AUTONOMOUS VISUAL SERVO ROBOTIC CAPTURE OF NON-COOPERATIVE TARGET

GANGQI DONG

A DISSERTATION SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN
EARTH AND SPACE SCIENCE AND ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO
SEPTEMBER 2016

Abstract

This doctoral research develops and validates experimentally a vision-based control scheme for the autonomous capture of a non-cooperative target by robotic manipulators for active space debris removal and on-orbit servicing. It is focused on the final capture stage by robotic manipulators after the orbital rendezvous and proximity maneuver being completed. Two challenges have been identified and investigated in this stage: the dynamic estimation of the non-cooperative target and the autonomous visual servo robotic control. First, an integrated algorithm of photogrammetry and extended Kalman filter is proposed for the dynamic estimation of the non-cooperative target because it is unknown in advance. To improve the stability and precision of the algorithm, the extended Kalman filter is enhanced by dynamically correcting the distribution of the process noise of the filter. Second, the concept of incremental kinematic control is proposed to avoid the multiple solutions in solving the inverse kinematics of robotic manipulators. The proposed target motion estimation and visual servo control algorithms are validated experimentally by a custom built visual servo manipulator-target system. Electronic hardware

for the robotic manipulator and computer software for the visual servo are custom designed and developed. The experimental results demonstrate the effectiveness and advantages of the proposed vision-based robotic control for the autonomous capture of a non-cooperative target. Furthermore, a preliminary study is conducted for future extension of the robotic control with consideration of flexible joints.

Dedication

To the memory of my grandmother.

Acknowledgements

I would like to extend my sincere appreciation to many people for their supports and encouragements, without which this dissertation would never have been possible.

First of all, I would like to express my heartfelt gratitude to my supervisor and mentor, Professor George Z. H. Zhu, for his guidance, encouragement and consistent support throughout my PhD study. Professor Zhu provides numerous constructive suggestions for my research without any reservation, and I benefit greatly from many inspiring discussions and conversations. His patient guidance, immense knowledge, unremitting efforts and rigorous academic attitude motivated me to progress not only academically but also personally.

I am grateful to my supervisor committee members, Professor Michael Daly and Professor Brendan Quine, for many valuable comments and encouragements in the annual meetings of my research evaluation, which inspired me to keep improving the quality of this doctoral research.

I would also like to thank my friends and group members, Dr. Rui Zhong,

Dr. Benoit P. Larouche, Dr. Guanghui Sun, Dr. Shen Gong, Dr. Yanfang Liu, Dr. Haichao Gui, Dr. Hao Wen, Dr. Banafsheh H. Pour, Gangqiang Li, Junjie Kang, Jian Zhang, Peng Li, Shi Lyu, Udai Bindra, Latheepan Murugathasan, and many others for companionship, rewarding academic exchanges and many interesting conversations. These are reasons that my past years are wonderful and full of memories.

Finally, my deepest gratitude goes to my families for the unconditional love, understanding and encouragement. Thanks to my parents for the everlasting love. Although we are far apart, your love is always my strongest support. Thanks to my wife for the consistent support and all the sacrifices. It has been a long journey that full of beauties and dangers which I am very grateful we have travelled together. Thanks to my lovely daughter and son, you are the source of my unprecedented happiness and my greatest pride forever. To my sister and brother, thank you for your infinite support to our family while I am far away for such a long time. Without your support I could never purely concentrate on my research.

Thank you everyone for your support during my PhD study, I wish you all the best.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

LEO	=	Low Earth Orbit
GEO	=	Geostationary Earth Orbit
OOS	=	On-Orbit Servicing
ISS	=	International Space Station
MER	=	Mars Exploration Rovers
ODS	=	Orbital Docking System
PES	=	Pure Experimental System
IBVS	=	Image-Based Visual Servo
PBVS	=	Position-Based Visual Servo
3D	=	Three Dimensional
KF	=	Kalman Filter
EKF	=	Extended Kalman Filter
AEKF	=	Adaptive Extended Kalman Filter
DH	=	Denavit-Hartenberg
FOV	=	Field of View
IPD	=	Interception-Point-Driven

TD	=	Target-Driven
CPR	=	Cycles per Revolution
CW	=	Clockwise
CCW	=	Counterclockwise
PCINT	=	Pin Change Interrupt
MFC	=	Microsoft Foundation Classes
ROI	=	Region of Interest

Chapter 1 INTRODUCTION AND JUSTIFICATION

Summary: In this chapter, we survey the application of space robotic manipulators, justify the research activities, define the research objectives, and outline the method of approach. At the end, we outline the layout of this dissertation and provide a full list of publications out of the doctoral study.

