world coordinate frame *{W}* and the end-effector is attached with a coordinate frame *{E}*. The frame *{E}* can be represented with respect to the world coordinate frame *{W}*, through a position vector ${}^{W}p_{E}$ which locates the origin of the frame *{E}*, and a rotation matrix which specifies the orientation. The two parameters can be combined as a single entity in the form of a homogeneous transformation matrix, which would then describe the relationship between the frames *{W}* and *{E}*. The robotic arm used in this research work is a floor-mounted configuration; hence, the method should be developed link by link starting from the ground fixed link. The D-H method of obtaining the forward kinematic transformation is a popular approach which first starts with frame assignments to the robotic arm followed by estimation of the four D-H parameters for each link. The frame assignments for the robotic arm are shown in Figure-3.

Table-1 shows the set of D-H parameters of the manipulator. The parameters shown in the table were experimentally determined, by measuring the corresponding features of each link.

The workspace of the robot arm is shown in Figure-4. The workspace is obtained only by considering a limited range of joint angles (shown in degrees), since the maximum range of motion of the rotational joints in a robot generally does not exceed 270 degrees.

Once the D-H parameters of the robot are obtained, the final homogeneous transformation which relates the base frame to the end-effector frame may be obtained. When building the homogeneous transformation for the various links in the robot, the elements mentioned in Table 1can be compacted into a matrix of size 4 × 4. The homogeneous transform relating two consecutive frames say {n} and {m} is given by:

$$ {}^{m}T_{n} = \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta.\sin\alpha & a\cos\theta \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & a\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

Applying the D-H parameters to various frames assigned, the following matrices may be obtained.

Table 1. D-H Parameters

Joint	*n*	Variable, *θ* (in radians)	Offset, *d* (in m)	Length, *a* (in m)	Twist Angle, *α* (in radians)
Waist	1	θ_1	0.08	0	−π/2
Shoulder	2	θ_2	0	0.30	0
Elbow	3	θ_3	0	0.34	0
Pitch	4	θ_4	0	0	−π/2
Yaw	5	θ_5	0	0	π/2
Roll	6	θ_6	0	0	0

$$^{0}T_{1}=\begin{bmatrix} \text{cosè}_1 & 0 & -\text{sinè}_1 & 0 \\ \text{sinè}_1 & 0 & \text{cosè}_1 & 0 \\ 0 & -1 & 0 & 0.08 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$^{1}T_{2}=\begin{bmatrix} \text{cosè}_2 & -\text{sinè}_2 & 0 & 0.3\text{cosè}_2 \\ \text{sinè}_2 & \text{cosè}_2 & 0 & 0.3\text{sinè}_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$^{2}T_{3}=\begin{bmatrix} \text{cosè}_3 & -\text{sinè}_3 & 0 & 0.34\text{cosè}_3 \\ \text{sinè}_3 & \text{cosè}_3 & 0 & 0.34\text{sinè}_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$^{3}T_{4}=\begin{bmatrix} \text{cosè}_4 & 0 & -\text{sinè}_4 & 0 \\ \text{sinè}_4 & 0 & \text{cosè}_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$^{4}T_{5}=\begin{bmatrix} \text{cosè}_5 & 0 & \text{sinè}_5 & 0 \\ \text{sinè}_5 & 0 & -\text{cosè}_5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$$^{5}T_{6}=\begin{bmatrix} \text{cosè}_6 & -\text{sinè}_6 & 0 & 0 \\ \text{sinè}_6 & \text{cosè}_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Apart from the description of the links and joints the precise location of the first frame, {*0*} and TCP frame, {*E*}, with respect to the universal reference frame {*W*} is of reasonable importance. The homogeneous transformation matrix form of the relations is as follows:

$$^{W}T_{0}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } {}^{6}T_{w}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

In the above matrices $\theta_1, \theta_2, \ldots \theta_n$ are the independent variables in the matrix, since all the joints are revolute joints. The transformation matrix defining the pose of the end-effector of the robot with respect to the base frame, which basically is the kinematic framework of the manipulator, can be represented in a compact form as $\boldsymbol{\chi} = {}^{0}\mathbf{T}_{6}(\mathbf{q})$ with $\mathbf{q}$ being $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^{T}$.

