Robotica (2009) volume 27, pp. 425–445. © 2008 Cambridge University Press
doi:10.1017/S0263574708004839 Printed in the United Kingdom

Autonomous target capturing of free-floating space robot: Theory and experiments

Wenfu Xu†*, Bin Liang†, Cheng Li†, Yu Liu† and Yangsheng Xu‡

†*The Institute of Space Intelligent System, Harbin Institute of Technology, Harbin, P.R. China.*
‡*Department of Automation and Computer-Aided Engineering, The Chinese University of Hong Kong, Hong Kong, P.R. China.*

(Received in Final Form: June 5, 2008. First published online: July 17, 2008)

SUMMARY

Space robotic systems are expected to play an increasingly important role in the future. Unlike on the earth, space operations require the ability to work in the unstructured environment. Some autonomous behaviors are necessary to perform complex and difficult tasks in space. This level of autonomy relies not only on vision, force, torque, and tactile sensors, but also the advanced planning and decision capabilities. In this paper, the authors study the autonomous target capturing from the issues of theory and experiments. Firstly, we deduce the kinematic and dynamic equations of space robotic system. Secondly, the visual measurement model of hand–eye camera is created, and the image processing algorithms to extract the target features are introduced. Thirdly, we propose an autonomous trajectory planning method, directly using the 2D image features. The method predicts the target motion, plans the end-effector's velocities and solves the inverse kinematic equations using practical approach to avoid the dynamic singularities. At last, numeric simulation and experiment results are given. The ground experiment system is set up based on the concept of *dynamic simulation* and *kinematic equivalence*. With the system, the experiments of autonomous capturing a target by a free-floating space robot, composed of a 6-DOF manipulator and a satellite as its base, are conducted, and the results validate the proposed algorithm.

KEYWORDS: Space robot; Target capturing; Path planning; Visual information; On-orbital service.

1. Introduction

Robotic systems are expected to play an increasingly important role in future space activities. One broad area of application is in the servicing, construction, and maintenance of satellites and large space structures in orbit. Therefore, space robotic technologies have been emphasized by many countries.[1–3] Recently, the *Orbital Express* system, sponsored and led by the *Defense Advanced Research Projects Agency* (DARPA), validated on-orbit satellite servicing technologies.[4] The most remarkable mission was that a satellite autonomously used its robotic arm to rendezvous with and capture another satellite in space, paving the way for future space servicing operations.[5] Such technologies could lower costs and prolong legacy satellites flying for 5, 10, or even 15 extra years. The planning and control of space robot pose additional problems beyond those on earth, due to the dynamic coupling between space manipulators and their spacecraft, especially for free-floating system (Both the attitude and position of the base are not controlled), which exhibits nonholonomic behavior.[6,7] Carter and Cherchas studied the motion control of nonfixed base robotic manipulators.[8] and de Rivals-Mazères *et al.* considered the control and stabilization problem of flexible space robots.[9] Caccavale and Siciliano presented solution algorithms to the inverse kinematics of a space manipulator mounted on a free-floating spacecraft, and developed case studies for a system of a spacecraft with a six-joint manipulator attached.[10,11]

Capturing technology is one of the key technologies of space robot for on-orbital servicing. Unlike on the earth, space operations require the ability to work in the unstructured environment. Some autonomous behaviors are necessary to perform complex and difficult tasks in space. This level of autonomy relies not only on vision, force, torque, and tactile sensors, but also the advanced planning and control capabilities. Yoshida and Umetani developed on-line control scheme with vision feedback, which used generalized Jacobian matrix (GJM) concept for motion control and guaranteed workspace (GWS) for path planning.[12,13] Agrawal *et al.* proposed a systematic approach to analytically model the process of capture using the combination of simple motion primitives.[14] Papadopoulous *et al.* studied the dynamics and control of multi-arm space robot involved in chase and capture operations of satellites.[15] Nagamatsu *et al.* presented a capture strategy for retrieval of a tumbling satellite by a space manipulator.[16] Nagamatsu *et al.* also designed a control system for autonomous capture of a target satellite in space using predictive trajectory based on the target satellite dynamics.[17] Huang *et al.* developed a trajectory planning method of space manipulator that can track, approach, and catch the uncontrolled spinning satellite (USS) in free-floating situation.[18] McCourt and de Silva investigated the use of model-based predictive control for the capture of a multi-(DOF) object that moves in a somewhat arbitrary manner.[19]

In this paper, the authors consider the important problems about target capturing from the issues of theory and

* Corresponding author. E-mail: wfxu@robotsat.com

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

experiments. Firstly, the target may move with unknown trajectory, and the hand–eye camera needs to track the target, which means that the target motion must be estimated according to the measurement in real time. However, the output frequency of the visual measurement system is very low (the typical frequency is 2–4 Hz for on-orbital application), because the capability of the on-orbit processor is very limited. Secondly, the algorithm based on the differential kinematic equations will be affected by the singularities of the system Jacobian. The joint rates will become infinite and discontinuous when close to the singularities. Therefore, some algorithms are required to avoid the singularities, in order to guarantee that the target is captured successfully. Thirdly, in order to assure the space robot to implement the task successfully, ground experiments will be required to thoroughly explore the capabilities and limitations of the planning and control algorithms.

This paper focuses on the autonomous target capturing, from the theoretical modeling, numerical simulation, and experiment study. We deduce the kinematic and dynamic equations of space robotic system, and create the visual measurement model of hand–eye camera. Then, an autonomous trajectory planning method, directly using the 2D image features, is proposed. The planning approach has at least the following functions: (1) Predict the target motion in real time; and (2) avoid the kinematic and dynamic singularities. Furthermore, in order to verify the method, the authors combine the *dynamic emulation* with the *kinematic equivalence*, and set up an experiment system using commercial devices. The system can emulate the capturing process observed from the space base. By small modification, it can be extended to evaluate different technology of target capturing. The supplied 3D simulation system visualizes the capturing process intuitively.

The paper is organized as follows: Section 2 derives the kinematic and dynamic equations of space robot. Section 3 introduces the camera model and image processing algorithms to extract the target features. Then, the autonomous target capturing approach of space robot is proposed in Section 4. In Section 5, the numeric simulation and experiment study results are given. The last section deals with the discussion and conclusion of the work.

2. Modeling of Space Robotic System

2.1. Symbols and variables

Major research achievements on space robot were collected by Xu and Kanade,[20] and were also reviewed by Moosavian and Papadopoulos recently.[21] Here, we model the space robotic system using Lagrangian method. Figure 1 shows a general model of a space robot system, which is regarded as an $n+1$ serial link system connected with n active joints. B_0 denotes the satellite main body, B_i $(i=1,\ldots,n)$ denotes the ith link of the manipulator, and J_i is the joint which connects B_{i-1} and B_i.

In order to conveniently discuss, some symbols and variables are defined as follows (the following vectors are described in the inertia frame, if not pointed out specially):

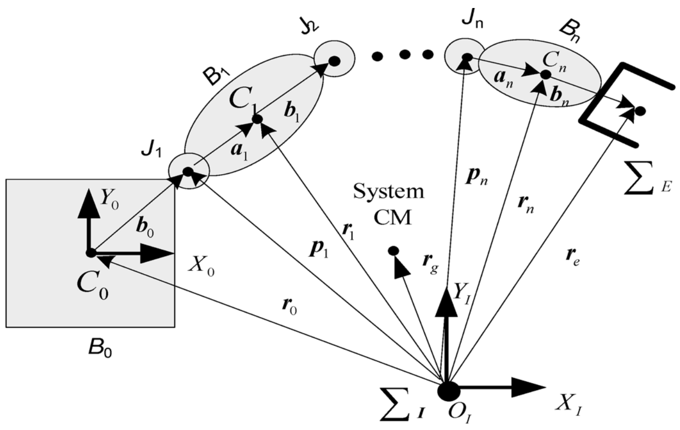

Fig. 1. General model of a single arm space robot.

$\sum_I, \sum_E, \sum_T$: the inertia frame, the end-effector frame, and the frame of the target's CM (center of mass), respectively;

$\sum_C, \sum_W, \sum_L$: the camera frame, the world frame and the light frame for the modeling of the camera;

$\sum_i$ $(i=0,\ldots,n)$: the body fixed frame of B_i;

C_i $(i=0,\ldots,n)$: the position of the CM of B_i;

$\boldsymbol{a}_i, \boldsymbol{b}_i \in \boldsymbol{R}^3$ $(i=0,\ldots,n)$: the position vectors from J_i to C_i and C_i to J_{i+1}, respectively;

$\boldsymbol{r}_i \in \boldsymbol{R}^3$ $(i=1,\ldots,n)$: the position vector of C_i;

$\boldsymbol{r}_g \in \boldsymbol{R}^3$: the position vector of the system's CM;

$\boldsymbol{p}_i \in \boldsymbol{R}^3$ $(i=1,\ldots,n)$: the position vector of J_i;

$\boldsymbol{p}_e \in \boldsymbol{R}^3$ $(i=1,\ldots,n)$: the position vector of the end-effector;

$\boldsymbol{k}_i$ $(i=1,\ldots,n)$: the unit vector representing the rotation direction of J_i;

$\psi_b, \psi_e \in \boldsymbol{R}^3$: the attitude angle of the base and the end-effector, expressed in terms of z–y–x Euler angles, i.e. $\psi_b=[\alpha_b,\beta_b,\gamma_b]^T$ and $\psi_e=[\alpha_e,\beta_e,\gamma_e]^T$.

${}^i\boldsymbol{A}_j \in \boldsymbol{R}^{3\times3}$: the coordinate transformation matrix of $\sum_j$ with respect to $\sum_i$. When $\sum_i$ is the inertia frame, the superscript i can be missed. The matrix ${}^i\boldsymbol{A}_j$ is described by $[{}^i\boldsymbol{n}_j, {}^i\boldsymbol{o}_j, {}^i\boldsymbol{a}_j]$;

${}^i\boldsymbol{T}_j \in \boldsymbol{R}^{4\times4}$: the homologous matrix of $\sum_j$ with respect to $\sum_i$. When $\sum_i$ is the inertia frame, the superscript i can be missed. The matrix ${}^i\boldsymbol{T}_j$ can be described by ${}^i\boldsymbol{T}_j=[{}^i\boldsymbol{n}_j, {}^i\boldsymbol{o}_j, {}^i\boldsymbol{a}_j, {}^i\boldsymbol{d}_j]$;

$\boldsymbol{X}_e \in \boldsymbol{R}^6$: the pose (position and attitude) of the end-effector, i.e. $\boldsymbol{X}_e=[\boldsymbol{p}_e^T, \boldsymbol{\Psi}_e^T]^T$;

$\boldsymbol{X}_t \in \boldsymbol{R}^6$: the pose of the target's CM, i.e. $\boldsymbol{X}_t=[\boldsymbol{p}_t^T, \boldsymbol{\Psi}_t^T]^T$;

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

${}^k r_{ij} \in \boldsymbol{R}^3$: the position vector from the origin of $\sum_i$ to that of $\sum_j$, expressed in $\sum_k$; if $\sum_k$ is the inertial frame, the superscript k can be missed. In addition if $\sum_i$ is the inertial frame, the symbol i can also be missed.

$\dot{\boldsymbol{x}}_b \in \boldsymbol{R}^6$: the linear velocity and angular velocity of B_0, i.e. $\dot{\boldsymbol{x}}_b = [\boldsymbol{v}_0^T, \boldsymbol{\omega}_0^T]^T$;

$\dot{\boldsymbol{x}}_e \in \boldsymbol{R}^6$: the linear velocity and angular velocity of the end-effector, i.e. $\dot{\boldsymbol{x}}_e = [\boldsymbol{v}_e^T, \boldsymbol{\omega}_e^T]^T$

$\Theta \in \boldsymbol{R}^n$: the actual joint angle vector, i.e. $\Theta = [\theta_1, \ldots, \theta_n]$;

$\Theta_d \in \boldsymbol{R}^n$: the desired joint angle vector, i.e. $\Theta_d = [\theta_{d1}, \ldots, \theta_{dn}]$;

$m_i\ (i = 0, \ldots, n)$: the mass of B_i;

w: total mass of the system, i.e. $w = \sum_{i=0}^n m_i$

$\boldsymbol{I}_i\ (i = 0, \ldots, n)$: the inertia matrix of B_i;

$\boldsymbol{E}$: 3×3 identity matrix;

2.2. *Kinematic and dynamic equations of free-floating robotic system*

From Fig. 1, the position of the end-effector is

$$\boldsymbol{p}_e = \boldsymbol{r}_0 + \boldsymbol{b}_0 + \sum_{k=1}^{n} (\boldsymbol{a}_k + \boldsymbol{b}_k). \qquad (1)$$

And the orientation of the end-effector is

$$\boldsymbol{A}_e = \boldsymbol{A}_0 \cdot {}^0\boldsymbol{A}_1 \cdot \cdots \cdot {}^{n-1}\boldsymbol{A}_n = \boldsymbol{A}_0 \cdot \boldsymbol{f}_m(\boldsymbol{\Theta}) = \boldsymbol{f}_s(\boldsymbol{\Psi}_b, \boldsymbol{\Theta}), \qquad (2)$$

where $\boldsymbol{f}_m(\Theta)$ is the ratio matrix from $\sum_1$ to $\sum_n$. It is function of the joint angles. And $\boldsymbol{f}_s(\boldsymbol{\Psi}_b, \boldsymbol{\Theta})$, determined by the base attitude and the joint angles, is the ratio matrix from $\sum_0$ to $\sum_n$.

Differentiating (1) with respect to time, a relationship between end-effector linear velocity and joint velocity is obtained, i.e.,

$$\boldsymbol{v}_e = \dot{\boldsymbol{p}}_e = \boldsymbol{v}_0 + \boldsymbol{\omega}_0 \times (\boldsymbol{p}_e - \boldsymbol{r}_0) + \sum_{k=1}^{n} [\boldsymbol{k}_k \times (\boldsymbol{p}_e - \boldsymbol{p}_k)] \dot{\theta}_k. \qquad (3)$$

On the other hand, a relationship between end-effector angular velocity and joint velocity is expressed with

$$\boldsymbol{\omega}_e = \boldsymbol{\omega}_0 + \sum_{k=1}^{n} \boldsymbol{k}_k \dot{\theta}_k. \qquad (4)$$

Then, the differential kinematic equation can be determined according to (3) and (4), i.e.,

$$\begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \boldsymbol{J}_b \begin{bmatrix} \boldsymbol{v}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} + \boldsymbol{J}_m \dot{\boldsymbol{\Theta}}, \qquad (5)$$

where $\boldsymbol{J}_b$ and $\boldsymbol{J}_m$ are the Jacobian matrixes dependent on the base and the manipulator, respectively,

$$\boldsymbol{J}_b = \begin{pmatrix} \boldsymbol{E} & -\tilde{\boldsymbol{p}}_{0e} \\ \boldsymbol{O} & \boldsymbol{E} \end{pmatrix} \in \boldsymbol{R}^{6\times6}, \quad \boldsymbol{p}_{0e} = \boldsymbol{p}_e - \boldsymbol{r}_0, \qquad (6)$$

$$\boldsymbol{J}_m = \begin{bmatrix} \boldsymbol{k}_1 \times (\boldsymbol{p}_e - \boldsymbol{p}_1) & \ldots & \boldsymbol{k}_n \times (\boldsymbol{p}_e - \boldsymbol{p}_n) \\ \boldsymbol{k}_1 & \ldots & \boldsymbol{k}_n \end{bmatrix} \in \boldsymbol{R}^{6\times n}. \qquad (7)$$

Operator $\tilde{\boldsymbol{r}}$ is the cross-product operator, i.e.,

$$\text{if } \boldsymbol{r} = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}, \quad \text{then } \tilde{\boldsymbol{r}} = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}. \qquad (8)$$