1.1 Backgrounds

The first manmade satellite, Sputnik 1, was launched into space by the former Soviet Union in 1957 and marked the beginning of the epochal space age. Following this significant footstep and along with the fast development of space exploration over the past few decades, there have been more and more satellites launched into space for navigation, communication, surveillance and remote sensing. There are around 6,600 satellites that have been launched into earth orbits and nearly a half are still in space [1]. In fact, the total number of operational satellites is only 1,381 (till 12/31/2015) [2], which means that roughly two thirds of the satellites in space are 'dead' and can be labeled as space debris. Including the space debris generated by the in-orbit explosions, collisions and other fragmentation events, the total number of the routinely tracked large[1] space debris by the United States Space Surveillance Network is around 23,000 in low Earth orbit (LEO) and geostationary Earth orbit (GEO) [3]. The continuously growing population of space debris in orbits poses severe threats to the safety of operational satellites as well as the long-term sustainability of space activities.

Over the past few decades, considerable efforts have been devoted to the space debris reduction and removal [4-15]. However, according to [4], no instance of debris removal has been successfully performed in space yet. In order to remediate the space environment, increasing attention of the international scientific community is attracted by disposing the whole or remnants of 'dead' satellites in space as well as extending the lifespan of operational satellites. These operations, including capture, installation, maintenance and repair work on an orbital object, are the so-called on-orbit servicing (OOS), which is an emerging and promising key technique in the future of space exploration.

To satisfy the capture demands for safety, efficiency and automation, robotic manipulators are increasingly adopted in OOS missions to perform dexterous operations [4]. Till now, robotic manipulators in space have been mainly performed in four fields: International Space Station (ISS), Mars

[1] Larger than 5-10cm in low Earth orbit and 30cm-1m in geostationary Earth orbit.

Exploration Rovers (MER), Orbital Docking System (ODS) and Pure Experimental System (PES) [16]. The Mobile Servicing System (MSS, or Canadarm2) [17], Japanese Experiment Module Remote Manipulator System (JEMRMS) [18] and European Robotic Arm (ERA) [19] are typical examples of space robotic manipulators performed on ISS. These robotic manipulators are mainly for assembly, maintenance, and payload exchange. The operations can be conducted autonomously as well as by astronauts. Robotic manipulators of MER, such as those mounted on the landers of Viking 1 and 2 [20], Spirit and Opportunity [21], Phoenix [22] and Curiosity [23] are used to dig into the ground of Mars, collect soil samples and/or position instruments on a target. These operations are performed by pre-programmed scripts when receiving the tele-operation commands sent from earth and/or relayed by the Mars Orbiter. Cameras mounted on these robotic manipulators are used to monitor the movements of the manipulators and take photographs of the surroundings. Robotic manipulators of ODS, such as the Shuttle Remote Manipulator System (SRMS, or Canadarm) [24] mounted on spacecraft, and the robotic manipulators mounted on Orbital Express [25], are used to perform grappling, docking, refueling, repairing and/or servicing of another orbiter. ROTEX and ETS-VII [26] are PES, operations such as assembling, grasping, docking and other orbit-replaceable-unit exchanging operations had been done to demonstrate the readiness and principles of ground control under time delay constraints.

A comparative study of selected space robotic manipulators is shown in Table 1.1 [27]. As can be seen in Table 1.1, the most of the robotic manipulators either adopt preprogrammed scripts or involve human-in-the-loop control, which lay the majority of decision-making and guidance tasks into human hands. Due to the facts of the longtime delay in earth-based manual control and the higher cost and risk of life loss in sending astronauts into space to control the capture process, the development of an autonomous control strategy for robotic capture of space objects becomes a research highlight in space robotic applications [28-32].