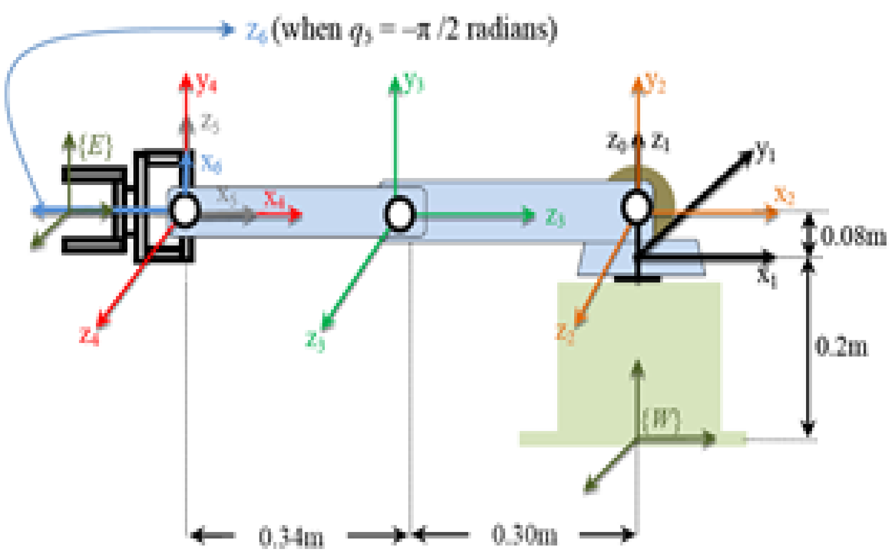

Figure 3. Frame Assignments.

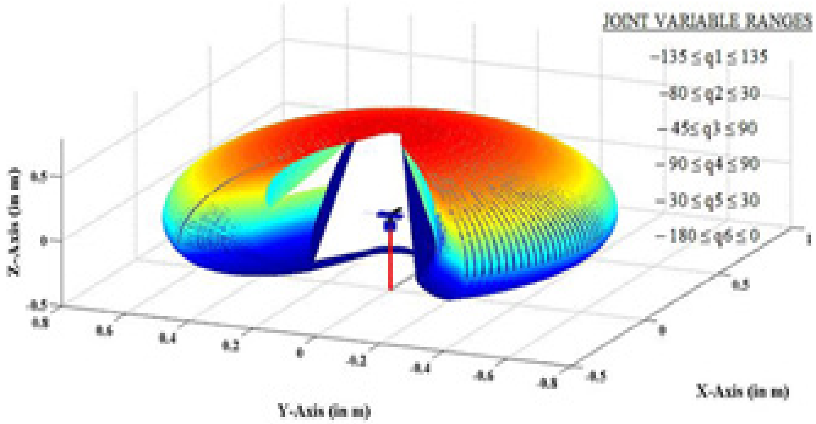

Figure 4. Workspace of the Robot.

4. Inverse Kinematics

The inverse kinematics problem attempts to compute the joint space vector, **q** given the end-effector's pose vector, **χ**. Solving the inverse kinematics problem is important, because in many situations the manipulator motion is defined in the task space, which defines trajectories. The joint angles are computed using inverse kinematics, to make the end-effector track the desired trajectory. Solving the inverse kinematics problem analytically is straightforward, for manipulators with a few degrees of freedom, though as the degrees of motion of the manipulator increase it becomes difficult. In the case of a 6-DoF manipulator, assuming no mechanical constraints (clearly an ideal case), it is possible to obtain at least 16 different inverse kinematics solutions for a given pose vector defining the end-effector's position and orientation[4]. As mechanical constraints are introduced the number of solutions decreases. In general, for a 6-DoF robot, the inverse kinematics solution would be difficult to obtain analytically, except for two special cases[5]. The cases are either the presence of a spherical joint or a planar pair anywhere in the kinematic chain. In the current work, the robot is designed with a spherical wrist intentionally, to simplify the derivation of the inverse kinematics solution. The spherical wrist paves the way for partial decoupling of the position and orientation. It can be observed that joints q_1, q_2 and q_3 contribute to position, and the remaining joint variables q_4, q_5 and q_6 contribute to different orientations about the positioned point. The various joint angles are found in a decoupled manner, and the corresponding expressions are as follows. Referred to Figure-5, the position of the point p_3 depends on the joint variables q_1, $q_2$and q_3. These three values are computed, depending on the location of point p_3. The value of the joint q_3 may be found as

$$q_3 = \frac{\pi}{2} - \varphi \tag{9}$$

where,

$$\varphi = \arccos\left(\frac{0.30^2 + 0.34^2 - |R_3|^2}{2 \cdot 0.30 \cdot 0.34}\right) \tag{10}$$

The above expression is obtained from the law of cosine,

$$|R_3|^2 = 0.3^2 + 0.34^2 - 2 \cdot 0.3 \cdot 0.34 \cdot \cos\varphi \tag{11}$$

and,

$$|R_3| = \sqrt{\left(p_{3x}^2 + p_{3y}^2 + \left(p_{3z}^2 + 0.2\right)^2\right)} \tag{12}$$

The joint variable q_2 may be derived as

$$q_2 = -(\beta_1 + \beta_2) \tag{13}$$

where,

$$\beta_1 = \arcsin\left(\frac{|p_z| - 0.2}{|R_3|}\right) \text{ and}$$

$$\beta_2 = \arccos\left(\frac{0.30^2 + |R_3|^2 - 0.34^2}{2 \cdot 0.30 \cdot |R_3|}\right)$$