Since no external forces and torques act on the free-floating system, the linear and angular momentums are conserved. With the assumption that their initial values are zeros, the equations are

$$\boldsymbol{H}_b \begin{bmatrix} \boldsymbol{v}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} + H_{bm} \dot{\boldsymbol{\Theta}} = 0. \qquad (9)$$

The matrixes $\boldsymbol{H}_b$ and $\boldsymbol{H}_{bm}$ are the inertia matrix of the base and coupling inertia matrix, respectively. They are defined as

$$\boldsymbol{H}_b = \begin{pmatrix} w\boldsymbol{E} & w\tilde{\boldsymbol{r}}_{0g}^T \\ w\tilde{\boldsymbol{r}}_{bg} & \boldsymbol{H}_w \end{pmatrix} \in \boldsymbol{R}^{6\times6}, \qquad (10)$$

$$\boldsymbol{H}_{bm} = \begin{bmatrix} \boldsymbol{J}_{Tw} \\ \boldsymbol{H}_{w\phi} \end{bmatrix} \in \boldsymbol{R}^{6\times n}, \qquad (11)$$

$$\boldsymbol{H}_w = \sum_{i=1}^{n} \left(\boldsymbol{I}_i + m_i \cdot \tilde{\boldsymbol{r}}_{0i}^T \cdot \tilde{\boldsymbol{r}}_{0i}\right) + \boldsymbol{I}_0 \in \boldsymbol{R}^{3\times3}, \qquad (12)$$

$$\boldsymbol{J}_{Tw} = \sum_{i=1}^{n} (m_i \boldsymbol{J}_{Ti}) \in \boldsymbol{R}^{3\times n}, \qquad (13)$$

$$\boldsymbol{H}_{w\phi} = \sum_{i=1}^{n} (\boldsymbol{I}_i \boldsymbol{J}_{Ri} + m_i \tilde{\boldsymbol{r}}_{0i} \boldsymbol{J}_{Ti}) \in \boldsymbol{R}^{3\times n}, \qquad (14)$$

$$\boldsymbol{J}_{Ti} = [\boldsymbol{k}_1 \times (\boldsymbol{r}_i - \boldsymbol{p}_1), \boldsymbol{k}_2 \times (\boldsymbol{r}_i - \boldsymbol{p}_2), \ldots, \boldsymbol{k}_i \times (\boldsymbol{r}_i - \boldsymbol{p}_i), 0, \ldots, 0] \in \boldsymbol{R}^{3\times n}, \qquad (15)$$

$$\boldsymbol{J}_{Ri} = [\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_i, 0, \ldots, 0] \in \boldsymbol{R}^{3\times n}, \qquad (16)$$

$$\boldsymbol{r}_{0g} = \boldsymbol{r}_g - \boldsymbol{r}_0, \quad \text{and} \qquad (17)$$

$$\boldsymbol{r}_{0i} = \boldsymbol{r}_i - \boldsymbol{r}_0. \qquad (18)$$

From (9), $\boldsymbol{v}_0$ and $\boldsymbol{\omega}_0$ are solved as

$$\begin{bmatrix} \boldsymbol{v}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} = \dot{\boldsymbol{x}}_b = -\boldsymbol{H}_b^{-1} \boldsymbol{H}_{bm} \dot{\boldsymbol{\Theta}} = \boldsymbol{J}_{bm} \dot{\boldsymbol{\Theta}} = \begin{bmatrix} \boldsymbol{J}_{bm\_v} \\ \boldsymbol{J}_{bm\_\omega} \end{bmatrix} \dot{\boldsymbol{\Theta}}. \qquad (19)$$

The matrix $\boldsymbol{J}_{bm}$ is base-manipulator Jacobian, and $\boldsymbol{J}_{bm\_v}$ and $\boldsymbol{J}_{bm\_\omega}$ are its submatrixes. Substituting (19) into (5), the following relationship is given:

$$\begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = [\boldsymbol{J}_m + \boldsymbol{J}_b \boldsymbol{J}_{bm}]\dot{\boldsymbol{\Theta}} = \boldsymbol{J}_g(\boldsymbol{\Psi}_b, \boldsymbol{\Theta}, m_i, \boldsymbol{I}_i)\dot{\boldsymbol{\Theta}}. \qquad (20)$$

$\boldsymbol{J}_g(\boldsymbol{\Psi}_b, \boldsymbol{\Theta}, m_i, \boldsymbol{I}_i)$ is the GJM.[22] It is the function of the spacecraft attitude, joint angles, and mass properties of each body. If the GJM is of full rank, the joint rate can be calculated by $\dot{\boldsymbol{\Theta}} = \boldsymbol{J}_g^{-1}\dot{\boldsymbol{x}}_e$. Unfortunately, the singularities of $\boldsymbol{J}_g$ exist in most of the workspace. Since $\boldsymbol{J}_g$ is the function of the dynamic parameters of each body, its singularities are called *dynamic singularities*.[23] Different from the kinematic singularities (the singularities of traditional Jacobian are called *kinematic singularities* for the difference) of fixed-based manipulators, the dynamic singularities in Cartesian space are dependent on the path, and cannot be determined beforehand. These characteristics complicate the Cartesian path planning of space robot.

Moreover, the dynamic equation of space robot is derived from the Lagrange function, and is generally expressed in the following form[3]:

$$\begin{bmatrix} \boldsymbol{H}_b & \boldsymbol{H}_{bm} \\ \boldsymbol{H}_{bm}^T & \boldsymbol{H}_m \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_b \\ \ddot{\boldsymbol{\Theta}}_s \end{bmatrix} + \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_b \\ \boldsymbol{\tau}_m \end{bmatrix} \qquad (21)$$

where

$\boldsymbol{H}_m \in \boldsymbol{R}^{n\times n}$: inertia matrix of the manipulator arm;
$\boldsymbol{c}_b \in \boldsymbol{R}^6$: velocity-dependent nonlinear term for the base;
$\boldsymbol{c}_m \in \boldsymbol{R}^6$: velocity dependent nonlinear term for the arm;
$\boldsymbol{F}_b \in \boldsymbol{R}^6$: force and torque on the centroid of the base;
$\boldsymbol{\tau}_m \in \boldsymbol{R}^n$: torque on the manipulator joints.

3. Camera Projection Model and Image Processing

In this paper, it is assumed to capture a cooperative target, i.e. some white lighting markers with a known geometrical arrangement are mounted on the target.

3.1. Image frames

The image plane corresponds to CCD discrete array of photosensing elements. Associated with it is the 2D image frame $\{x, y\}$. A point P in the image plane is denoted by (x, y). The image of the scene on the CCD is digitalized, transferred to the computer memory via the image capture. We define the 2D image frame $\{u, v\}$, and (u_0, v_0) stands for the image center. The coordinates of point P on the image plane are denoted by (u, v) in pixels. The relationship between (x, y) and (u, v) is

$$\begin{cases} u = x\alpha_x + u_0 \\ v = y\alpha_y + v_0 \end{cases}, \qquad (22)$$

where α_x is the horizontal pixel pitch of the sensor (pixels/m) and α_y is the vertical pixel pitch of the sensor (pixels/m).

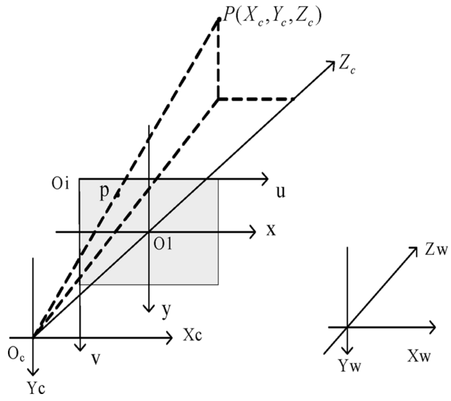

Fig. 2. Pinhole model of the camera.

3.2. Pinhole model

A special case of the projective camera is the perspective (or central) projection, reducing to the familiar pinhole camera model, which is shown in Fig. 2.

The origin of the camera frame (i.e. point O_c) is on the optical axis and in the center of the lens. The axes X and Y are aligned with x and y, respectively. They define a plane perpendicular to the optical axis; thus the axis Z is aligned with the optical axis.

Assume that a point P, whose coordinates with respect to $\sum_C$ are denoted by ${}^c\boldsymbol{P} = [X_c, Y_c, Z_c]$, will project on the image plane with coordinates (x, y), given by (O_cO_1 equals the focus length f)

$$\begin{cases} x = \dfrac{fX_C}{Z_C} \\ y = \dfrac{fY_C}{Z_C} \end{cases}. \qquad (23)$$

Substituting (22) into (23), the result is

$$\begin{cases} u = f\alpha_x \dfrac{X_c}{Z_c} + u_0 = f_u \dfrac{X_c}{Z_c} + u_0 \\ v = f\alpha_y \dfrac{Y_c}{Z_c} + v_0 = f_v \dfrac{Y_c}{Z_c} + v_0 \end{cases}, \qquad (24)$$

where

$$\begin{cases} f_u = f\alpha_x \\ f_v = f\alpha_y \end{cases}. \qquad (25)$$

When the homologous coordinates are used, Eq. (24) can be written in the following form:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_C} \begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix}. \qquad (26)$$

If the coordinate of point P in the world frame is $[X_w, Y_w, Z_w]^T$, and the homologous transformation matrix from

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

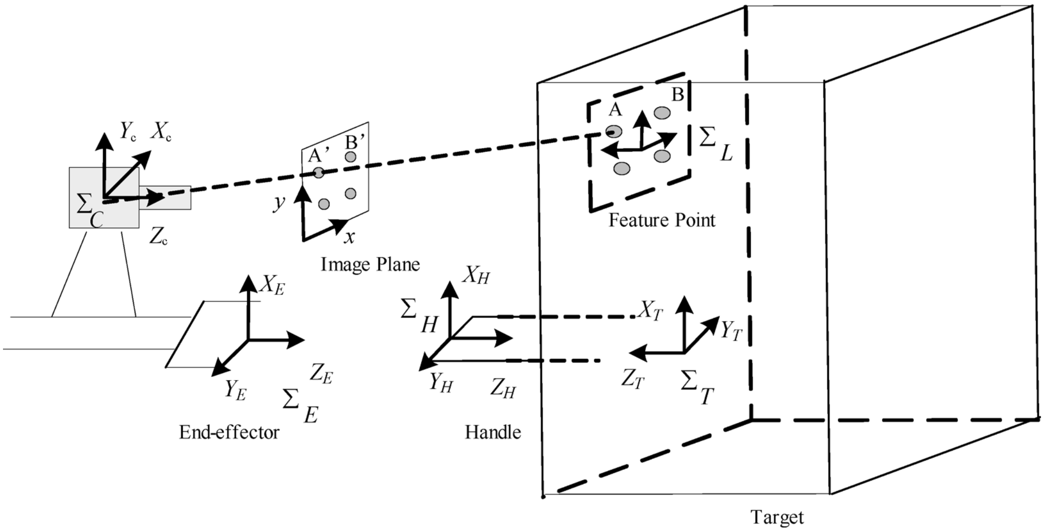

Fig. 3. Frames used for measurement.

the camera frame to the world frame is ${}^{C}\boldsymbol{T}_W$, the following relationship is given:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_C}\begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} {}^{C}\boldsymbol{T}_W \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}, \qquad (27)$$

where, f_u, f_v, u_0, and v_0 are the intrinsic parameters and ${}^{C}\boldsymbol{T}_W$ is the extrinsic parameters matrix.

3.3. Vision frames

The vision frames are defined as shown in Fig. 3. $\sum_E$ is the end-effector frame and $\sum_H$ is the handle frame. When the position and orientation of $\sum_H$ with respect to $\sum_E$ are close to zero, the end-effector can grasp the handle. That is to say, the aim to activate the manipulator is to make the relative pose (position and orientation) smaller and smaller. Frames $\sum_L$ and $\sum_C$ are the light frame (four lighting markers are fixed on the target for the visual measure; their reference frame is named light frame) and the target and camera frame, respectively. And the coordinates of the feature points with respect to $\sum_L$ are known, as *a priori* information.

3.4. Image processing to extract the target features

3.4.1. Image smoothing—Median filter. The acquired images must pass through a stage of image preprocessing in order to remove distracting and useless information from the images. For example, the existence of impulsive noise in the images is one of the most habitual problems. The image smoothing algorithms are applied in order to reduce noise and/or to prepare images for further processing such as segmentation. They can be classified into linear and nonlinear algorithms where the former are amenable to analysis in the Fourier domain and the latter are not.

Median filter is a nonlinear filter more used to remove the impulsive noise from an image. It is a more robust method than the traditional linear filtering, because it preserves the sharp edges. The detailed algorithm was given in ref. [24].

3.4.2. Image segmentation—Region growing approach. Image segmentation is one of the most important steps leading to the analysis of processed image data. Its main goal is to divide an image into parts that have a strong correlation with objects or areas of the real world contained in the image. Segmentation methods can be divided into three groups according to the dominant features they employ: First is global knowledge about an image or its part; the knowledge is usually represented by a histogram of image features. Edge-based segmentations form the second group and region-based segmentations the third. Region growing techniques are generally better in noisy images, where borders are extremely difficult to detect. Homogeneity is an important property of regions and is used as the main segmentation criterion in region growing, whose basic idea is to divide an image into zones of maximum homogeneity.[25] We use the region growing method to segment the images smoothed by the median filter. A processed result is shown in Fig. 4.

3.4.3. Feature extraction. After segmentation, each region is labeled. One method of region labeling involves examining

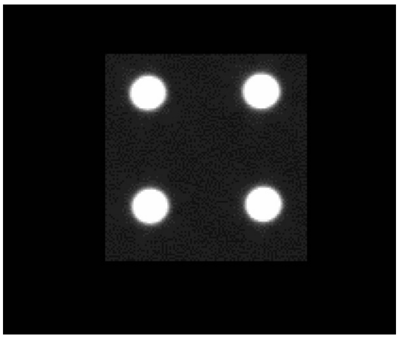

Fig. 4. The image after segmentation.

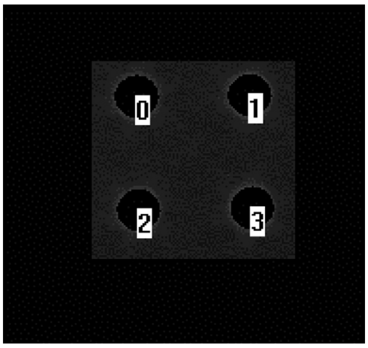

Fig. 5. The region labeling results.

each pixel in a mapping and comparing its value to that of its neighbors. If its value is "close enough" (as determined by some distance function) then it is determined as being in the same region as that neighbor. An equivalence table is maintained for a second pass that merges any regions determined to be the same. As in the convolution operation, this method of region labeling is directly extendible to high-dimensional data by simply checking all N-dimensional neighbors. The region labeling results are shown in Fig. 5.

When the lighting patterns (the circular area) are recognized and labeled, we can further extract their centroids as follows:

$$\begin{cases} u_{ci} = \dfrac{\sum_{k=1}^{N_i} u_{nk} I_k}{\sum_{k=1}^{N_i} I_k} \\ v_{ci} = \dfrac{\sum_{k=1}^{N_i} v_{nk} I_k}{\sum_{k=1}^{N_i} I_k} \end{cases}, \quad i = 0, \ldots, 3, \tag{28}$$

where u_{ci} and v_{ci} are the image coordinate of the ith circular area, u_{nk} is the image coordinate of kth point contained in the circular area (each point is denoted by a single index k), I_k is the corresponding grey value, and N_i is the number of the pixels contained in the ith circular area.

Since the feature patterns are fixed on the target, the target tracking task can be realized by tracking the centroids of the circular area. We named these centroids as feature points. For simplification, u_{ci} and v_{ci} are replaced by u_i and v_i in the following parts, and the target feature is defined as

$$\boldsymbol{f} = [u_1, v_1, \ldots, u_4, v_4]^T. \tag{29}$$

4. Trajectory Planning for Autonomous Target Capturing

4.1. Main procedure

The autonomous planning method, partly reported in the conference IROS'06,[26] is proposed to generate the trajectory of space robot to capture the target in space.