Table 1.1 Comparative study of space robotic manipulators

Mission	Date & Agency	Field	DOF	Vision System	Operation Approach	Comments
Landers of Viking 1 and 2	1975-1982 NASA	MER	120° radius	Two 360° cylindrical scan cameras	Pre-programmed and Tele-operation (from Earth, relayed by the Mar Orbiter)	The arm is with a collector head, temperature sensor, and magnet on the end, was used to dig into the ground and taking out samples of Martian soil.
Canadarm	1981-2011 CSA, NASA	ODS	6	One at the elbow joint and one at the wrist joint.	astronaut control	The Canadarm now retired, was a remote-controlled, payload-handling device used on the space shuttle.
ROTEX	04/1993-05/1993 DLR	PES	6	Two stereo cameras on the end effector, two fixed cameras	Tele-operation (on board or from ground) or pre-programmed	Three basic tasks were successfully performed: assembling a mechanical truss structure, connecting/disconnecting an electrical plug and grasping a floating object.
ETS-VII	1997-2002 NASDA (JAXA)	PES	6	Two hand cameras, two shoulder cameras.	Tele-operation (from Earth, 5-7 seconds of time delay)	The robotic arm was attached to the chaser satellite of ETS-II.
Canadarm2	2001-now CSA, NASA	ISS	7	Four TV cameras	Autonomous operation or astronaut control	Canadarm2 is a new generation Canadarm. Each end of the arm is equipped with a Latching End Effector.
Spirit and Opportunity	2004-now NASA	MER	5	One microscopic imager on the arm.	Pre-programmed and Tele-operation (from Earth, relayed by the Mar Orbiter)	The arm carries multiple instruments, and is used to position these instruments on a target. 1 Panoramic Cam, 1 Navigation Cam, 2 B&W Hazard Avoidance Cams mounted on the rover.

Mission	Date & Agency	Field	DOF	Vision System	Operation Approach	Comments
Orbital Express	03/2007-07/2007 DARPA, MSFC	ODS	6	One camera mounted at the end effector.	Autonomous (by executing pre-planned scripts)	Orbital Express is demonstrating the technologies required for on-orbit servicing. The arm is for grapple, dock, refuel, repair and service another satellite.
JEM Remote Manipulator System	2008-now JAXA	ISS	6	Two cameras mounted on the main arm.	Astronaut control	The Main Arm is used for exchanging EF (Exposed Facility) payloads. The SFA handles small items.
Phoenix	05/2008-11/2008 NASA	MER	4	One full-color camera attached above the scoop.	Pre-programmed and Tele-operation (from Earth, relayed by the Mar Orbiter)	The arm has the ability to dig down to 0.5 m below the surface.
European Robotic Arm (ERA)	2010-now (still in building) ESA	ISS	7	Four Camera and Lighting Units (CLU)	Autonomous operation or astronaut control	ERA can walk around the ISS under its own control, hand-over-hand between base points. Astronauts can control it from inside or outside the Station.
Curiosity (The lander of Mars Science Laboratory)	08/2012 (landed) NASA	MER	5	It has 17 cameras: HazCams (8), NavCams (4), MastCams (2), MAHLI (1), MARDI(1), ChemCam (1).	Pre-programmed and Tele-operation (from Earth, relayed by the Mar Orbiter)	One set of moves has never been tried before on Mars: pulling pulverized samples from the interior of Martian rocks and placing them into laboratory instruments inside the rover. The arm has two joints at the shoulder, one at the elbow and two at the wrist.

1.2 Justification of Research

Robotic manipulators, mechanically consisting of links and joints in sequence, are driven by motors at joints to achieve desired position/motion of the end-effector. Sensors, such as shaft encoders, force/torque gauges, and vision systems, are usually equipped in order to detect changes in the workspace and adjust the behaviors of the robotic manipulator accordingly [33]. Therefore, the robotic control generally concerns two issues: detection of the workspace and operation of the robotic manipulator.