The joint variable for the waist joint may be found as

$$q_1 = \text{arctan2}\,(p_{3y}, p_{3x}) \tag{14}$$

and the variable q_5 from the expression,

$$q_5 = \arccos\,({}^{W}z_3 \cdot a) \tag{15}$$

The calculation of q_5 is as follows:

$${}^{W}T_E = {}^{W}T_1 \cdot {}^{1}T_2 \cdot {}^{2}T_3 \cdot {}^{3}T_4 \cdot {}^{4}T_5 \cdot {}^{5}T_6 \cdot {}^{6}T_W \tag{16}$$

$${}^{W}T_E = {}^{W}T_3 \cdot {}^{3}T_6 \tag{17}$$

$${}^{3}T_6 = [{}^{W}T_3]^{-1} \cdot {}^{W}T_E \tag{18}$$

$${}^{3}T_6 = [{}^{W}T_1 \cdot {}^{1}T_2 \cdot {}^{2}T_3]^{-1} \cdot {}^{W}T_E \tag{19}$$

The right side of the equation is known, since the required pose of the end-effector, ${}^{W}T_6$ is given, and the remaining part is known from the D-H matrices. The left side of the equation depends on q_4, q_5 and q_6 out of which q_5 is available

$${}^{W}_{3}\mathbf{T} = \begin{pmatrix} {}^{W}x_3 & {}^{W}y_3 & {}^{W}z_3 & {}^{W}p_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{20}$$

Rather than multiplying out the right side of the equation, an easier way is to solve q_4 by eliminating q_6. This can be done by considering the rotation matrices alone, since in a spherical wrist the transformations between the joints are related only by rotations.

$${}^{3}R_6 = {}^{3}R_4 \cdot {}^{4}R_5 \cdot {}^{5}R_6 \tag{21}$$

$${}^{3}R_6 \cdot {}^{5}R_6^{-1} = {}^{3}R_4 \cdot {}^{4}R_5 \tag{22}$$

$${}^{3}R_6 \cdot {}^{5}R_6^{T} = {}^{3}R_4 \cdot {}^{4}R_5 \tag{23}$$

Since in the rotation matrices, only the final column contributes, the other columns of the matrix are not shown in the expressions below, so as to avoid visual clutter.

$${}^{3}\mathbf{R}_4\, {}^{4}\mathbf{R}_5 = \begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 \\ \sin\theta_4 & 0 & -\cos\theta_4 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta_5 & 0 & -\sin\theta_5 \\ \sin\theta_5 & 0 & \cos\theta_5 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} \cdots & \cdots & -\cos\theta_4 \sin\theta_5 \\ \cdots & \cdots & -\sin\theta_4 \sin\theta_5 \\ \cdots & \cdots & \cos\theta_5 \end{bmatrix}$$

$${}^{3}\mathbf{R}_6\, {}^{5}\mathbf{R}_6^{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} \cos\theta_6 & \sin\theta_6 & 0 \\ -\sin\theta_6 & \cos\theta_6 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cdots & \cdots & r_{13} \\ \cdots & \cdots & r_{23} \\ \cdots & \cdots & r_{33} \end{bmatrix}$$

From the above expression, it is possible to eliminate q_6 and calculate q_4. The last column on the right side of the above equations can be equated to obtain

$$\arctan\left(\frac{23}{13}\right) \tag{24}$$

The joint variable q_6 can be computed as

$$q_6 = \arccos\,({}^{W}y_5 \cdot o) \tag{25}$$

where ${}^{W}y_5$ is known from

$${}^{W}_{5}\mathbf{T} = \begin{pmatrix} {}^{W}x_5 & {}^{W}y_5 & {}^{W}z_5 & {}^{W}p_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{26}$$

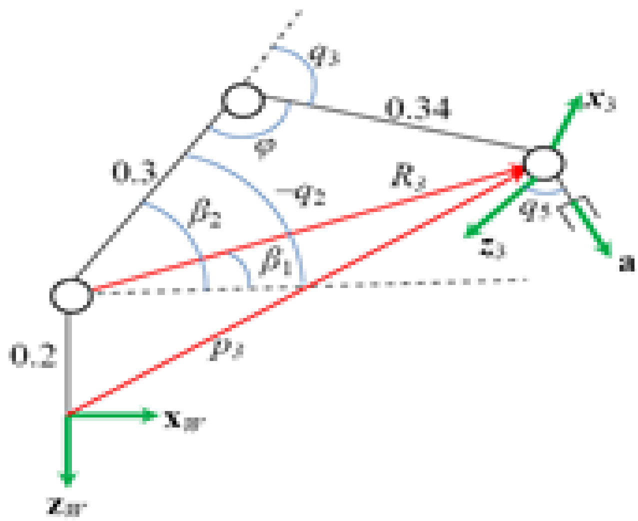

Figure 5. Inverse Kinematics.