The main steps are as follows:

(1) Initialization, i.e. setting initial time ($t = 0$), determining the desired features $\boldsymbol{f}_d = [u_{d1}, v_{d1}, \ldots, u_{dN}, v_{dN}]^T$ (N is the number of the feature points, $N = 4$) and defining the stopping criteria (allowed feature error $\varepsilon = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{2N}]^T$ and the maximal time $t_{\max}$);

(2) Extract the image features of the patterns on the target, $\boldsymbol{f} = [u_1, v_1, \ldots, u_N, v_N]^T$;

(3) Calculate the feature errors $\Delta\xi = [u_{d1} - u_1,\ v_{d1} - v_1, \ldots,\ u_{dN} - u_N,\ v_{dN} - v_N]^T$ and judge whether the target lies within the capturing box (i.e. the grasp area, $\|\Delta\xi_i\| \le \varepsilon_i, i = 1, \ldots, 2N$). If so, the manipulator closes its gripper and grasps the target; else, go to step 4;

(4) Predict the target motion according to the measured values;

(5) Read the current states (i.e. $\boldsymbol{\Psi}_b, \boldsymbol{\omega}_0, \boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}$) of the space robotic system from the corresponding sensors;

(6) Plan the end-effector velocities, which drive the end-effector to track and approach the target along the closest path (i.e. straight lines), using the measured and estimated results;

(7) The singularity avoiding algorithm is called to determine the desired joint rates, i.e. $\dot{\boldsymbol{\Theta}}_d$;

(8) The desired joint angles $\boldsymbol{\Theta}_d$ are defined by integrating $\dot{\boldsymbol{\Theta}}_d$;

(9) The joint controllers generate driving torques of the joints to follow $\boldsymbol{\Theta}_d$ and $\dot{\boldsymbol{\Theta}}_d$;

(10) $t = t + \Delta t$. If $t < t_{\max}$, go to step 2; else, the algorithm stops, meaning that the space robot cannot capture the target in the prescribed time.

The main procedure is shown in Fig. 6.

4.2. The relationship between the image feature and the end-effector pose

The relationship between the image features and the end-effector pose can be described by image Jacobian matrix, which is a linear transformation from the image feature space to the task space.

According to (24), the following relationship exists:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} -\dfrac{f_u}{Z_c} & 0 & \dfrac{u - u_0}{Z_c} & \dfrac{(u-u_0)(v-v_0)}{f_v} & -\dfrac{f_u^2 + (u-u_0)^2}{f_u} & \dfrac{f_u(v-v_0)}{f_v} \\ 0 & -\dfrac{f_v}{Z_c} & \dfrac{v - v_0}{Z_c} & \dfrac{f_v^2 + (v-v_0)^2}{f_v} & -\dfrac{(u-u_0)(v-v_0)}{f_u} & -\dfrac{f_v(u-u_0)}{f_u} \end{bmatrix} \begin{bmatrix} {}^c\boldsymbol{v}_c \\ {}^c\boldsymbol{\omega}_c \end{bmatrix}. \tag{30}$$

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

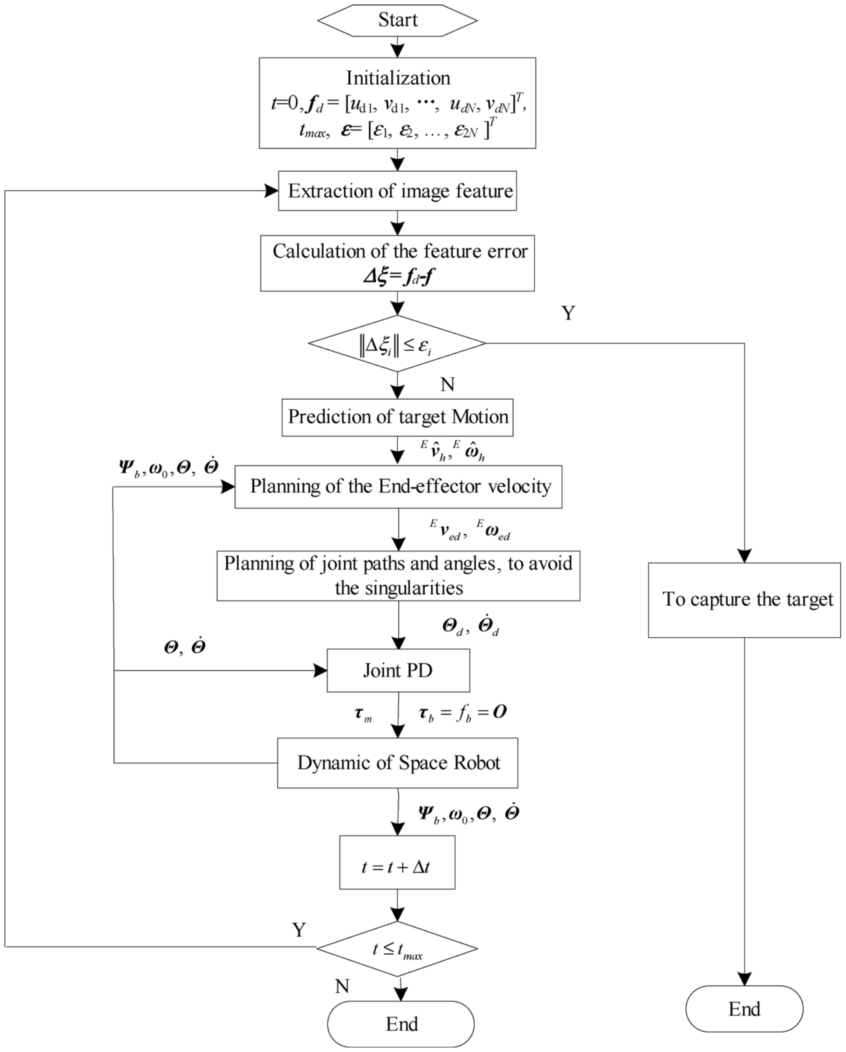
Start
Initialization
t=0, $\boldsymbol{f}_d = [u_{d1}, v_{d1}, \cdots, u_{dN}, v_{dN}]^T$,
t_{max}, $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{2N}]^T$
Extraction of image feature
Calculation of the feature error
$\Delta\xi = f_d - f$
$\|\Delta\xi_i\| \leq \varepsilon_i$
Y
N
Prediction of target Motion
${}^E\hat{v}_h$, ${}^E\hat{\omega}_h$
$\Psi_b, \omega_0, \Theta, \dot{\Theta}$
Planning of the End-effector velocity
${}^E v_{ed}$, ${}^E\omega_{ed}$
Planning of joint paths and angles, to avoid the singularities
Θ_d, $\dot{\Theta}_d$
Θ, $\dot{\Theta}$
Joint PD
τ_m
$\tau_b = f_b = O$
Dynamic of Space Robot
$\Psi_b, \omega_0, \Theta, \dot{\Theta}$
$t = t + \Delta t$
Y
$t \leq t_{max}$
N
End
To capture the target
End

Fig. 6. Flowchart of the path planning for autonomous capturing.

where, ${}^c\boldsymbol{v}_c$ and ${}^c\boldsymbol{\omega}_c$ are the linear and angular velocities of the camera. Then the image Jacobian matrix is defined as

$$\boldsymbol{J}_{\text{image}}(u, v, Z_c) = \begin{bmatrix} -\dfrac{f_u}{Z_c} & 0 & \dfrac{u-u_0}{Z_c} & \dfrac{(u-u_0)(v-v_0)}{f_v} & -\dfrac{f_u^2+(u-u_0)^2}{f_u} & \dfrac{f_u(v-v_0)}{f_v} \\ 0 & -\dfrac{f_v}{Z_c} & \dfrac{v-v_0}{Z_c} & \dfrac{f_v^2+(v-v_0)^2}{f_v} & -\dfrac{(u-u_0)(v-v_0)}{f_u} & -\dfrac{f_v(u-u_0)}{f_u} \end{bmatrix}. \quad (31)$$

When n features are used, the image feature vector and corresponding depth vector are denoted by $\xi = [u_1, v_1, \ldots, u_n, v_n]$ and $\boldsymbol{Z}_c = [Z_{c1}, Z_{c2}, \ldots Z_{cn}]$, respectively. Then the extended image Jacobian matrix is defined by

$$\boldsymbol{J}_{\text{image}}(\xi, \boldsymbol{Z}_c) = \begin{bmatrix} \boldsymbol{J}_{\text{image}}(u_1, v_1, Z_{c1}) \\ \vdots \\ \boldsymbol{J}_{\text{image}}(u_n, v_n, Z_{cn}) \end{bmatrix} \in \boldsymbol{R}^{2n\times 6}. \quad (32)$$

Correspondingly,

$$\dot{\boldsymbol{\xi}} = [\dot{u}_1, \dot{v}_1, \ldots, \dot{u}_n, \dot{v}_n]^T = \boldsymbol{J}_{\text{image}}(\boldsymbol{\xi}, \boldsymbol{Z}_c) \begin{bmatrix} {}^c\boldsymbol{v}_c \\ {}^c\boldsymbol{\omega}_c \end{bmatrix}. \quad (33)$$

Equation (33) establishes the mapping from the camera velocities to the variation rates of the image features. For the control of 6-DOF manipulator, the rank of $\boldsymbol{J}_{\text{image}}(\boldsymbol{\xi}, \boldsymbol{Z}_c)$ is required not to be less than 6, since there are six joint rates to be determined. That is to say, the planning method needs at least three features. If more than three features are used, the pseudoinverse of the image Jacobian matrix can improve the accuracy. Here, N feature points ($N \geq 3$) are utilized. If $\boldsymbol{J}_{\text{image}}(\boldsymbol{\xi}, \boldsymbol{Z}_c)$ is nonsingular, the velocities of the camera can be determined by Eq. (33).

4.3. Prediction of the target motion

4.3.1. The nonlinear relationship between image feature and relative pose. The target motion is predicted by the 2D image

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

features. From Fig. 3, the relationship between the frames is

$$^{E}\boldsymbol{T}_{C} \cdot {}^{C}\boldsymbol{T}_{L} = {}^{E}\boldsymbol{T}_{H} \cdot {}^{H}\boldsymbol{T}_{L}. \tag{34}$$

And the matrix $^{E}\boldsymbol{T}_{H}$ is the homologous matrix and is of the following form:

$$^{E}\boldsymbol{T}_{H} = \begin{bmatrix} ^{E}\boldsymbol{A}_{H} & ^{E}\boldsymbol{r}_{eh} \\ \boldsymbol{O}_{3\times 3} & 1 \end{bmatrix}. \tag{35}$$

where, $^{E}\boldsymbol{r}_{eh}$ and $\boldsymbol{\Psi}_{eh}$ are relative position and orientation, which are described by

$$^{E}\boldsymbol{r}_{eh} = [X, Y, Z]^{T}, \quad \boldsymbol{\Psi}_{eh} = [\alpha, \beta, \gamma]^{T}. \tag{36}$$

From Eq. (34), the following result can be obtained:

$$^{C}\boldsymbol{T}_{L} = \left(^{E}\boldsymbol{T}_{C}\right)^{-1} \cdot {}^{E}\boldsymbol{T}_{H} \cdot {}^{H}\boldsymbol{T}_{L} = \begin{bmatrix} ^{C}\boldsymbol{A}_{L} & ^{C}\boldsymbol{r}_{cL} \\ \boldsymbol{O}_{3\times 3} & 1 \end{bmatrix}, \tag{37}$$

where the matrixes $^{E}\boldsymbol{T}_{C}$ and $^{H}\boldsymbol{T}_{L}$ are constants and can be calibrated beforehand. And $^{C}\boldsymbol{r}_{cL}$ is the vector from the origin of $\sum_{C}$ to that of $\sum_{O}$, and expressed in $\sum_{C}$. The relative attitude $\boldsymbol{\Psi}_{cL} = [\alpha_{cL}\ \beta_{cL}, \gamma_{cL}]$ is represented by z–y–x Euler angles. The coordinate transformation matrix $^{C}\boldsymbol{A}_{L}$ is determined by $\boldsymbol{\Psi}_{cL}$.

The position vector of ith feature point is described in $\sum_{C}$ as

$$^{C}\boldsymbol{P}_{i} = {}^{C}\boldsymbol{r}_{co} + {}^{C}\boldsymbol{A}_{L} \cdot {}^{L}\boldsymbol{P}_{i}, \tag{38}$$

where $^{L}\boldsymbol{P}_{i} = [X_{li}, Y_{li}, Z_{li}]$ is the position of ith feature point in $\sum_{L}$, and $^{C}\boldsymbol{P}_{i} = [X_{ci}, Y_{ci}, Z_{ci}]$ is in $\sum_{C}$. From the pinhole model, the image features of ith feature point are

$$\begin{cases} u_{i} = f_{u}\dfrac{X_{ci}}{Z_{ci}} + u_{0} \\ v_{i} = f_{v}\dfrac{Y_{ci}}{Z_{ci}} + v_{0} \end{cases}. \tag{39}$$

Then the relationship between $^{E}\boldsymbol{T}_{H}$ and $[u_{i},\ v_{i}]$ can be determined according to (37)–(39), which is written in the following form:

$$\begin{cases} u_{i} = g_{u_{i}}\left(^{E}\boldsymbol{r}_{eh}, \boldsymbol{\Psi}_{eh}\right) = g_{u_{i}}(X, Y, Z, \alpha, \beta, \gamma) \\ v_{i} = g_{v_{i}}\left(^{E}\boldsymbol{r}_{eh}, \boldsymbol{\Psi}_{eh}\right) = g_{v_{i}}(X, Y, Z, \alpha, \beta, \gamma) \end{cases}. \tag{40}$$

4.3.2. Prediction of target motion based on extended Kalman filter. Target in space may move in various modes, such as free-floating, tumbling, and so on. That means the target moves with unknown trajectory. And the observer (hand–eye camera) moves too. Therefore, the target motion must be estimated in real time. Kalman filter plays the important role in solving this problem.

The system state vector is defined as

$$\boldsymbol{W} = [X, \dot{X}, Y, \dot{Y}, Z, \dot{Z}, \alpha, \dot{\alpha}, \beta, \dot{\beta}, \gamma, \dot{\gamma}]^{T}. \tag{41}$$

The system model and output model are

$$\boldsymbol{W}_{k} = \boldsymbol{A}\boldsymbol{W}_{k-1} + \boldsymbol{\delta}_{k}, \tag{42}$$

$$\boldsymbol{Z}_{k} = \boldsymbol{G}(\boldsymbol{W}_{k}) + \boldsymbol{v}_{k}, \tag{43}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 & T & \dots & \dots & \dots \\ 0 & 1 & \dots & \dots & \dots \\ \vdots & \vdots & \ddots & \dots & \dots \\ \vdots & \vdots & \dots & 1 & T \\ \vdots & \vdots & \dots & 0 & 1 \end{bmatrix}_{12\times 12}, \tag{44}$$

$$\boldsymbol{Z}_{k} = [u_{1}, v_{1}, \dots, u_{N}, v_{N}]_{k}^{T}, \tag{45}$$

$$\boldsymbol{G}(\boldsymbol{W}_{k}) = [g_{u_{1}}, g_{v_{1}}, \dots, g_{u_{N}}, g_{vN}]_{k}^{T}, \tag{46}$$

δ_{k} and υ_{k} are the disturbance noise and measurement noise, respectively, which are assumed to be a zero mean Gaussian noise with covariance of $\boldsymbol{R}$. And, T is the sample period ($T = 0.25$ s). Since the output model is nonlinear, the extended Kalman filter is used to provide the optimal estimate of the system state.