1.2.1 Challenges of autonomous robotic capture

The task of autonomous capture of non-cooperative targets by robotic manipulators is not trivial. Many technical challenges arise and can be generally summarized as follows:

(i) ***Detection, estimation and modeling of the target***. A non-cooperative target is defined as a target that the motion of which is not controllable and no communication can be performed between the target and the robotic manipulator during the capture operation. Thus, accurate estimation and tracking of the target motion shall be carried out promptly once the non-cooperative targets or space debris are detected.

(ii) ***Path planning of the end-effector.*** The path of the end-effector shall be well-designed according to the estimated target motion to accomplish a safe capture. In addition, other constraints, such as, minimum time, shortest path, capture conditions, just to name a few, should be taken into account for a safe and efficient capture process.

(iii) ***Tracking control of the robotic manipulator.*** Since the desired position/motion of the end-effector is achieved by joint actuators of the robotic manipulator, inverse kinematics related issues such as multiple solutions arise. In addition, for the sake of high accuracy and high load capacity, stepper motors are considered as competitive candidates of joint actuator due to the movement of precise increments, which should also be taken into account by robotic control.

(iv) ***Real time capability***. The robotic capture of non-cooperative targets is a real time process. Thus, the first three challenges should be addressed in a real time environment and are required to interact with each other flawlessly.

(v) ***Autonomy.*** The non-cooperative nature of the target determines the timeliness of robotic capture strategies. Thus, for an autonomous robotic capture operation, decision making mechanisms should be dynamically adapted and updated.

1.2.2 Limitations of existing treatments

Robotic capture of non-cooperative targets attract much attention in the fields of robotics and space engineering [34-37]. Most of the existing target estimation methods in the literature are based on vision system [38-41] and can be generally classified [16] as: geometric methods [42-44], learning-based methods [45, 46], offline methods and filtering methods. As a typical example of the geometric method, photogrammetry is widely used if a calibrated camera is adopted and the geometry of the target features is known in advance [32]. Generally, the estimation process by the geometric method begins with an initial guess. Then a least squares approach is applied iteratively to estimate the pose information only. Furthermore, estimation by the geometric method is memoryless and purely based on the current measurement. Thus, it is prone to measurement noise and mechanical disturbance. The learning-based method is time-consuming and requires a sufficiently large database of the target in different poses, which is hard to satisfy especially when the target is non-cooperative. As the name implies, the offline method is employed afterwards, which is not applicable for real-time control. The filtering method, especially based on the Kalman filter (KF), is extensively adopted for estimation based on a set of noisy observations over time with the known initial conditions. Since the KF was proposed in 1960 [47], it has been adopted widely in almost all engineering fields, and has been extended from the original linear system to nonlinear systems with different enhancing techniques. A detailed survey of KF-based methods for robotic vision was presented in [48].

For nonlinear systems, feedback linearization based control methods are gaining popularity in modern system theory. The dynamics and kinematics of robotic manipulators are generally nonlinear. The existing robotic control methods can be classified as computed-torque-like or non-computed-torque-like [49]. As the name implies, the computed-torque-like robotic control method, such as, impedance control [50-56], is a special application of feedback linearization. It considers a predefined close-loop function as control objective and torques as control input of joint actuators. The limitation of control input arises in the implementation of many existing control methods when involving stepper motors, which is controlled by position. Other control methods, such as, adaptive control [57-59], robust control [60-64], learning control [65-69], etc., are often validated theoretically by simulations with specific assumptions and limitations due to complexity, which may be insufficient to guarantee the possibility of real time implementations.

1.3 Objectives of Proposed Research

To address the challenges and limitations, this research is focused on the development of robotic control strategy for autonomous capture of non-cooperative targets with the consideration of feasibility and reliability requirements for potential applications in space. Since the kinematic model of the target is not available, the estimation of target motion becomes the first challenge encountered. Secondly, robotic control schemes have to be developed in order to address the multiple solutions of inverse kinematics. Finally, the effectiveness and efficiency of the proposed control strategy have to be validated experimentally. Therefore, the objectives of this research are as follows:

(i) ***Vision-based target estimation*** — Development of algorithms for accurate estimation of position and motion of non-cooperative targets.

(ii) ***Robotic control theory*** — Development of innovative control theory for the autonomous robotic capture without multiple solutions in the inverse kinematics.