5. Motion Analysis

This section of the paper deals with the mappings from the joint velocities to the corresponding end-effector translational velocities. These parameters are related by a term called Jacobian. The importance of the Jacobian in the scope of the thesis is to understand the singular regions in the workspace. The trajectories planned for the camera motion are based on the knowledge of singular regions, where some orientations of the end-effector are not possible. Another form of understanding is that, in the singular regions the joint velocities would be very high, tending to infinity sometimes. A minimal representation can be used expressing $\dot{\chi} = v$ leading to

$$v = \begin{bmatrix} \dot{\mathbf{p}} \\ \omega \end{bmatrix} \tag{27}$$

where $\dot{\mathbf{p}}$ is the Cartesian linear velocity vector, and ω is the angular velocity. The Jacobian matrix **J** relates the translational motion and the angular velocity of each link. This can be realized, by writing it in the explicit form as follows:

$$v = \begin{bmatrix} \mathbf{J}_p \\ \mathbf{J}_\omega \end{bmatrix} \dot{\boldsymbol{q}} \tag{28}$$

The above expression can be decomposed into two parts as $\dot{\mathbf{p}} = \mathbf{J}_p(\boldsymbol{q})\dot{\boldsymbol{q}}$ and $\omega = \mathbf{J}_\omega(\boldsymbol{q})\dot{\boldsymbol{q}}$. The usage of $\boldsymbol{q}$ indicates that the Jacobian matrix of the robot depends on the joint space vector. It is this nature of the Jacobian matrix, relating velocities in joint space to velocities in task space, which makes it a convenient quantity to be used for analyzing singularities.

6. Performance Measures

The task space trajectories to be followed by the robot should be carefully selected, such that the point along the trajectory is not a part of the singular regions of the workspace. Many authors have proposed measures to evaluate the workspace of a robotic manipulator to identify positions where some orientation may not be possible. The performance measures for the robot's ability to locate the end-effector in a particular position and orientation in the workspace, is of high importance in the design, analysis, evaluation and optimization of the robot's mechanical system. Generally, the measure is a scalar, which quantifies how well the system behaves, with regard to force and motion transmission. The measures include service angle[6], conditioning[7], and measures based on the Jacobian matrix of the robot. Some of the commonly used measures are the Jacobian based performance measures, such as manipulability, singularity and dexterity[8]. The use of the Singular Value Decomposition (SVD) of the Jacobian matrix offers further mathematical insight into the manipulability characteristics. In this research, the end-effector should follow a straight line trajectory in the task space, during the stage of image acquisition for the SFF-inspired algorithm. In order to perform this, the robotic arm must have the ability to maintain a particular orientation for the set of via points of the desired trajectory. This is decided by the manipulator configuration, and the dexterity of the manipulator in the workspace. The workspace analysis is carried using a quantitative measure called Manipulability[9]. The manipulability measure is a generalized concept of the determinant of the Jacobian matrix. Manipulability is the measure of the manipulating ability of the robotic arms in positioning and orienting its end-effectors. The numbers obtained for the measure can be used for understanding the dexterity that the robot has in different regions of the workspace. The dexterity of the robot is good in regions, where the value of manipulability is high. The manipulability index may be defined, based on the Jacobian matrix of the robot for a given joint configuration $\boldsymbol{q}$ as follows:

$$H = \sqrt{\det(\boldsymbol{J}(q)\boldsymbol{J}^T(q))} \tag{29}$$

Figure-6 shows the 2-D projection of the surface plot of the manipulability index, against the task space position variables **x** and **z**. The manipulability index is encoded to the color of the plot, as shown in the plot's legend. The projection is done on the **x**-**z** plane, since the translational trajectory that the eye-in-hand camera should follow, is along the **z**-axis (in line with the optical axis of the camera).

The observation is that, the value of the manipulability index is very low, which is mainly due to the fact that the translational manipulability is extremely poor for a robot with all rotational joints. It is from this plot that a suitable trajectory can be selected, which guarantees feasible solutions for inverse kinematics of the manipulator. The