4.4. Planning of the end-effector

The velocities of the camera are planned according to (33), i.e.

$$\begin{bmatrix} ^{C}\boldsymbol{v}_{c} \\ ^{C}\boldsymbol{\omega}_{c} \end{bmatrix} = \boldsymbol{K}[\boldsymbol{J}_{\text{image}}(\boldsymbol{\xi}, \boldsymbol{Z})]^{\#}\Delta\xi, \tag{47}$$

where $\boldsymbol{K}$ is the gain matrix, $[\boldsymbol{J}_{\text{image}}(\boldsymbol{\xi}, \boldsymbol{Z})]^{\#}$ is the pseudoinverse of $\boldsymbol{J}_{\text{image}}(\boldsymbol{\xi}, \boldsymbol{Z})$ defined by $\boldsymbol{J}^{\#} = (\boldsymbol{J}^{T}\boldsymbol{J})^{-1}\boldsymbol{J}^{T}$, and $\Delta\xi$ is the difference of the image features.

$$\begin{aligned} \Delta\xi &= \boldsymbol{f}_{d} - \boldsymbol{f} \\ &= [u_{d1} - u_{1}, v_{d1} - v_{1}, \dots, u_{dN} - u_{N}, v_{dN} - v_{N}]^{T}, \end{aligned} \tag{48}$$

$\boldsymbol{f}_{d}$ and $\boldsymbol{f}$ are the desired and actual image feature vectors, respectively,

$$\boldsymbol{f}_{d} = [u_{dl},\ v_{dl}, \dots, u_{dN},\ v_{dN}]^{T}, \text{ and} \tag{49}$$

$$\boldsymbol{f} = [u_{1},\ v_{1}, \dots, u_{N},\ v_{N}]^{T}. \tag{50}$$

Then, the end-effector velocities are determined according to the fixed relationship between the frame $\sum_{E}$ and the frame $\sum_{C}$. And the estimated target motion should be included in the end-effector velocities too, i.e.

$$\begin{bmatrix} ^{E}\boldsymbol{v}_{e} \\ ^{E}\boldsymbol{\omega}_{e} \end{bmatrix} = \begin{bmatrix} ^{E}\boldsymbol{A}_{C} & -{}^{E}\boldsymbol{A}_{C} \cdot {}^{C}\tilde{\boldsymbol{r}}_{ce} \\ \boldsymbol{O} & ^{E}\boldsymbol{A}_{C} \end{bmatrix} \begin{bmatrix} ^{C}\boldsymbol{v}_{c} \\ ^{C}\boldsymbol{\omega}_{c} \end{bmatrix} + \begin{bmatrix} ^{E}\boldsymbol{v}_{h} \\ ^{E}\boldsymbol{\omega}_{h} \end{bmatrix}. \tag{51}$$

According to the law of relative motion,

$$^{E}\boldsymbol{v}_h = {}^{E}\boldsymbol{v}_h^e + {}^{E}\boldsymbol{v}_e + {}^{E}\boldsymbol{\omega}_e \times {}^{E}\boldsymbol{r}_{eh}, \tag{52}$$

$$^{E}\boldsymbol{\omega}_h = {}^{E}\boldsymbol{\omega}_h^e + {}^{E}\boldsymbol{\omega}_e, \tag{53}$$

where, $^{E}\boldsymbol{v}_h^e$ is the linear velocity of the handle (mounted on the target satellite) with respect to the end-effector, which is expressed in $\sum_E$, i.e.

$$^{E}\boldsymbol{v}_h^e = [\dot{X}, \dot{Y}, \dot{Z}]^T, \tag{54}$$

$$^{E}\boldsymbol{\omega}_h^e = \begin{bmatrix} 0 & -s_\alpha & c_\alpha c_\beta \\ 0 & c_\alpha & s_\alpha c_\beta \\ 1 & 0 & -s_\beta \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}. \tag{55}$$

From (52) and (53), the target motion can be calculated as

$$\begin{bmatrix} ^{E}\boldsymbol{v}_h \\ ^{E}\boldsymbol{\omega}_h \end{bmatrix} = \begin{bmatrix} ^{E}\boldsymbol{v}_h^e \\ ^{E}\boldsymbol{\omega}_h^e \end{bmatrix} + \begin{bmatrix} \boldsymbol{I}_{3\times3} & -{}^{E}\boldsymbol{r}_{eh} \\ \boldsymbol{O}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \begin{bmatrix} ^{E}\boldsymbol{v}_e \\ ^{E}\boldsymbol{\omega}_e \end{bmatrix}. \tag{56}$$

When the vectors are expressed in $\sum_E$, the differential kinematic equations are

$$\begin{bmatrix} ^{E}\boldsymbol{v}_e \\ ^{E}\boldsymbol{\omega}_e \end{bmatrix} = ({}^{E}\boldsymbol{J}_g)\dot{\boldsymbol{\Theta}}. \tag{57}$$

Substituting (57) into (56), the following relationship is given:

$$\begin{bmatrix} ^{E}\boldsymbol{v}_h \\ ^{E}\boldsymbol{\omega}_h \end{bmatrix} = \begin{bmatrix} ^{E}\boldsymbol{v}_h^e \\ ^{E}\boldsymbol{\omega}_h^e \end{bmatrix} + \begin{bmatrix} \boldsymbol{I}_{3\times3} & -{}^{E}\boldsymbol{r}_{eh} \\ \boldsymbol{O}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} ({}^{E}\boldsymbol{J}_g)\dot{\boldsymbol{\Theta}}. \tag{58}$$

At last, the end-effector velocities are determined by the following equation:

$$\begin{bmatrix} ^{E}\boldsymbol{v}_{ed} \\ ^{E}\boldsymbol{\omega}_{ed} \end{bmatrix} = \begin{bmatrix} ^{E}\boldsymbol{A}_C & -{}^{E}\boldsymbol{A}_C \cdot {}^{C}\tilde{\boldsymbol{r}}_{ce} \\ \boldsymbol{O} & ^{E}\boldsymbol{A}_C \end{bmatrix} \boldsymbol{K}[\boldsymbol{J}_{\text{image}}(\boldsymbol{\xi}, \boldsymbol{Z})]^{\#}\Delta\xi$$
$$+ \begin{bmatrix} ^{E}\boldsymbol{v}_h^e \\ ^{E}\boldsymbol{\omega}_h^e \end{bmatrix} + \begin{bmatrix} \boldsymbol{I}_{3\times3} & -{}^{E}\boldsymbol{r}_{eh} \\ \boldsymbol{O}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} ({}^{E}\boldsymbol{J}_g)\dot{\boldsymbol{\Theta}}. \tag{59}$$

In practice, the end-effector velocity is not allowed to be large, because the base, which is free floating according to the reaction of the manipulator, will lean much with large disturbance on it. Therefore, it is required to be careful when choosing the gain matrix $\boldsymbol{K}$, and to set the upper bound of the end-effector velocity (the bounds of the linear and angular velocity are 40 mm/s and 5 deg/s, respectively).

4.5. The practical approach to avoid the dynamic singularities

If the GJM $^{E}\boldsymbol{J}_g$ is of full rank, the desired joint rate can be determined after the end-effector velocities are planned, i.e.

$$\dot{\boldsymbol{\Theta}}_d = \left({}^{E}\boldsymbol{J}_g\right)^{-1} \begin{bmatrix} ^{E}\boldsymbol{v}_{ed} \\ ^{E}\boldsymbol{\omega}_{ed} \end{bmatrix}. \tag{60}$$

However, the singularities of $^{E}\boldsymbol{J}_g$ exist in most workspace. Dynamic singularities are functions of the system mass properties and cannot be predicted from its kinematic structure. These characteristics complicate the planning and control of free-floating space robot system. For the singularity avoiding of fixed-base manipulator, the DLS (damped least-squares) method[27,28] provides continuous and feasible solutions even at or in the neighborhood of singular points. Nevertheless, the exactness of the solutions in every direction of the end-effector task space has been sacrificed. To improve the exactness in the achievable directions and still keeping the merits of the DLS method, Cheng developed the "*singularity isolation plus compact quadratic programming*" (SICQP) method.[29] However, there are little literatures on dynamic singularity avoiding problem. In this paper, a practical approach is proposed to avoid the dynamic singularities, which transforms it into real-time kinematic singularities avoiding the problem.

4.5.1. Simplification of the kinematic equations. As pointed out above, the linear and angular momentums of free-floating system are conserved. By eliminating the holonomic constraints of linear momentum conservation, the total system is formulated as a nonholonomic system of $n + 3$ variables including three dependent variables.

Since there are no external forces on the system, with the assumption that the linear and angular momentum are initially zero, the center of mass of the system remains stationary, i.e.

$$\sum_{i=0}^{n} m_i \boldsymbol{r}_i = M\boldsymbol{r}_g. \tag{61}$$

Therefore, the base CM is determined by

$$\boldsymbol{r}_0 = \boldsymbol{r}_g - \frac{(m_1 + \cdots + m_n)(\boldsymbol{b}_0 + \boldsymbol{a}_1)}{M} - \cdots - \frac{m_n(\boldsymbol{b}_{n-1} + \boldsymbol{a}_n)}{M}. \tag{62}$$

Substituting (62) into (1), the position of the end-effector is

$$\boldsymbol{p}_e = \boldsymbol{r}_g + \hat{\boldsymbol{b}}_0 + \sum_{i=1}^{n} (\hat{\boldsymbol{a}}_i + \hat{\boldsymbol{b}}_i), \tag{63}$$

where

$$\hat{\boldsymbol{b}}_i = \frac{\sum_{q=0}^{i} m_q}{M} \boldsymbol{b}_i, \quad \hat{\boldsymbol{a}}_i = \frac{\sum_{q=0}^{i-1} m_q}{M} \boldsymbol{a}_i, \quad i = 1, \ldots, n. \tag{64}$$

Vectors $\hat{\boldsymbol{a}}_i$ and $\hat{\boldsymbol{b}}_i$ are aligned with $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$, respectively, and their lengths are constantly proportional to those of $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$. Therefore, $\hat{\boldsymbol{a}}_i$ and $\hat{\boldsymbol{b}}_i$ are called "virtual link vectors". Equation (63) is the direct kinematic equation of space robot when the holonomic constraints are eliminated. Differentiating both sides of (63), the following results are

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

obtained:

$$\boldsymbol{v}_e = \boldsymbol{v}_g + \dot{\hat{\boldsymbol{b}}}_0 + \sum_{i=1}^{n}(\dot{\hat{\boldsymbol{a}}}_i + \dot{\hat{\boldsymbol{b}}}_i)$$
$$= \boldsymbol{\omega}_0 \times \hat{\boldsymbol{b}}_0 + \sum_{i=1}^{n} \boldsymbol{\omega}_i \times (\hat{\boldsymbol{a}}_i + \hat{\boldsymbol{b}}_i), \qquad (65)$$

$\boldsymbol{\omega}_i$ is the angular velocity of ith body, which is calculated by

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_0 + \sum\nolimits_{k=1}^{i} \boldsymbol{k}_k \dot{\theta}_k. \qquad (66)$$

Hence, (65) can be simplified as

$$\boldsymbol{v}_e = -\tilde{\boldsymbol{p}}_{ge}\boldsymbol{\omega}_0 + \sum_{k=1}^{n} [\boldsymbol{k}_k \times (\boldsymbol{p}_e - \hat{\boldsymbol{p}}_k)]\dot{\theta}_k, \qquad (67)$$

$$\boldsymbol{p}_{ge} = \boldsymbol{p}_e - \boldsymbol{r}_g = \hat{\boldsymbol{b}}_0 + \sum_{i=1}^{n} \boldsymbol{A}_i({}^i\hat{\boldsymbol{a}}_i +{}^i \hat{\boldsymbol{b}}_i). \qquad (68)$$

Then, the differential kinematic equation of space robot can be given by combining (67) and (4)

$$\begin{bmatrix} \boldsymbol{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \hat{\boldsymbol{J}}_b \boldsymbol{\omega}_0 + \hat{\boldsymbol{J}}_m \dot{\boldsymbol{\Theta}}, \qquad (69)$$

where

$$\hat{\boldsymbol{J}}_b = \begin{bmatrix} -\tilde{\boldsymbol{p}}_{ge} \\ \boldsymbol{I}_3 \end{bmatrix} \in \boldsymbol{R}^{6\times 3}, \qquad (70)$$

$$\hat{\boldsymbol{J}}_m = \begin{bmatrix} \boldsymbol{k}_1 \times (\boldsymbol{p}_e - \hat{\boldsymbol{p}}_1) & \ldots & \boldsymbol{k}_n \times (\boldsymbol{p}_e - \hat{\boldsymbol{p}}_n) \\ \boldsymbol{k}_1 & \ldots & \boldsymbol{k}_n \end{bmatrix} \in \boldsymbol{R}^{6\times n}. \qquad (71)$$

4.5.2. The method to avoid the dynamic singularities. For the target pose relative to the end-effector is measured by the hand–eye camera, Eq. (69) is denoted by

$$\begin{bmatrix} {}^E\boldsymbol{v}_{ed} \\ {}^E\boldsymbol{\omega}_{ed} \end{bmatrix} = ({}^E\hat{\boldsymbol{J}}_b)({}^E\boldsymbol{\omega}_0) + ({}^E\hat{\boldsymbol{J}}_m) \cdot \dot{\boldsymbol{\Theta}}_d, \qquad (72)$$

where

$$\begin{bmatrix} {}^E\boldsymbol{v}_{ed} \\ {}^E\boldsymbol{\omega}_{ed} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_e^T & 0 \\ 0 & \boldsymbol{A}_e^T \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{ed} \\ \boldsymbol{\omega}_{ed} \end{bmatrix}, \qquad (73)$$

$${}^E\boldsymbol{\omega}_0 = \boldsymbol{A}_e^T \boldsymbol{\omega}_0. \qquad (74)$$

The base attitude $\boldsymbol{\Psi}_b$ and angular velocity $\boldsymbol{\omega}_0$ are generally measured by corresponding sensors in real time. Then, the following relationship is given according to Eq. (72):

$$\begin{bmatrix} {}^E\boldsymbol{v}_{ed} \\ {}^E\boldsymbol{\omega}_{ed} \end{bmatrix} - ({}^E\hat{\boldsymbol{J}}_b)({}^E\boldsymbol{\omega}_0) = {}^E\hat{\boldsymbol{J}}_m \dot{\boldsymbol{\Theta}}_d. \qquad (75)$$

The left-hand side of (75) is the end-effector velocity with relative to the base (expressed in the end-effector frame), which is denoted by

$${}^E\dot{\boldsymbol{x}}_{ed}^0 = \begin{bmatrix} {}^E\boldsymbol{v}_{ed}^0 \\ {}^E\boldsymbol{\omega}_{ed}^0 \end{bmatrix} = \begin{bmatrix} {}^E\boldsymbol{v}_{ed} \\ {}^E\boldsymbol{\omega}_{ed} \end{bmatrix} - ({}^E\hat{\boldsymbol{J}}_b)({}^E\boldsymbol{\omega}_0). \qquad (76)$$

So, (75) can be written as

$${}^E\dot{\boldsymbol{x}}_{ed}^0 = {}^E\hat{\boldsymbol{J}}_m \dot{\boldsymbol{\Theta}}_d, \qquad (77)$$

where, ${}^E\boldsymbol{x}_{ed}^0$ is the desired velocity (linear velocity and angular velocity) with respect to the base. From (71), ${}^E\hat{\boldsymbol{J}}_m$ is independent on the mass parameters, i.e. the singularities of ${}^E\hat{\boldsymbol{J}}_m$ are kinematic. It should be pointed out that the dynamic singularity points in the task space cannot be identified beforehand. However, when the angular velocity and the attitude of the base are measured in real time (at each sampling period), a simplified kinematic equation, i.e. (77), is derived. Equations (20), (72)–(74), and (77) show that, the desired inertial motion of the end-effector ($\boldsymbol{v}_{ed}, \boldsymbol{\omega}_{ed}$) at time $t + \Delta t$ can be realized by the planned joint motion ($\dot{\boldsymbol{\Theta}}_d$) according to Eq. (77), using the states of the space robotic system at time t. It is undeniable that there are errors in the results given by the ideal relationship (i.e. (20)), since the states at $t + \Delta t$ are different from the states at t. However, if Δt is short, or the desired motion is slow, the errors will not be large. And the actual relative pose between the end-effector and the target can be measured in real time, which means that the errors will not cumulate (the main procedure is shown in Fig. 6).