(iii) ***Validation*** — Validation of the effectiveness of the proposed vision-based target estimation algorithms and robotic control theory.

These objectives lead to an implementable vision-based robotic control scheme for the autonomous capture of a non-cooperative target.

1.4 Methodology of Approach

The methodology of approach adopted in this research begins with a brief literature review of visual servo robotic control. According to the identified challenges and limitations of existing approaches, target estimation is addressed first by a position-based vision system due to the intuitive relationship between the target and robotic manipulator in the Cartesian space.

Then the robotic control theory is developed with considerations of safety and efficiency to address the limitations, such as, inverse kinematics, lack of feedback information, temporal effectiveness of target estimates, control of actuators by precise increments, etc. Finally, supporting hardware and computer software are developed to upgrade an existing customized manipulator-target system in order to validate the feasibility and reliability of the proposed approaches experimentally.

1.5 Outline

The dissertation is manuscript-based because almost all the works are either published by the peer-reviewed journals and conference proceedings or submitted to the peer-reviewed journals for consideration of publication. It includes seven chapters. Following the introduction and justification in Chapter 1, a detailed literature review of the visual servo robotic control is conducted in Chapter 2. Chapter 3 presents the development of vision-based estimation algorithm of non-cooperative targets, while Chapter 4 focuses on the development of robotic control strategies. The hardware and software development of the testbed for the validation of the proposed visual servo control scheme is described in Chapter 5. To address the influence of flexible joint on the overall performance of the robotic manipulator discovered in the experiments, the modeling and control of flexible joint robotic manipulator is investigated in Chapter 6. Finally, the contributions of this doctoral research are summarized and the future research directions are outlined in Chapter 7. The full-text of the published and submitted manuscripts have been attached at the end of this dissertation. The theories and experimental results of the target estimation developed in Chapter 3 have been published in reference paper A and B. The concepts and hardware-in-the-loop simulation results of the kinematics-based robotic control developed in Chapter 4 have been published in reference paper C and D. Then the experimental results of the autonomous visual servo robotic capture of non-cooperative target based on the experimental systems described in Chapter 5 have been included in the submitted manuscripts, reference paper E and F. In all the attached six manuscripts, the first author is the main contributor and the second author is the corresponding author.

1.6 List of Publications

The following is a full list of publications associated with this study.

1.6.1 Published and submitted peer-reviewed journal papers

1. **(Reference Paper A) G. Dong** and Z. H. Zhu. "Position-based visual servo control of autonomous robotic manipulators". ***Acta Astronautica***, Vol. 115, pp. 291-302, 2015.

   doi:10.1016/j.actaastro.2015.05.036

2. **(Reference Paper B) G. Dong** and Z. H. Zhu. “Autonomous Robotic Capture of Non-cooperative Target by Adaptive Extended Kalman Filter Based Visual Servo”. ***Acta Astronautica***, Vol. 122, pp. 209-218, 2016.

   doi:10.1016/j.actaastro.2016.02.003

3. **(Reference Paper C) G. Dong** and Z. H. Zhu. “Incremental Inverse Kinematics based Vision Servo for Autonomous Robotic Capture of Non-Cooperative Space Debris”. ***Advances in Space Research***, Vol. 57(7), pp. 1508-1514, 2016.

   doi:10.1016/j.asr.2016.01.011

4. **(Reference Paper D) G. Dong** and Z. H. Zhu. “Incremental Visual Servo Control of Robotic Manipulator for Autonomous Capture of Non-cooperative Target”. ***Advanced Robotics***, Vol. 30(22), pp. 1458-1465, 2016.

   doi:10.1080/01691864.2016.1229633

5. **(Reference Paper E) G. Dong** and Z. H. Zhu. "Predictive Visual Servo Kinematic Control for Autonomous Robotic Capture of Non-cooperative Target". ***Autonomous Robots***, submitted on April 02, 2016.

6. **(Reference Paper F) G. Dong** and Z. H. Zhu. “A Kinematics-based Incremental Visual Servo for Robotic Capture of Non-cooperative Target”. ***IEEE/ASME Transactions on Mechatronics***, submitted on June 03, revised form submitted on August 05, 2016.