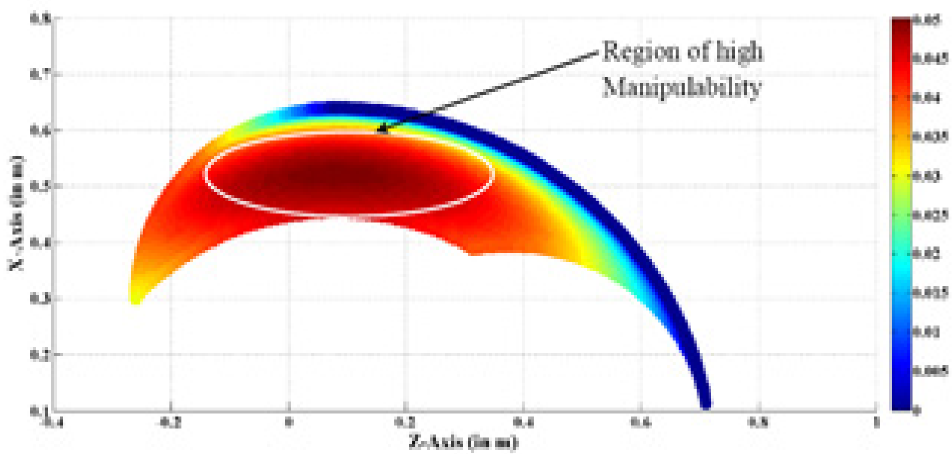

Figure 6. X-Z Projection of the Manipulability Index.

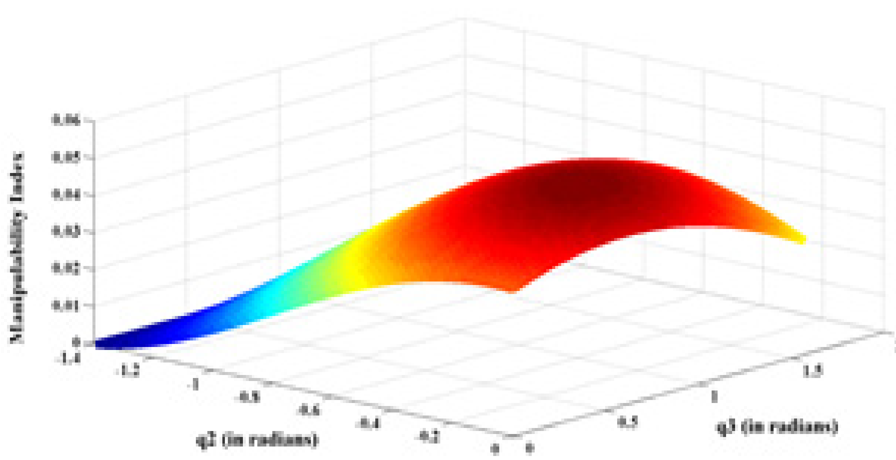

Figure 7. Manipulability Index vs. Joint Angles.

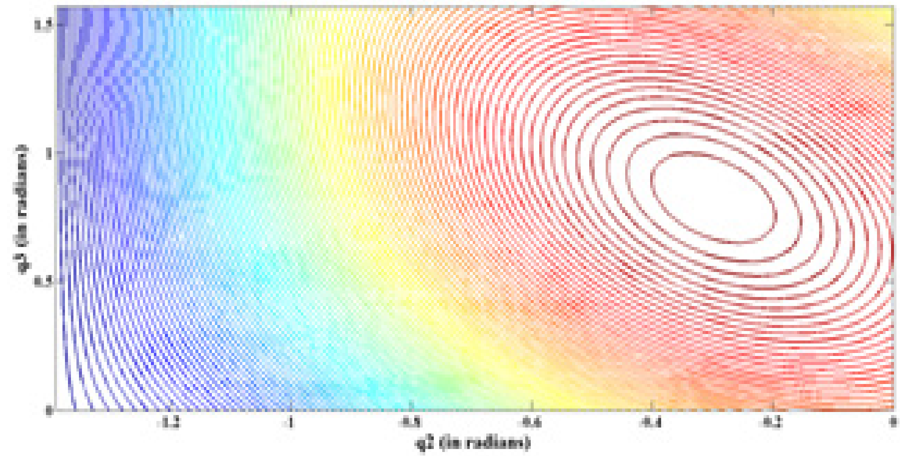

Figure 8. Level Curves of the Manipulability Index.

region of high manipulability is highlighted in the figure. From this region any straight line with a constant value for the x-axis would give the trajectory required. Figure-7 shows the manipulability index plotted against the joint variables q_1 and q_2. The position of the TCP frame is primarily governed by the joint angles $q_2$and q_3 for a constant coordinate of the y-axis. Figure-8 shows the level curves of the manipulability index. From Figure-7 and Figure-8, it is possible to conclude that the manipulability index, H is the maximum when q_2 is negative and q_3 is positive, and this should be ensured when the inverse kinematics solution for the trajectory planned is obtained.