To solve Eq. (77), there only exist kinematic singularities. According to the definition of Jacobian, the following relationship exits:

$$({}^0\hat{\boldsymbol{J}}_m) = \begin{bmatrix} \boldsymbol{I} & -(\hat{L}_6 + \hat{L}_7)\tilde{\boldsymbol{Z}}_6 \\ \boldsymbol{O} & \boldsymbol{I} \end{bmatrix} ({}^0\hat{\boldsymbol{J}}_W). \qquad (78)$$

${}^0\hat{\boldsymbol{J}}_W$ is the wrist-reference Jacobian matrix establishing the relationship between the joint rates and the wrist velocities. It is of the following form:

$${}^0\hat{\boldsymbol{J}}_W = \begin{bmatrix} {}^0\hat{\boldsymbol{J}}_{11} & \boldsymbol{O}_{3\times 3} \\ {}^0\hat{\boldsymbol{J}}_{21} & {}^0\hat{\boldsymbol{J}}_{22} \end{bmatrix}. \qquad (79)$$

Then the method "Singularity Separation Plus Damped Reciprocal," proposed by Xu *et al.*,[30] can be used. The flowchart of the algorithm is shown in Fig. 7.

5. Simulation and Experiment Study of Autonomous Target Capturing

5.1. The space robotic system

The space robotic system is shown in Fig. 8. It is composed of a carrier spacecraft (called *space base* or *base*), a PUMA-type manipulator (called *space manipulator*) and a target satellite (called *target*) to be captured. The frames fixed on

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

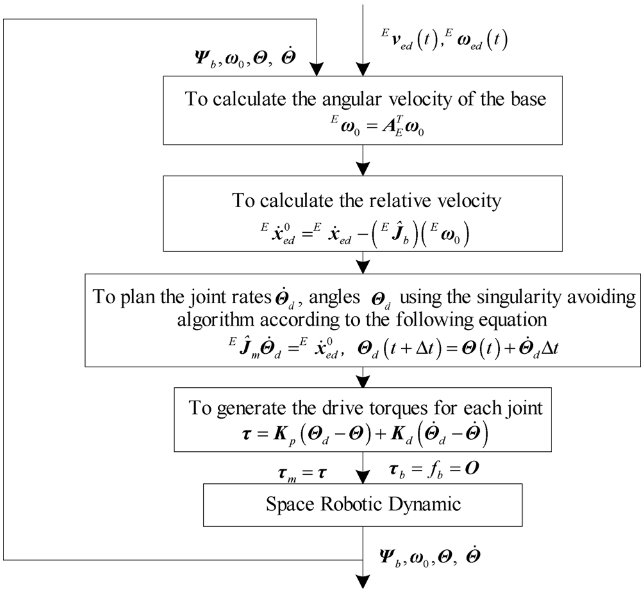
$\Psi_b, \omega_0, \Theta, \dot{\Theta}$
${}^E v_{ed}(t), {}^E \omega_{ed}(t)$
To calculate the angular velocity of the base
${}^E\omega_0 = A_E^T \omega_0$
To calculate the relative velocity
${}^E\dot{x}_{ed}^0 = {}^E\dot{x}_{ed} - ({}^E\hat{J}_b)({}^E\omega_0)$
To plan the joint rates $\dot{\Theta}_d$, angles Θ_d using the singularity avoiding algorithm according to the following equation
${}^E\hat{J}_m\dot{\Theta}_d = {}^E\dot{x}_{ed}^0$, $\Theta_d(t+\Delta t) = \Theta(t) + \dot{\Theta}_d\Delta t$
To generate the drive torques for each joint
$\tau = K_p(\Theta_d - \Theta) + K_d(\dot{\Theta}_d - \dot{\Theta})$
$\tau_m = \tau$
$\tau_b = f_b = O$
Space Robotic Dynamic
$\Psi_b, \omega_0, \Theta, \dot{\Theta}$

Fig. 7. Practical algorithm for avoiding the singularities of free floating robot.

the multibody system are defined as shown in Fig. 9 (when the joint angles are all zero), where Z_i is the direction of J_i. Table I lists the dimensions and mass properties of the bodies (Sat and B_i stand for the satellite and the ith body, respectively). The vectors ${}^i\boldsymbol{a}_i$, ${}^i\boldsymbol{b}_i$, and ${}^i\boldsymbol{I}_i$ are expressed in $\sum_i$. In the following sections, the origin of the inertia frame is the system' CM, i.e. $\boldsymbol{r}_g = \boldsymbol{O}$.

The coordinates of the four features in the light frame are

$$\begin{cases} \boldsymbol{P}_1^L = [0.01, 0.01, 0.0]^T, \boldsymbol{P}_2^L = [-0.01, 0.01, 0.0]^T \\ \boldsymbol{P}_3^L = [-0.01, -0.01, 0.0]^T, \boldsymbol{P}_4^L = [0.01, -0.01, 0.0]^T \end{cases}. \tag{80}$$

The homologous transformation matrix from the light frame to the handle frame is

$${}^L\boldsymbol{T}_H = \begin{bmatrix} 0.0269 & 0.9996 & 0.0134 & 0.1076 \\ -0.9994 & 0.0272 & -0.0221 & 0.1492 \\ -0.0224 & -0.0128 & 0.9997 & -0.1963 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}. \tag{81}$$

The pose of the handle frame $\sum_H$ with respect to $\sum_T$ is assumed to be

$${}^T\boldsymbol{T}_H = \begin{bmatrix} 1 & 0 & 0 & 0.0050 \\ 0 & -1 & 0 & 0.0000 \\ 0 & 0 & -1 & 0.2200 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{82}$$

5.2. Simulation study of autonomous target capturing

5.2.1. The simulation model. Computer simulations have been performed under Simulink®, developed by the MathWorks, with customized S-function blocks written in C in addition to building blocks. The model is composed of four modules (see Fig. 10): planner, joint controllers (PDs), hand–eye camera, target model, and space robot dynamic.

The *planner* autonomously plans the end-effector's velocities and generates the desired joint angles ($\boldsymbol{\Theta}_d$) and rates ($\dot{\boldsymbol{\Theta}}_d$), according to the current system state (the base attitude $\boldsymbol{\Psi}_b$, angular velocity $\boldsymbol{\omega}_b$, and joint angles $\boldsymbol{\Theta}$) and the hand–eye measurement ($\boldsymbol{U}V$ and $\boldsymbol{Z}c$, i.e. the image features and the estimated depth values). It also judges whether the target within the capturing box (i.e. each component of $\boldsymbol{U}V$ is less than 10 pixels), if so, the "CapFlag" is set to one and the simulation stops, showing that the target is captured. The

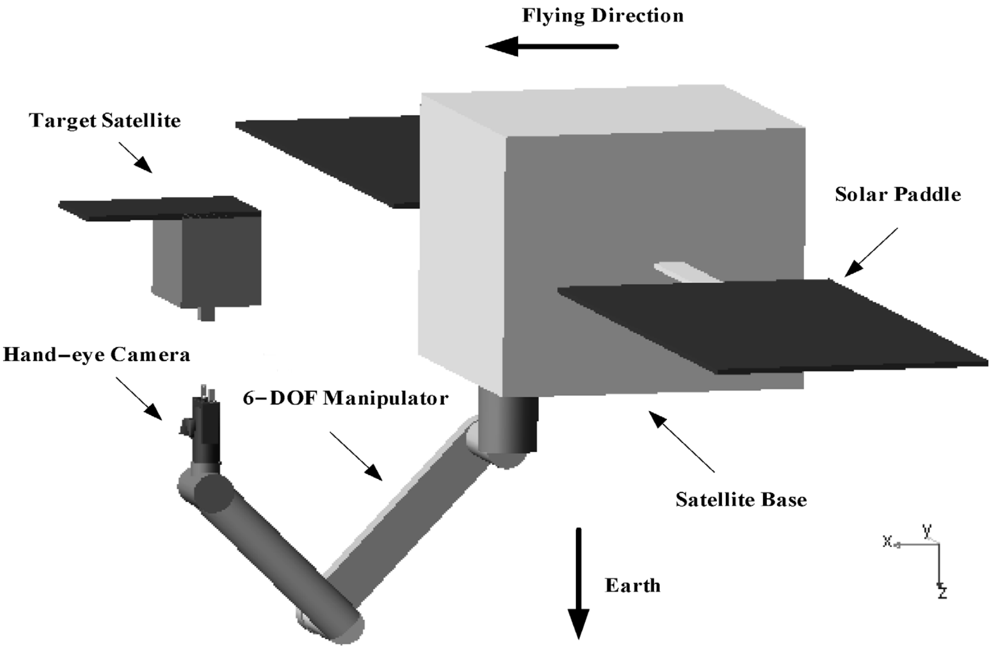

Fig. 8. The space robotic system.

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

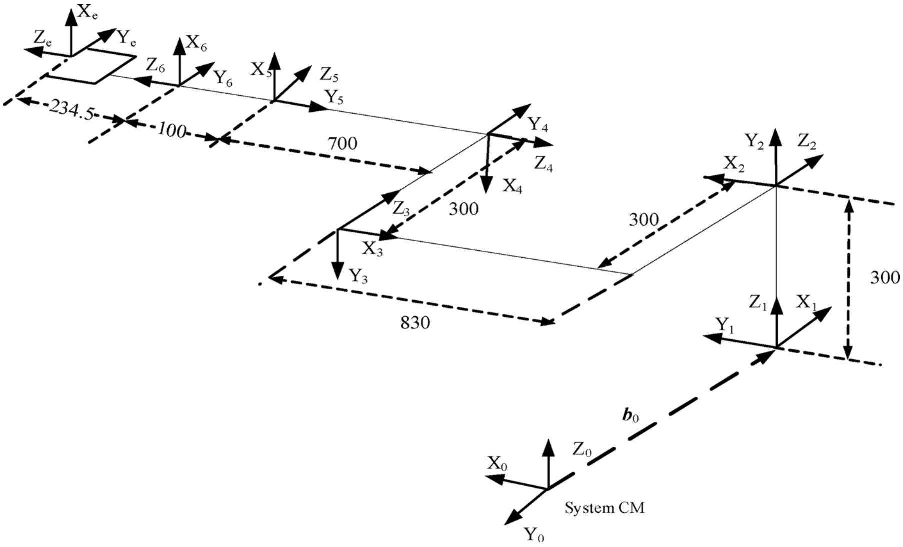

Fig. 9. The body-fixed frames of the space robotic system.

joint controllers, including six joint controllers, generates the control torques ($\boldsymbol{T}_m$) and drives the six joints to track the desired trajectories (i.e. $\boldsymbol{\Theta}_d$ and $\dot{\boldsymbol{\Theta}}_d$). The *space robot dynamic,* modeling the space robotic system based on the essential equations of (21), calculates the system dynamic state according to the drive forces and torques (for the free-floating mode, the forces and torques exerting on the base are zero, i.e. $\boldsymbol{f}_b = \boldsymbol{T}_b = [0, 0, 0]^T$). The *hand–eye camera*, simulating the hand–eye camera, extracts the image features of the light points $\boldsymbol{U}V$ and estimates the corresponding depth values $\boldsymbol{Z}c$, using the current state of the space robotic system (the base position $\boldsymbol{r}_b$, attitude $\boldsymbol{\Psi}_b$, and joint angles $\boldsymbol{\Theta}$) and the target (the inertia position $\boldsymbol{P}_t$ and attitude $\boldsymbol{\Psi}_t$). The *target model* models the space target and outputs the position $\boldsymbol{P}_t$ and orientation $\boldsymbol{\Psi}_t$ of the target's CM, according to its linear velocity $\boldsymbol{V}_t$ and angular velocity $\boldsymbol{W}_t$.

5.2.2. The intrinsic and extrinsic parameters of the hand–eye camera. The parameters are from the calibration results of the left camera used in the experiment system (see Section 5.3). The intrinsic parameters of the hand–eye camera are

$$\begin{cases} u_0 = 256.03758,\ v_0 = 256.45714 \\ f = 12.8\times 10^{-3},\ \alpha_x = 1.5331\times 10^5,\ \alpha_y = 1.5289\times 10^5. \end{cases} \tag{83}$$

The extrinsic parameters of the ***camera are***

$$^{E}\boldsymbol{T}_C = \begin{bmatrix} -0.0255 & -0.9994 & 0.0225 & 0.1529 \\ 0.9996 & -0.0257 & -0.0086 & -0.1128 \\ 0.0093 & 0.0222 & 0.9997 & -0.2282 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}. \tag{84}$$

5.2.3. Simulation of capturing a moving target when no dynamic singularities. Initially, the joint angles and the base attitude are

$$\boldsymbol{\Theta}_0 = [0°, 47.7200°, -93.9100°, 0°, -4308200°, 0°], \tag{85}$$

Table I. The mass properties of the space robotic system.

								B_6	
		Sat	B_1	B_2	B_3	B_4	B_5	Before capturing	After capturing
Mass (kg)		400	6	5	5	4	3	2	30
$^i\boldsymbol{a}_i$			0	0.2702	0	0	0	0	−4.7E-003
			0	0	0	0	−0.0338	0	0
			0.15	−0.2513	0.15	−0.35	0	0.0750	0.4292
		0.3570	0	0.5598	0	0	0	0	0
$^i\boldsymbol{b}_i$		−0.0095	0	0	0	0	−0.0662	0	0
		0.419	0.15	−0.0487	0.15	−0.35	0	0.1595	0
	I_{xx}	30	0.15	0.0926	0.105	0.2498	0.0330	5.152E-002	0.7992
	I_{yy}	28	0.15	0.9053	0.105	0.2498	0.0172	5.152E-002	0.8925
$^i\boldsymbol{I}_i$ (kg m^2)	I_{zz}	32	0.075	0.8451	0.0294	0.0196	0.0260	2.192E-002	0.4682
	I_{xy}	0.26	0	0	0	0	0	0	−2.3E-002
	I_{xz}	0.37	0	0.1315	0	0	0	0	4.458E-003
	I_{yz}	−0.29	0	0	0	0	0	0	1.7E-002