1.6.2 International conference papers

1. Z. H. Zhu and **G. Dong**. “Visual Servo Robotic Capture of Non-Cooperative Target in Active Space Debris Removal”. Canadian Aeronautics and Space Institute ASTRO 2016, May 2016, Ottawa.
2. **G. Dong** and Z. H. Zhu. “Autonomous Robotic Capture of Non-cooperative Target by Vision-based Kinematic Control”. *AIAA SPACE 2015 Conference and Exposition*, Aug. 2015, Pasadena, California. doi:10.2514/6.2015-4428
3. **G. Dong** and Z. H. Zhu. “Visual-Servo Autonomous Robotic Manipulators for Capturing Non-Cooperative Target”, *ASME 2014 International Mechanical Engineering Congress and Exposition*, Nov. 2014, Montreal. doi:10.1115/IMECE2014-38574
4. **G. Dong** and Z. H. Zhu. “Vision-based Autonomous Control of Space Robotic Manipulators”, *65th International Astronautical Congress (IAC)*, Oct. 2014, Toronto. IAC-14,D1,P,11,x23034
5. **G. Dong** and Z. H. Zhu. “Vision-based Pose and Motion Estimation of Non-cooperative Target for Space Robotic Manipulators”. *AIAA SPACE 2014 Conference and Exposition*, Aug. 2014, San Diego. doi:10.2514/6.2014-4263
6. **G. Dong** and Z. H. Zhu. “Input Shaping-based Impedance Control for

Flexible Joint Robotic Manipulator". *AIAA Guidance, Navigation, and Control Conference*, Aug. 2013, Boston. doi:10.2514/6.2013-4849

7. **G. Dong** and Z. H. Zhu. "Modelling and Control of Flexible Joint Robotic Manipulator". *AIAA Infotech@Aerospace Conference*, Aug. 2013, Boston. doi:10.2514/6.2013-4824

1.6.3 Academic bulletin paper

1. **G. Dong** and Z. H. Zhu. "Control of Space Robotic Manipulators with Joint Flexibility". *Canadian Society for Mechanical Engineering (CSME) Summer Bulletin*, pp11-16, 2013.

   Available at: *http://www.csme-scgm.ca/sites/all/themes/csme/uploaded/CSME_publications/Summer_2013_bulletin.pdf*

Chapter 2 LITERATURE REVIEW

Summary: In this chapter, we review the literature of visual servo robotic control in the two main research areas of this dissertation: vision systems and robotic control theory. Based on the literature review, the candidate methodology is suggested.

2.1 Vision System

Sensors are widely adopted to detect the behaviors of targets as well as robotic manipulators within the workspace. Due to the fact that space debris is in uncontrolled motion, the estimation of such dynamic non-cooperative targets becomes one of the major prerequisites in robotic capture. According to the types of feedback sensors, robotic control can be classified as vision-based control, force control and tactile control [33]. Because of the non-intrusive/contact nature, vision systems have been extensively adopted in the robotic control for detecting, monitoring, and tracking purpose [70]. Therefore, vision-based control is adopted in the current research and this review is conducted in this field.

In early robotic applications, vision is adopted in an open-loop fashion, which is known as static look-and-move method [71]. In order to improve the accuracy and extend vision-based applications to real time, the visual information is fed back in closed-loop robotic control called visual servo in the literature [72, 73]. According to the errors adopted in the control, visual servo can be categorized as: image-based visual servo (IBVS), position-based visual servo (PBVS), and hybrid visual servo.

In the IBVS, the control error between the desired and observed image features in the 2D (two dimensional) image space is fed into the control via an image Jacobian. Therefore, the IBVS is independent of the target model. However, the IBVS lacks the depth information, considerable efforts have been devoted to improve the performance and enhance the robustness in 3D (three dimensional) applications [74-77].

On the contrary, the PBVS extracts 3D pose and motion of the target features from the vision system, and the error between desired and estimated pose and motion is fed back to the control. Accordingly, the end-effector can be controlled relative to the target naturally. Since the pose and motion are estimated based on the vision feedback, the PBVS is prone to camera calibration errors, target model accuracy, and image measurement noise. The errors caused by uncalibrated cameras are addressed in [78, 79], and the suppression of image noise issue is discussed in [80].