7. Trajectory Generation

Trajectory refers to a time history of the position, velocity and acceleration for each degree of freedom. The basic problem to be addressed in any trajectory planning for a robotic arm, is to move the robot from the start position, given by the end-effector frame $\{\boldsymbol{E}_{initial}\}$, to the end position given by the frame $\{\boldsymbol{E}_{final}\}$. The specification of motion might include spatial constraints, such as the intermediate points or via points between the start and the end points. It may also include temporal constraints, such as the elapsed time between via points. Given these constraints the requirements for a trajectory generation algorithm may be to execute smooth motions, defined by continuous functions whose derivative is also continuous. Trajectories can be planned both in joint space and Cartesian space. In the joint space scheme, path shapes in space and time are described in terms of the functions of joint angles. The motion obtained from such a trajectory is known as a point-to-point motion. In Cartesian space schemes, path shapes in space and time are described in terms of task space coordinates. This type of motion is generally referred to as continuous path motion. In the current work, the trajectories are planned in Cartesian space, since the need for following a trajectory arises from the demands of image acquisition. For the purpose of image acquisition in the SFF-inspired algorithm, the end effector of the manipulator needs to track a straight line trajectory. It must be noted that the camera shares all the DoF of the robot, except the final roll motion of orientation. During this motion of the camera a stack of images from different focal points would be acquired. The start point, end points and via points must lie in the dexterous workspace of the robot arm so as to maintain the orientation along the path. This is required, because the selected trajectory must not be near the singular regions of the workspace. Such cases would demand very high joint angle velocities which may not be practically realizable. While selecting a line segment from the workspace, the requirement of a sufficient change in focus must also be considered. The trajectory tracked by the robot determines the camera pose for the set of images acquired. Hence, a precise planning of the trajectory is of high importance. The maximum size of the object of interest in the scene becomes a function of this trajectory, as the basic demand is the existence of the complete object in all the images of the stack. The problem of Cartesian straight line trajectory generation may be defined as follows:

Given the start point TCP frame say $\{A\}$ and the goal point TCP frame $\{B\}$ in the form specified below,

$$\{A\}=\begin{bmatrix} p_{x,A} \\ p_{y,A} \\ p_{z,A} \\ \alpha_A \\ \beta_A \\ \chi_A \end{bmatrix}=\begin{bmatrix} p_A \\ \Theta_A \end{bmatrix} \; and \; \{B\}=\begin{bmatrix} p_{x,B} \\ p_{y,B} \\ p_{z,B} \\ \alpha_B \\ \beta_B \\ \chi_B \end{bmatrix}=\begin{bmatrix} p_B \\ \Theta_B \end{bmatrix} \tag{30}$$

The two points must be connected by a straight line by linear interpolation between via points. The orientation of the end-effector must remain constant throughout the path of the trajectory, with reference to the world reference frame, $\{W\}$. This is because; any change in orientation would introduce perspective distortion in the images. The geometrical representation of the trajectory generation is shown in Figure-9.

The dotted lines indicate the trajectory, *p(t)* to be followed, and s_{PL} denotes the path length. In trajectory planning for robots, the position of points can be defined in the conventional manner. But when specifying the orientation as a rotation matrix at each via point, it is not possible to linearly interpolate its elements, as this would not result in a valid rotation matrix at all times. The so-called angle-axis representation can be used to specify an orientation with three numbers. This constitutes a 6 × 1 representation for defining each pose along the trajectory. Consider a via point pose $\{V\}$ specified, relative to the universal reference frame $\{W\}$, such that the position of the end-effector frame is given by ${}^{W}P_{Vorg}$, and the orientation given by a rotation matrix , which can be converted into an angle-axis representation, ${}^{W}K_{V}$. Thus, the pose of the end-effector for the particular via point is

$${}^{W}\chi_V=\begin{bmatrix} {}^{W}P_{Vorg} \\ {}^{W}K_V \end{bmatrix} \tag{31}$$

where ${}^{W}K_V$ is formed by scaling the unit vector ${}^{W}\hat{K}_V$ by the amount of rotation, θ_V required. If all via points are specified in this manner, a spline function which smoothly moves these six quantities from one via point to another as a function of time, would constitute the trajectory. When linear splines with parabolic blends are used, the shape of the trajectory will be linear. This would make sure that the linear and angular velocities are smoothly changed. Though smooth orientation changes are ensured, this method has one slight complication. The method does not guarantee that rotations occur about a single 'equivalent axis' in moving from point to point. This issue may be modeled as

$$\left({}^{W}\hat{K}_V,\theta_{WV}\right)=\left({}^{W}\hat{K}_V,\theta_{WV}+n360^{\circ}\right) \tag{32}$$

where n is any positive or negative integer. While going from a via point defined by the frame $\{A\}$ to a point defined by frame $\{B\}$ the difference in the representations of orientation $\left|{}^{W}K_B-{}^{W}K_A\right|$ should be minimum. For the simulation of trajectories, a constant velocity profile is adopted, though the literature suggests other profiles, such as trapezoids and sinusoids. Without loss of generality, a constant velocity profile eliminates the need for speed control, and the synchronization with the imaging device becomes simpler, when used in a real hardware. In the simulation environment the camera is mounted on the manipulator such that it shares 5-DoF of the manipulator starting from the first joint of the robot. The transformation defining the location of the camera with respect to the base frame is same as the transformation that defines the location of the last joint of the robot. Figure- 10 shows the robot with eye-in-hand camera in the simulation environment.