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

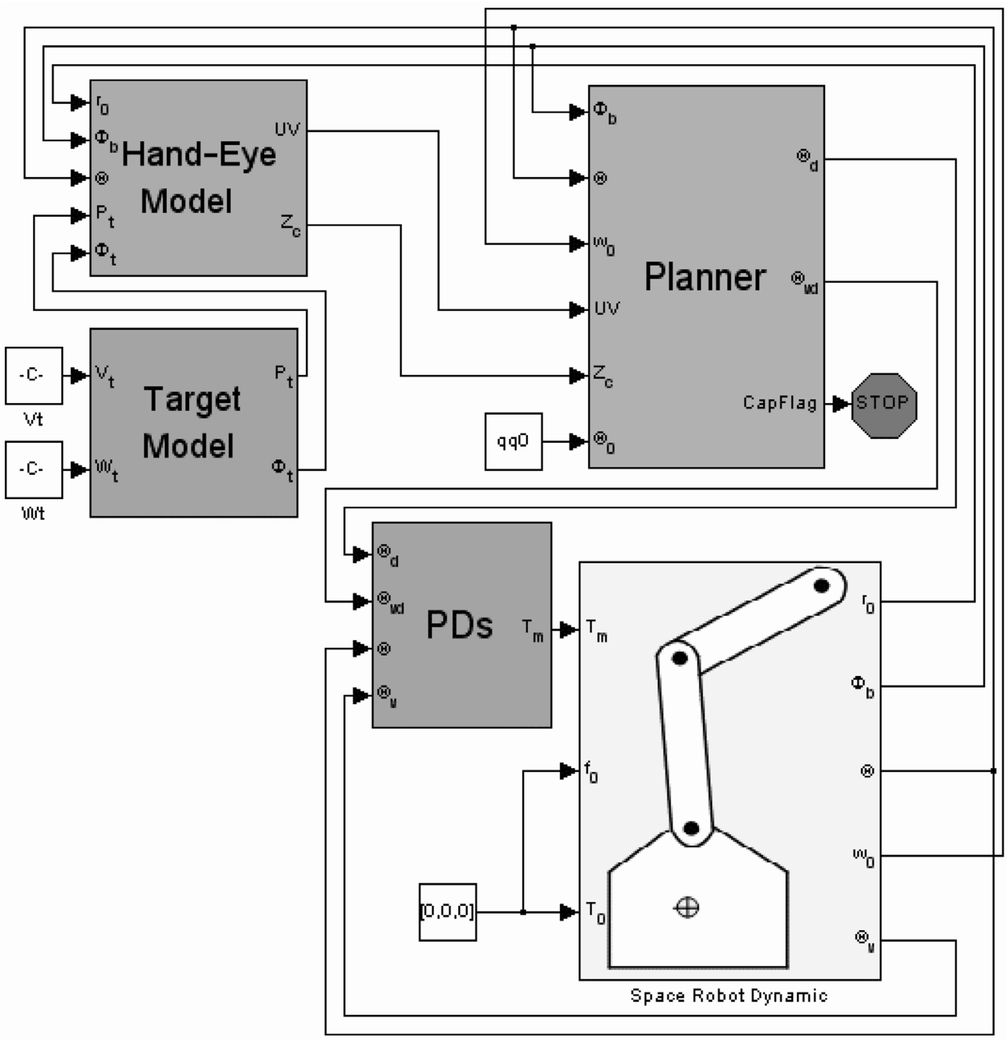

Fig. 10. The simulation model created in Simulink for capturing a moving target.

$$\boldsymbol{\Psi}_{b0} = [0 \quad 0 \quad 0]^T. \tag{86}$$

And the poses of the end-effector and the target are

$$\begin{aligned} {}^I\boldsymbol{X}_{e0} = [&1.6892\,\text{m} \quad -0.0137\,\text{m} \quad 0.7943\,\text{m} \quad 0.00^\circ \\ &-69.99^\circ \quad 180.00^\circ], \end{aligned} \tag{87}$$

$$\begin{aligned} {}^I\boldsymbol{X}_{t0} = [\,]\, &1.2955\,\text{m} \quad 0.0363\,\text{m} \\ &-0.0799\,\text{m} \quad 5.00^\circ \quad 5.00^\circ\ 5.00^\circ, \end{aligned} \tag{88}$$

The target is assumed to move with the constant velocities

$$\begin{cases} \boldsymbol{v}_t = [5\,\text{mm/s}, 5\,\text{mm/s}, 5\,\text{mm/s}] \\ \boldsymbol{\omega}_t = [0\ \text{deg/s}, 0\ \text{deg/s}, 0\ \text{deg/s}] \end{cases}. \tag{89}$$

The desired image features are

$$\begin{aligned} f_d = [&317.6644 \quad 394.7437, 225.5703 \quad 399.5873, \\ &220.8050 \quad 307.7116, 312.7073 \quad 302.9008]^T. \end{aligned} \tag{90}$$

According to the simulation results, the image features approach the desired features (see Fig. 11). The final image feature errors are all less than 10 pixels, which indicate that the space robot captures the target successfully. The trajectories of the joint angles are shown in Fig. 12. Within the whole capturing process, the CM position of the base changes from [−0.0495 m, −0.0042 m, −0.0534 m] to [−0.0495 m, −0.0051 m, −0.0455 m], and the base attitude changes from [0, 0, 0] to [−1.4504°, −9.0923°, 1.2312°].

5.2.4. Simulation of capturing a moving target when existing dynamic singularities. The parameters used to avoid the singularities are set as

$$\lambda_0^2 = 0.08, \quad \varepsilon_i = \varepsilon_b = 0.05, \quad \varepsilon_w = 0.15. \tag{91}$$

The constant ε_i, ε_b, and ε_w are the thresholds to determine the corresponding singularity areas, and λ_0 is the nominal damped coefficient.[29]

Initially, the joint angles and the base attitude are

$$\begin{aligned} \boldsymbol{\Theta}_0 = [&0.7162^\circ \quad 50.9990^\circ \quad -88.9116^\circ \quad 1.2490^\circ \\ &16.6960^\circ \quad -1.7934^\circ], \end{aligned} \tag{92}$$

$$\boldsymbol{\Psi}_{b0} = \left[0 \quad 0 \quad 0\right]^T. \tag{93}$$

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

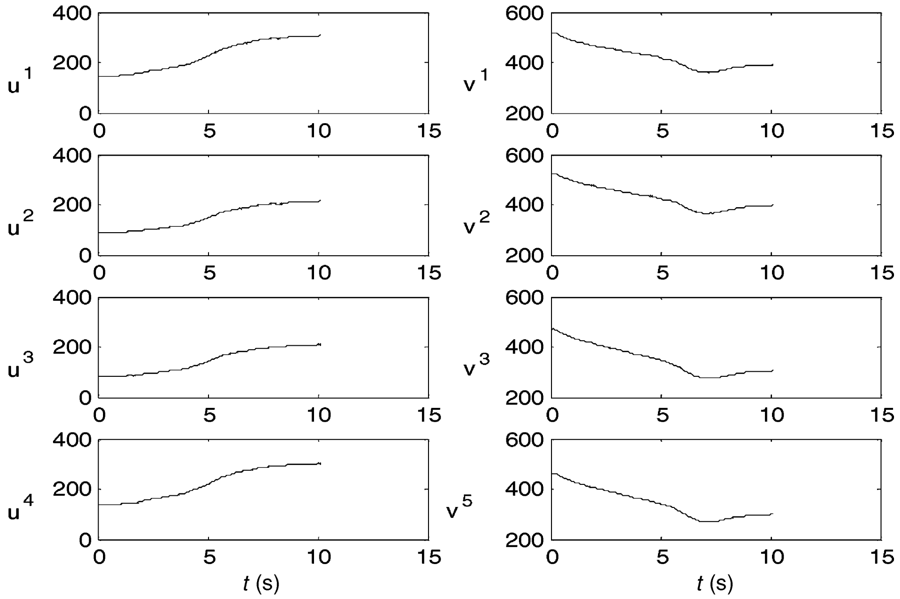

Fig. 11. The variation of the camera image features.

And the poses of the end-effector and the target are

$$^{I}\boldsymbol{X}_{e0} = [1.6927\,\text{m} \quad 0.0053\,\text{m} \quad 0.7566\,\text{m} \quad 8.9337^\circ \quad -68.5982^\circ \quad 172.7015^\circ] \tag{94}$$

$$^{I}\boldsymbol{X}_{t0} = [1.4222\,\text{m} \quad -0.1511\,\text{m} \quad 0.1653\,\text{m} \quad 19.5426^\circ \quad -11.2133^\circ \quad 162.3645^\circ] \tag{95}$$

The target is assumed to move with the constant velocities

$$\begin{cases} \boldsymbol{v}_t = [10\,\text{mm/s}, 10\,\text{mm/s}, 10\,\text{mm/s}] \\ \boldsymbol{\omega}_t = [1\,\text{deg/s}, 1\,\text{deg/s}, 1\,\text{deg/s}] \end{cases} \tag{96}$$

The space manipulator is required to capture the moving target. The end-effector's velocities are planned based on the measurement of the hand–eye camera. Using the traditional method (i.e. the joint rate is calculated by $\dot{\boldsymbol{\Theta}} = \boldsymbol{J}_g^{-1}\dot{x}_e$), the joint rates become infinite and discontinuous at the dynamic singularity, which is shown in Fig. 13. The planned motions of the joints are not feasible for the control.

Fortunately, the proposed practical algorithm can handle the dynamic singularities well. The simulation results are shown in Fig. 14. Different from the traditional method, the joint rates (shown in Fig. 14) are finite and continuous, which is adaptive to control the space manipulator. Fig. 15 illustrates the joint angle trajectories. The position and attitude of the base are shown in Fig. 16. Correspondingly, the end-effector tracks the inertial motion of the handle and captures it successfully. The inertial trajectories of the end-effector ($[P_{ex}, P_{ey}, P_{ez}]^T$) and the handle ($[P_{hx}, P_{hy}, P_{hz}]^T$) are shown in Fig. 17. During the tracking, the relative

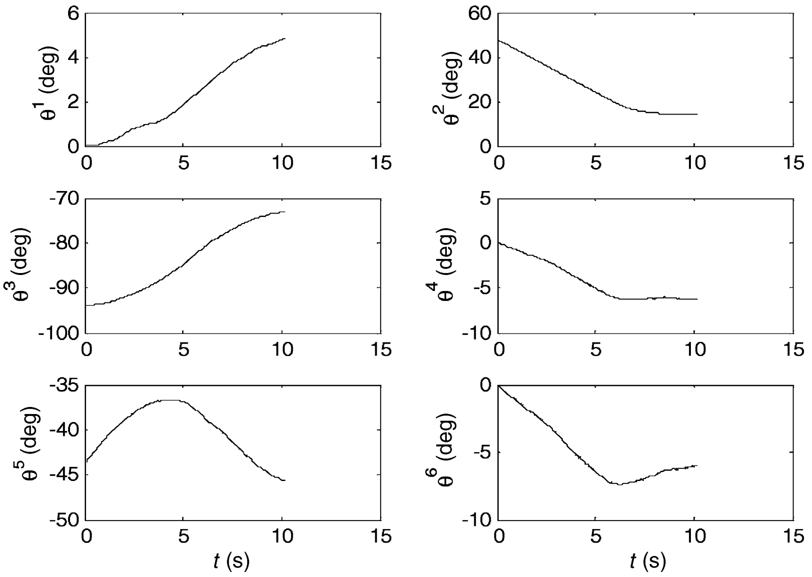

Fig. 12. The on-line planned joint trajectory.

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

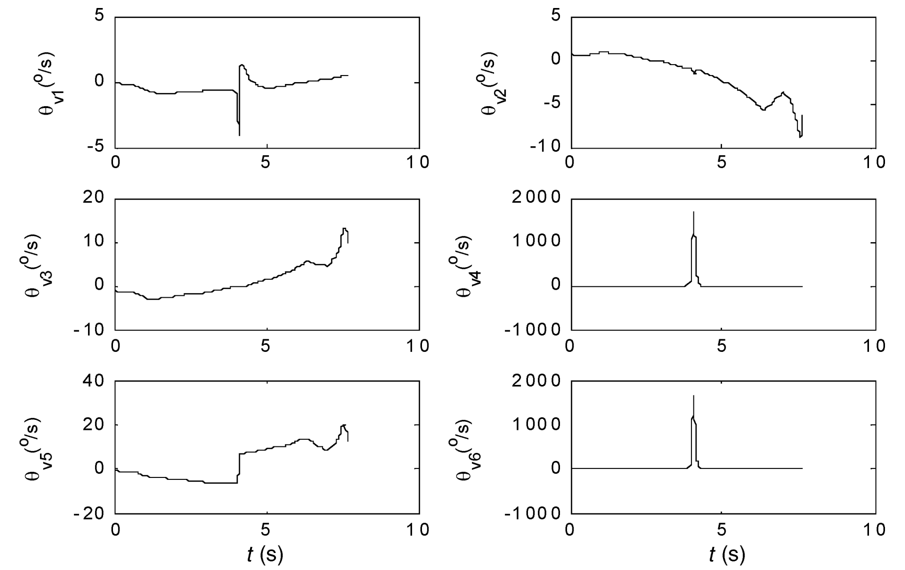

Fig. 13. The real joint rates using the traditional inverse of generalized Jacobian.

position and attitude vary as shown in Fig. 18. These results verify that the proposed method can be used to solve the dynamic singularity problem, and good performance will be attained.

5.3. Experiment study of autonomous target capturing

5.3.1. The experiment system of target capturing. A semi-physical simulation system was developed by Li *et al.*,[26] but it is not a real experiment system. The hardware-in-the-loop (HIL) simulators are powerful to verify the key algorithms of space robotic system.[31,32] Yoshida *et al.*[33] and Settelmeyer *et al.*[34] used two industrial manipulators to experiment the capture of a noncooperative satellite. We also set up a similar experiment system.[35] The experiment concept is illustrated in Fig. 19. Two industrial robots, i.e. *capturing robot* (Robot C) and *target robot* (Robot T), are controlled to implement the motion of the space robotic system. The end-effector (include the hand–eye camera) of *space robot* (Robot S) is mounted on Robot C, and the target mockup is installed on Robot T. The bases of Robots C and T are assumed to be fixed in the inertial frame (i.e. $\sum_C$ and $\sum_T$ is fixed relative to $\sum_I$, but $\sum_B$is free floating relative to $\sum_I$). In the laboratory environment, Robot C is used to implement the absolute motion of Robot S's end-effector (i.e. $\dot{\boldsymbol{r}}_e$), and Robot T is used to implement that of the target (i.e. $\dot{\boldsymbol{r}}_h$).

The ground experiment system established in our institute is shown in Fig. 20. It is composed of two industrial

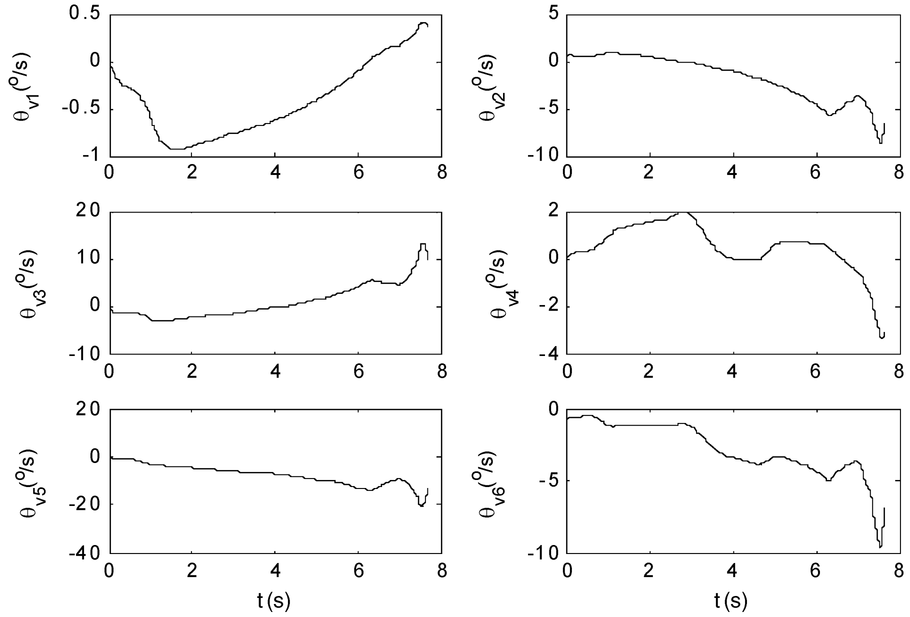

Fig. 14. The real joint rates during the maneuver using the practical algorithm to avoid the dynamic singularities.