The SFF-Inspired algorithm requires a sequence of images acquired from different focal distances. The start and goal points of the trajectory are selected, based on the understanding of the workspace of the robot and the range of camera motion required. The trajectory is planned for a camera travel range of 180 mm, since for many trials that are not reported in this paper, higher values of camera travel range were considered. The position of the start point is arbitrarily defined by the origin of the frame $\{A\}$ = (0.5, 0, 0.18) and origin of the goal point frame defined by $\{B\}$ = (0.5, 0, 0). Figure-11 shows the desired variation of the position coordinates plotted with respect to time.

It may be observed from the coordinates, that the camera travels away from the measurement plane. The orientations of the TCP frame and the camera along all the points of the trajectory, are held constant, with the z-axis pointing towards the measurement plane. The simulation is carried out for 2 seconds sliced by 40 steps. Increasing the number of steps may improve the smoothness of the trajectory at the cost of increased execution time. This is because, in a Cartesian space trajectory planning, inverse kinematics is performed at run time. Figure-12 shows the variation of the set of joint angles, which causes the camera to follow a planned trajectory. It may be observed from the figure that the joint variables q_1, q_5 and q_6 are constants, since they do not contribute to any change in position and/or orientation, as defined by the frames attached to the points along the trajectory. Figure-13 shows the variation of the manipulability index along the path of the trajectory. The plot reveals that, though the trajectory is satisfactorily achieved, the manipulating ability of the robot is not the same everywhere.

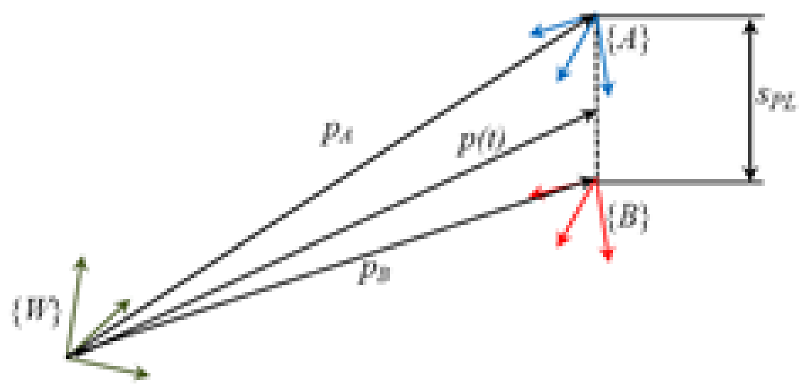

Figure 9. Linear Interpolation for Positioning of the Frame {E}.

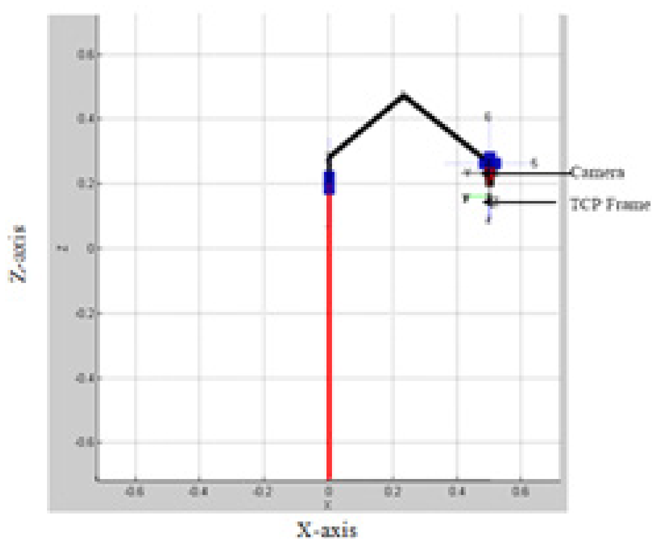

Figure 10. Eye-in-Hand Camera.

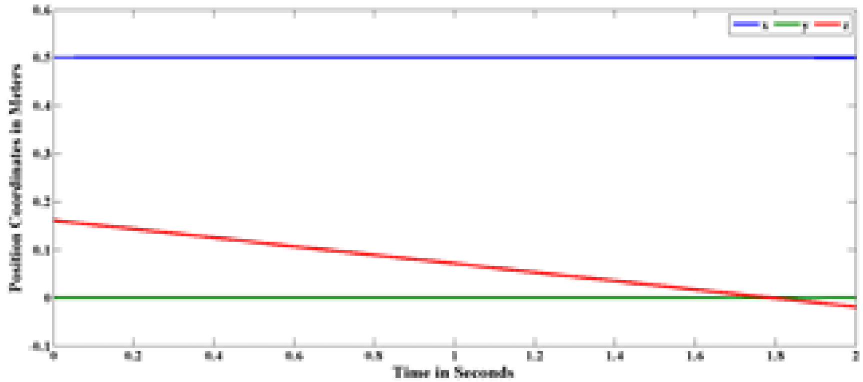

Figure 11. Position Variables of the Trajectory.

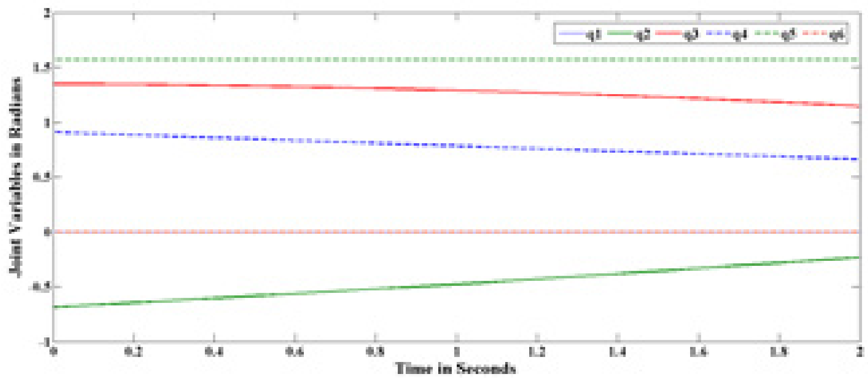

Figure 12. Joint Variables Conforming the Trajectory.

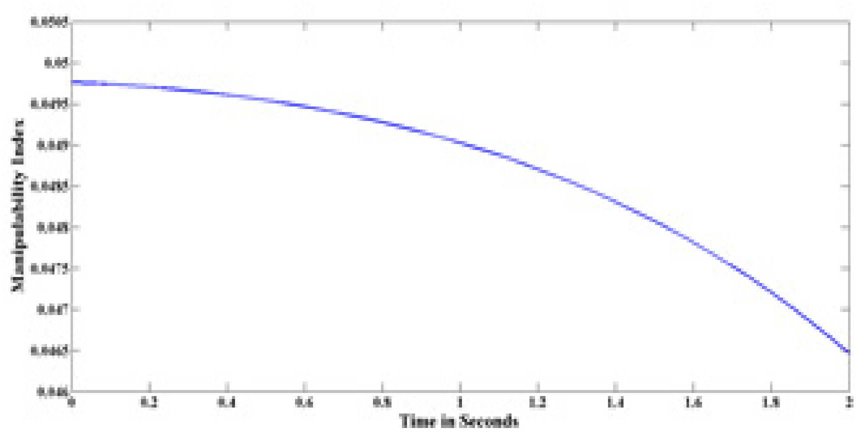

Figure 13. Manipulability Index along the Trajectory.

8. Conclusion

The paper presented a portion of the novel method of using an inspiration from conventional SFF to perform a sparse and coarse reconstruction of the scene which gives the benefit of using angle lenses resulting in large objects subjected to reconstruction. The SFF-inspired method desires a linear trajectory for the camera which requires a structured planning as per the need from the computer vision algorithm within the limitations of the manipulating ability of the manipulator. Such an approach is very common for all applications which involve Cartesian space trajectory planning but often bound by constraints arising only from the robotics domain. The trajectories planned and simulated were executed using a linear traversing mechanism where satisfactory results were obtained from the vision system. Though in a simulation environment, which deals only with robot kinematics, in a real world scenario the dynamics of the robot may be affected, when payloads of the manipulator are considered which is part of the future work to demonstrate the manipulation in the presence of dynamics involvement.

9. References

1. Corke PI. A robotics toolbox for MATLAB. IEEE Robotics and Automation Magazine. 1996; 3(1):24–32.
2. Sciavicco L, Siciliano B. Modelling and Control of Robot Manipulators. Springer–Verlag. 2002.
3. Craig JJ. Introduction to Robotics: Mechanics and Control, Mc– Graw Hill, 1989.
4. Tang CP. Manipulability–Based Analysis of Cooperative Payload Transport By Robot Collectives.Ph.D. Thesis, State University of New York, 2004.
5. Yang DC, Lai ZC. On the conditioning of robotics manipulations – service angle. ASME Journal of Mechanism, Transmission, and Automation in Design. 1985; 107:262–70.
6. Kumar AV, Waldron KJ. The workspace of a mechanical manipulator. ASME Journal of Mechanical Design. 1981; 103:665–72.
7. Nakamura Y. Advanced robotics: Redundancy and optimization.Addison–Wesley, 1991.
8. Yoshikawa T. Manipulability of robotic mechanisms. International Journal of Robotics Research. 1985; 4(2):3–9.