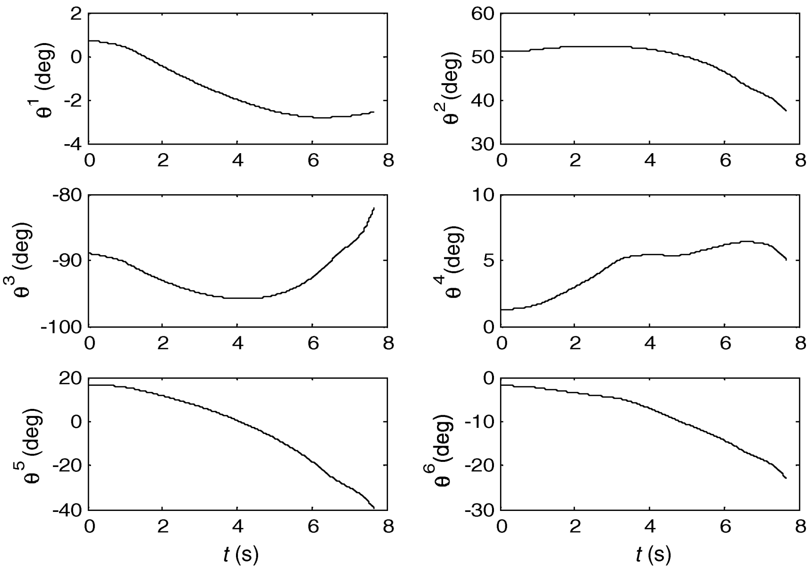

Fig. 15. The real joint angles during the maneuver using the practical algorithm to avoid the dynamic singularities.

robots (Motoman K10, including their controllers), two hand–eye cameras, two global cameras (used to observe the capturing process), and five industrial computers. The five computers are *Simu3D*, *Computer C*, *Image Processor*, *Global Displayer,* and *Computer T*. In these computers, *Computer C* is used to implement the dynamic emulation and kinematics equivalence algorithm. *Computer T* plans the motion of Robot T according to the target motion. *Image Processor* processes the images of hand–eye cameras and outputs the measurement results of the target pose relative to the end-effector. *Simu3D* creates the geometry model of space robot using OpenGL, and it is utilized to simulate the motion of the whole space robotic system in real time. *Global displayer* displays the images of the global cameras.

5.3.2. The calibration results of the two hand–eye cameras. The calibrated parameters of the left camera used in the experiment system are given as Eqs. (83) and (84). The intrinsic parameters of the right camera are

$$\begin{cases} u_0\_R = 254.53941, \quad v_0\_R = 256.85139 \\ f\_R = 12.8 \times 10^{-3}, \quad \alpha_x\_R = 1.5229 \times 10^5, \\ \alpha_y\_R = 1.5186 \times 10^5 \end{cases} \tag{97}$$

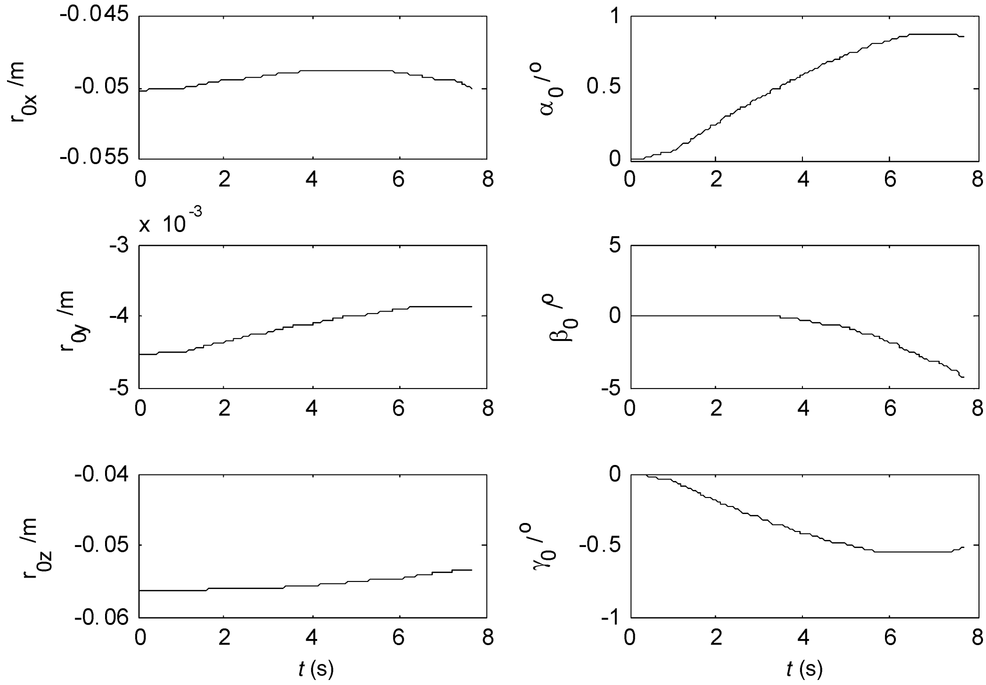

Fig. 16. The variation of the base position and attitude.

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

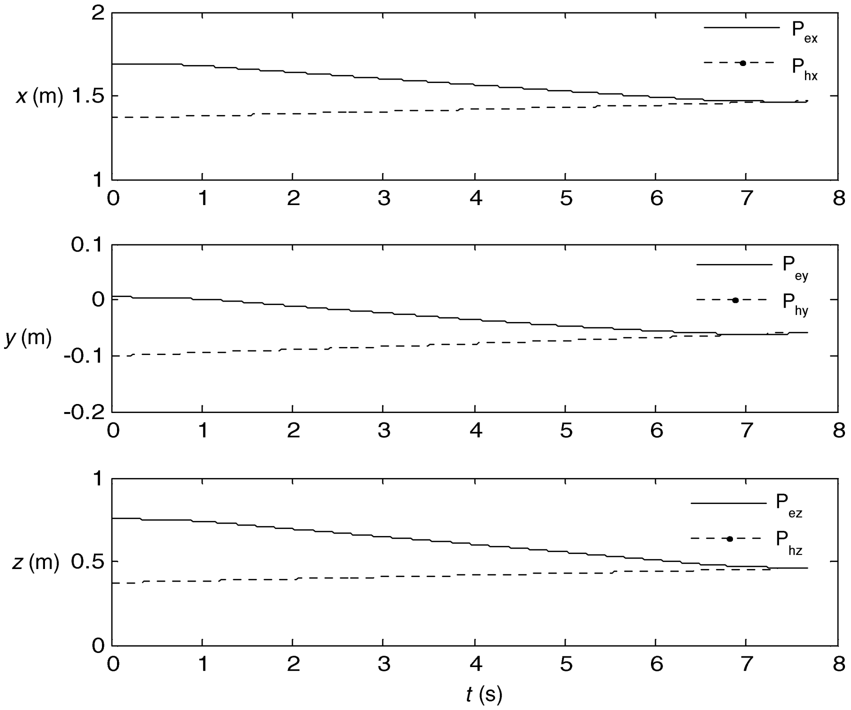

Fig. 17. The variation of the inertial position of the end-effector and handle.

And the extrinsic parameters of the ***right camera are***

$$
{}^{E}\boldsymbol{T}_{C\text{-}R} = \begin{bmatrix} -0.0102 & -0.9998 & 0.0154 & 0.1475 \\ 0.9999 & -0.0102 & 0.0017 & 0.1073 \\ -0.0015 & 0.0154 & 0.9999 & -0.2256 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{98}
$$

5.3.3. Experiment of capturing moving target in inertial space. The algorithm of capturing the moving target is also experimented. Initially, the attitude and CM position of the base is

$$
\boldsymbol{r}_0 = [-0.0495\,\mathrm{m} \quad -0.0042\,\mathrm{m} \quad -0.0534\,\mathrm{m}]^T,
$$
$$
\boldsymbol{\Psi}_0 = [0^\circ \quad 0^\circ \quad 0^\circ] \tag{99}
$$

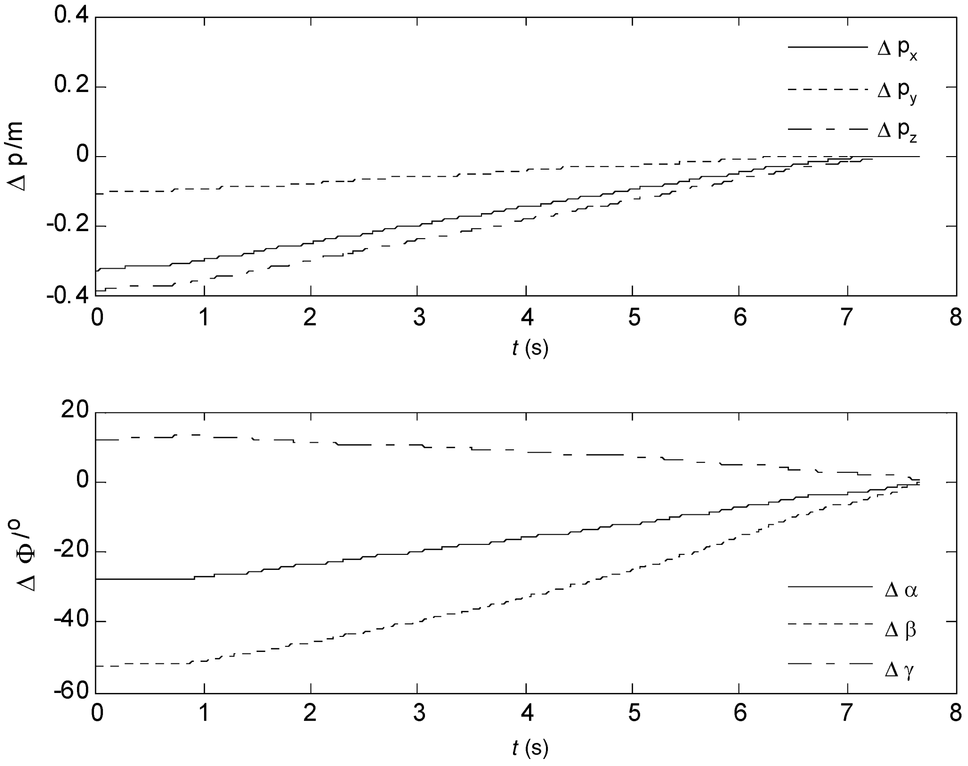

Fig. 18. The relative position and attitude between the end-effector and the handle.

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

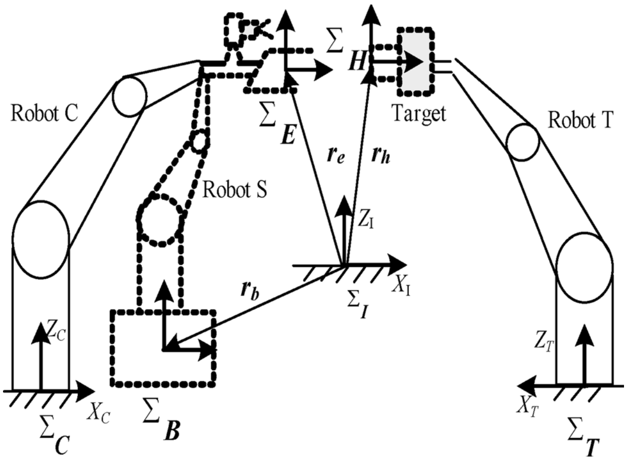

Fig. 19. The concept of the kinematic equivalence.

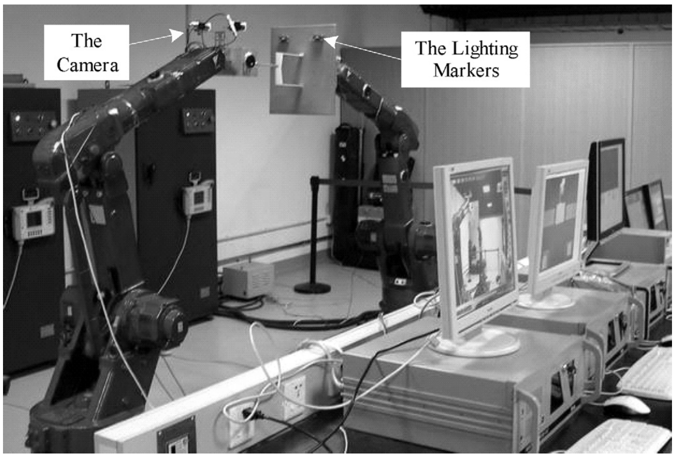

Fig. 20. The setup of the ground experiment system in our lab.

And the joint angles of space robot is

$$\mathbf{\Theta}_0 = [0^\circ \quad 47.7212^\circ \quad -93.9074^\circ\ 0^\circ \quad -43.8137^\circ \quad 15.0000^\circ]. \tag{100}$$

The position and attitude of the handle frame with respect to the inertial frame are

$$\boldsymbol{r}_{h0} = [1.3520\,\text{m} \quad -0.0290\,\text{m} \quad 0.1335\,\text{m}]^T,$$
$$\boldsymbol{\Psi}_{h0} = [0^\circ \quad 0^\circ \quad 180^\circ]^T. \tag{101}$$

For the experiment of target capturing, the gain matrix of Eq. (47) is set as identity matrix, and the upper limits of the end-effector linear velocity and angular velocity are 40 mm/s and 5 deg/s, respectively.

Assume the target is moving with the constant velocities

$$\boldsymbol{v}_h = [-5\,\text{mm/s},\ 5\,\text{mm/s},\ -5\,\text{mm/s}],$$
$$\boldsymbol{\omega}_h = [0.5\,\text{deg/s},\ 0.5\,\text{deg/s},\ 0.5\,\text{deg/s}]. \tag{102}$$

The experiment results are shown in Fig. 21–25. Fig. 21 and 22 are the cures of the joint rates and joint angles. Fig. 23 shows the real pose of the handle relative to the end-effector. The initial and final states of the ground experiment system are illustrated in Fig. 24, respectively. After capturing the target, the CM position of the base changes from [−0.0475 m, −0.0042 m, −0.0583 m] to [−0.0503 m, −0.0025 m, −0.0433 m], and the base attitude changes from [0, 0, 0] to [1.7804°, −10.7371°, −0.6903°].

6. Discussion and Conclusion

The autonomous ability, including autonomous prediction of target motion and autonomous avoidance of singularities, is very important for target capturing. Since the 2D image features are used for the path planning, the approach is not sensitive to the calibration error and it avoids the complex 3D reconstruction process. But it needs to compute the image Jacobian matrix on real time. The singularity of the image Jacobian matrix will disable the algorithm. As it is proved that, the image singularity does not exist when using the redundant ($N \geq 4$) nondegenerated image feature points.[36]

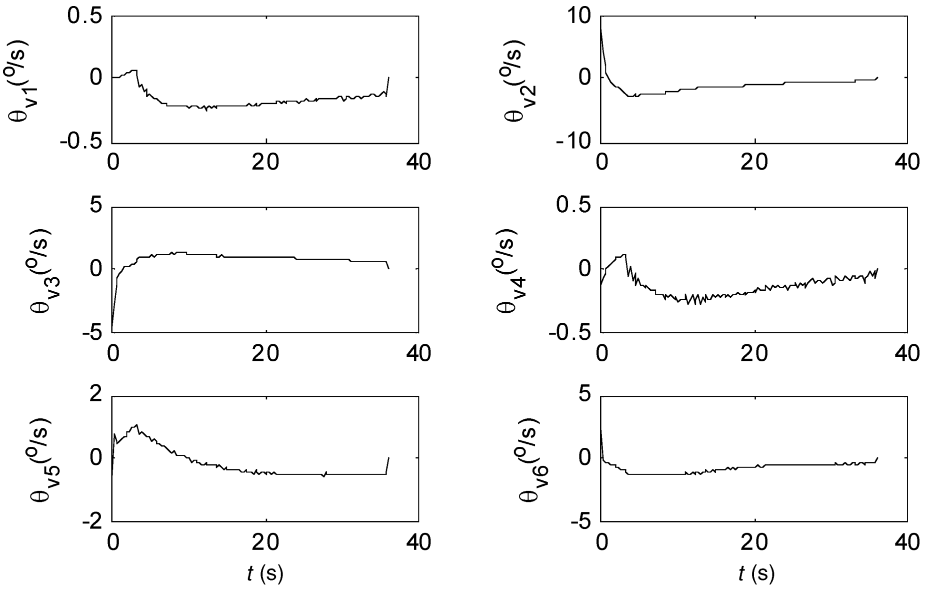

Fig. 21. The variation of the left camera image feature for capturing moving target.

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

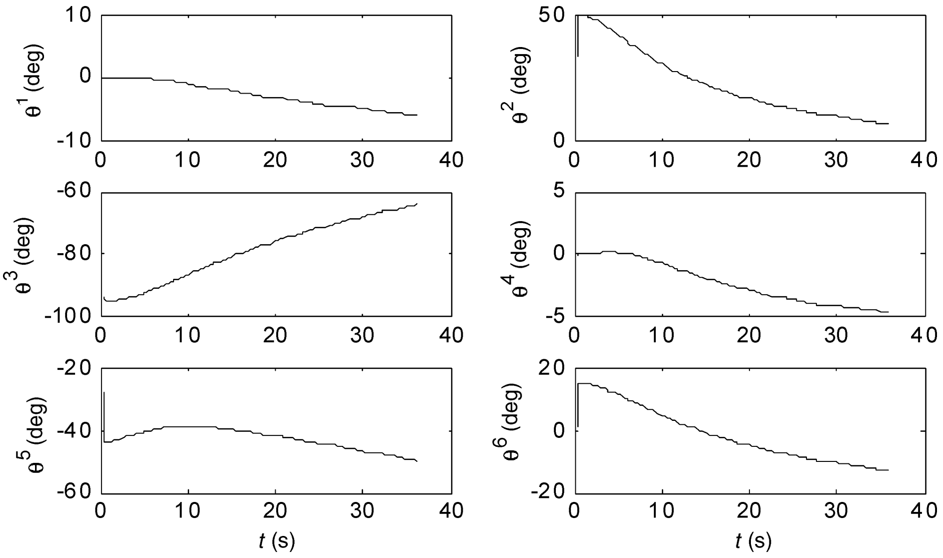

Fig. 22. The variation of the left camera image feature for capturing moving target.

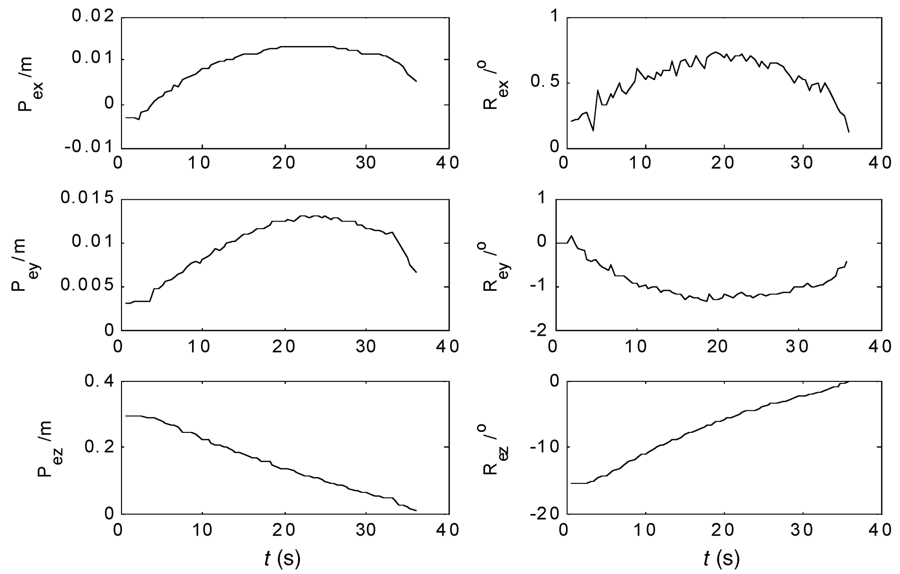

Fig. 23. The curves of relative pose for capturing moving target.

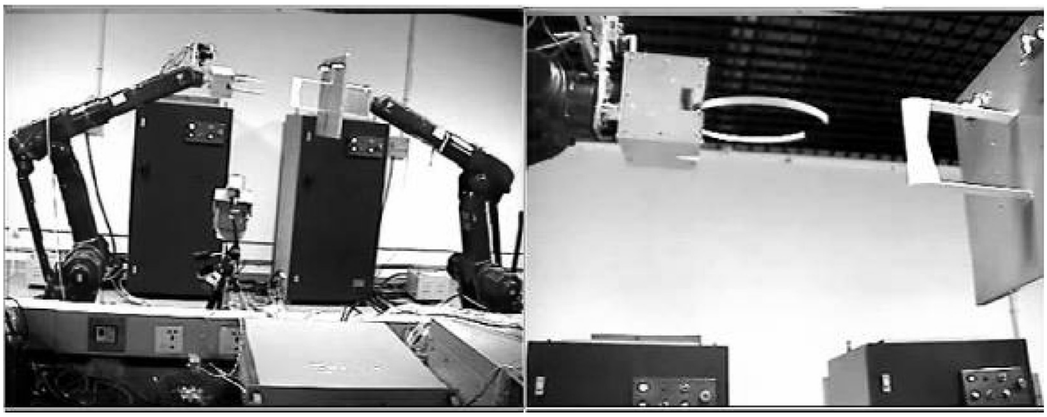

Fig. 24. The initial state of the experiment for capturing moving target (displayed by the two global cameras).

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

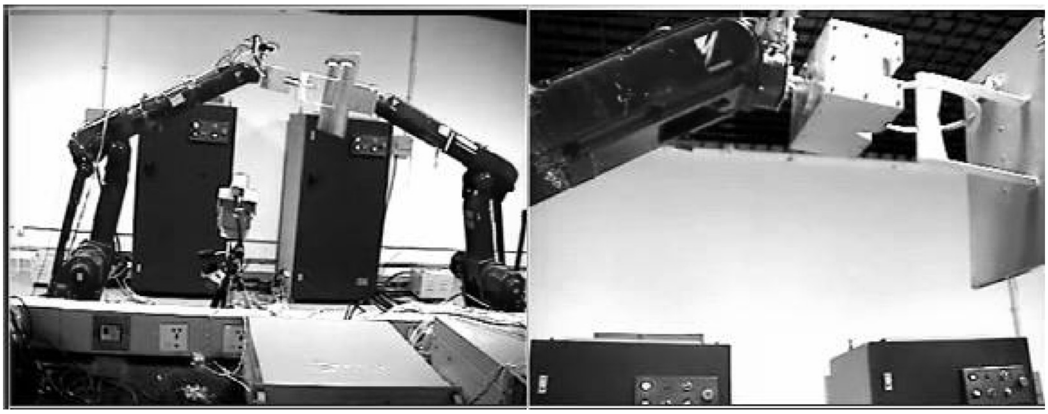

Fig. 25. The final state after capturing moving target (displayed by the two global cameras).

Dynamic singularities, which affect the control of the end-effector, are the intrinsic characteristics of space robot. So a practical algorithm is presented to avoid their effect. The base angular velocity is first estimated on real time according to angular momentum conservation equation. Then, the approximate linear relationship between the end-effector velocity and joint rates is established, by substituting the estimated values to the differential kinematic equations. At last, the dynamic singularities avoiding problem are transformed to the kinematic singularities avoiding problem, based on the estimation and measurement on real time.

To examine the proposed approach, numeric simulation and experiment study are introduced. The simulation model is created using Simulink. The experiment system is set up using cheap devices. The key of the realization of the experiment system is the dynamic simulation and kinematic equivalence. The system can emulate the capturing process observed from the inertia frame. And the geometry and mass properties of space robot are not limited. The system can be extended by small modification to verify different technology of target capturing. And, the real-time 3D simulation system visualizes the capturing process intuitively. The path planning methods proposed above are tested and verified using the experiment system.

In this paper, the target is assumed to be cooperative, i.e. some feature points (light points) are carefully designed for the visual measurement. In order to extend the application of the proposed method, we will study the measurement for the noncooperative target without special visual markers. On the other hand, the proposed method requires the knowledge of the inertia parameters (mass, position of CM, moment of inertia, and the product of inertia) of each body. These parameters were estimated from the design specification, but their real values in orbit may be very different. In the future, we will study practical approaches to identify the inertia parameters of each body.

Acknowledgments

We would like to thank Dr. L. J. Xue and Dr. L. T. Li for their valuable suggestion on the paper. This work is supported by National Nature Science Foundation of China (No. 60775049).

References

1. G. Hirzinger, K. Landzettel, B. Brunner, M. Fischer and C. Preusche, “DLR’s robotics technologies for on-orbit servicing,” *Adv. Rob.* **18**(2), 139–174 (2004).
2. K. Landzettel, C. Preusche, A. Albu-Schaffer and D. Reintsema, “Robotic On-Orbit Servicing–DLR’s Experience and Perspective,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China (2006) pp. 4587–4594.
3. K. Yoshida, “Engineering test satellite VII flight experiments for space robot dynamics and control: Theories on laboratory test beds ten years ago, now in orbit,” *Int. J. Rob. Res.* **22**(5), 321–335 (2003).
4. J. R. Wilson, “Satellite hopes ride on orbital express,” *Aerospace Am.* **45**(2), 30–35 (2007).
5. B. Web, Orbital Express–Mission Updates. Available at: http://www.boeing.com/ids/advanced_systems/orbital/updates.html (2007).
6. Y. Nakamura and R. Mukherjee, “Nonholonomic path planning of space robots via a bidirectional approach,” *IEEE Trans. Rob. Automat.* **7**(4), 500–514 (1991).
7. W. F. Xu, Y. Liu, B. Liang, Y. S. Xu, C. Li and W. Y. Qiang, “Nonholonomic Path Planning of a Free-Floating Space Robotic System Using Genetic Algorithms”, *Advanced Robotics*, **22**(4), 451–476 (2008).
8. F. M. Carter and D. B. Cherchas, “Motion control of non-fixed base robotic manipulators,” *Robotica* **17**(2), 143–157 (1999).
9. G. de Rivals-Mazères, W. Yim, F. Mora-Camino and S. N. Singh, “Inverse control and stabilization of free-flying flexible robots,” *Robotica* **17**(3), 343–350 (1999).
10. F. Caccavale and B. Siciliano, “Kinematic control of redundant free-floating robotic systems,” *Adv. Rob.* **15**, 429–448 (2001).
11. F. Caccavale and B. Siciliano, “Quaternion-Based Kinematic Control of Redundant Spacecraft/Manipulator Systems,” *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea (2001) pp. 435–440.
12. K. Yoshida and Y. Umetani, “Control of space free-flying robot,” *Proceedings of the 29th Conference on Decision and Control*, Honolulu, Hawaii (1990) pp. 97–102.
13. Y. Umetani and K. Yoshida, “Workspace and manipulability analysis of space manipulator,” *Trans. Soc. Instrum. Control Eng.* **E-1**(1), 116–123 (2001).
14. S. K. Agrawal, M. Y. Chen and M. Annapragada, “Modeling and simulation of assembly in a free-floating work environment by a free-floating robot,” *Trans. ASME J. Mech. Des.* **118**(1), 115–120 (1996).
15. E. Papadopoulous, S. Ali and A. Moosavian, “Dynamics and Control of Multi-Arm Space Robots During Chase and Capture Operations,” *Proceedings of the IEEE International*

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press

Conference on Intelligent Robots and Systems, Munich, Germany (1994) pp. 1554–1561.

16. H. Nagamatsu, T. Kubota and I. Nakatani, "Capture Strategy for Retrieval of a Tumbling Satellite by a Space Robotic Manipulator," *Proceedings of the IEEE Conference on Robotic and Automation*, Minneapolis, MN (1996) pp. 70–75.
17. H. Nagamatsu, T. Kubota and I. Nakatani, "Autonomous Retrieval of a Tumbling Satellite Based on Predictive Trajectory," *Proceedings of the International Conference on Robotics and Automation*, Albuquerque, New Mexico (1997) pp. 3074–3079.
18. P. F. Huang, Y. S. Xu and B. Liang, "Tracking trajectory planning of space manipulator for capturing operation," *Int. J. Adv. Rob. Syst.* **3**(3), 211–218 (2006).
19. R. A. McCourt and C. W. de Silva, "Autonomous robotic capture of a satellite using constrained predictive control," *IEEE/ASME Trans. Mechatron.* **11**(6), 699–708 (2006).
20. Y. S. Xu and T. Kanade, *Space Robotics: Dynamics and Control*, Massachusetts, USA (Kluwer Academic Publishers, 1992).
21. S. A. Moosavian and E. Papadopoulos, "Free-flying robots in space: An overview of dynamics modeling, planning and control," *Robotica* **25**(5), 537–547 (2007).
22. Y. Umetani and K. Yoshida, "Resolved motion rate control of space manipulators with generalized Jacobian matrix," *IEEE Trans. Rob. Automat.* **5**(3), 303–314 (1989).
23. E. Papadopoulos and S. Dubowsky, "Dynamic singularities in the control of free-floating space manipulators," *ASME J. Dyn. Syst., Meas. Control* **115**(1), 44–52 (1993).
24. L. Yin, R. Yang, M. Gabbouj and Y. Neuvo, "Weighted median filters: A tutorial," *IEEE Trans. Circuits and Systems II* **43**(3), 157–192 (1996).
25. M. Sonka, V. Hlavac and R. Boyle, *Image Processing, Analysis, and Machine Vision*, USA, 3rd ed. (Cengage-Learning, 2007).
26. C. Li, B. Liang and W. F. Xu, "Autonomous Trajectory Planning of Free-Floating Robot for Capturing Space Target," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China (2006) pp. 1008–1013.
27. S. Chiaverini, "Estimate of the two smallest singular values of the jacobian matrix: Application to damped least-squares inverse kinematics," *J. Rob. Syst.* **10**(8), 991–1008 (1993).
28. S. Chiaverini and B. Siciliano, "Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator," *IEEE Trans. Control Syst. Technol.* **2**(2), 123–134 (1994).
29. F. T. Cheng, T. L. Hour and Y. Y. Sun, "Study and resolution of singularities for a 6-Dof puma manipulator," *IEEE Trans. Syst., Man, Cybernet.–Part B: Cybernet.* **27**(2), 332–343 (1997).
30. W. F. Xu, B. Liang, Y. Liu, C. Li and W. Y. Qiang, "A novel approach of the singularity avoidance for puma-type manipulator," *Acta Automat. Sinica* **34**(6), 670–675 (2008).
31. J.-C. Piedboeuf, F. Aghili, M. Doyon, Y. Gonthier and E. Martin, "Dynamic Emulation of Space Robot in One-G Environment Using Hardware-in-the-Loop Simulation," *Proceedings of the 7th ESA Workshop on Advanced Space Technologies for Robotics and Automation, ESTEC*, Noordwijk, The Netherlands (2002).
32. O. Ma, J. G. Wang, S. Misra and M. Liu, "On the validation of Spdm task verification facility," *J. Rob. Syst.* **21**(5), 219–223 (2004).
33. K. Yoshida, H. Nakanishi, H. Ueno, N. Inaba, T. Nishimaki and M. Oda, "Dynamics, control, and impedance matching for robotic capture of a non-cooperative satellite," *RSJ Adv. Rob.* **18**(2), 175–198 (2004).
34. S. K. Agrawal, G. Hirzinger, K. Landzettel and R. Schwertassek, "A New Laboratory Simulator for Study of Motion of Free-floating Robots Relative to Space Targets", *IEEE Trans. Rob. Automat.* **12**(4), 627–633 (1996).
35. W. F. Xu, B. Liang, Y. S. Xu, C. Li and W. Y. Qiang, "A ground experiment system of free-floating space robot for capturing space target," *J. Intell. Rob. Syst.* **48**(2), 187–208 (2007).
36. L. F. Deng, Comparison of Image-Based and Position-Based Robot Visual Servoing Methods and Improvements *Ph.D. Dissertation* (Wateloo, Ontario, Canada: The University of Waterloo, 2003).

https://doi.org/10.1017/S0263574708004839 Published online by Cambridge University Press