Finally, the hybrid visual servo, referred as 2½D (two and a half dimensional) visual servo in the literature [81], evaluates the control errors partially in the 3D workspace and partially in the 2D image plane. Although effective, the hybrid visual servo is more complex than either IBVS or PBVS for real time implementation.

The camera in a vision system can be either mounted on the robotic manipulator, known as eye-in-hand, or fixed to the workspace, known as eye-to-hand, alternatively. The eye-in-hand configuration may provide a close and detailed scene of the target yet be coupled with the motion of the robotic manipulator, while the eye-to-hand configuration monitors the whole workspace but fewer details of the target will be detected. In some specific circumstances during robotic operations, the view of the target may be blocked by the robotic manipulator in the eye-to-hand configuration, and may further lead to task failures. In contrast, the coupling effect in the eye-in-hand configuration could be easily eliminated if the motion of the robotic manipulator is available. In addition, the accuracy and detail of the target will keep increasing in the eye-in-hand configuration as the end-effector approaches the target.

2.2 Robotic Control

Once the essential information of the workspace is detected, the operation of the robotic manipulator to fulfill desired tasks can be effectively and efficiently carried out under certain control schemes. The development of control schemes involves the dynamics and/or kinematics of the robotic manipulator.

The dynamics, obtained by the Euler-Lagrange approach, define the relationship between the force/torque input of the joint actuators and the motion output of the robotic manipulator. The kinematics, derived from the geometry of the robotic manipulator, define the mapping of states between the robotic manipulator in joint space and the end-effector in the Cartesian space. Generally, without the consideration of obstacle avoidance, the desired position and/or motion of the end-effector in the Cartesian space is regarded as the control objective of the robotic manipulator. Therefore, inverse kinematics shall be performed in advance to transform the desired position and motion of the end-effector in the Cartesian space into the corresponding position and motion of the robotic manipulator in the joint space to derive the control input (i.e., torques). Although inverse kinematics in the velocity-level and acceleration-level can be derived based on the pseudo inverse of the robotic Jacobian matrix, the position-level inverse kinematics may have multiple solutions due to the periodicity of trigonometric functions and the redundant geometric configuration of the robotic manipulator. In order to obtain a unique solution, extra constraints have to be introduced. For instance, a constrained least squares approach was introduced for the inverse kinematics of redundant space manipulators in order to locally minimize the dynamical coupling of the manipulator-spacecraft system [82]. Based on the gradient dynamics method, an acceleration-level solution was introduced to solve the inverse kinematics in order to avoid the pseudo inverse of the Jacobian and reduce the computational cost [83]. To satisfy the robustness demands of kinematics based robotic control with the occurrence of singularities, a task priority redundancy resolution technique was developed to avoid the inversion of the matrix which is close to the singularity point [84].

In space applications, the robotic manipulators are generally considered as free floating unless their masses are much smaller than the base spacecraft. The dynamic coupling between the robotic manipulator and the base spacecraft in the free-floating case complicates the dynamics and kinematics analysis of the manipulator-spacecraft system. In order to address this difficulty, the concept of resolved motion rate control was introduced to establish a control method for the manipulator-spacecraft system [85], in which the momentum conservation law was introduced to derive the generalized Jacobian matrix. The concept of a virtual manipulator was proposed for the modelling of free floating manipulators in 1987 [86]. A virtual manipulator is an idealized massless kinematic chain whose base is fixed to the mass center of the system and whose end is overlapped with the end-effector of the real robotic manipulator. Due to the external forces/torques exerting on the system being negligible in the free floating case, the base of a virtual manipulator is a fixed point in the inertial space. Therefore, the virtual manipulator can be used for simulation analysis of the dynamics and kinematics of a free floating manipulator-spacecraft system. Furthermore, the concept of a dynamically equivalent manipulator enables the mapping of dynamics and kinematics of a free floating manipulator to a conventional fixed base manipulator for experimental study on ground [87].

In addition, both the joint and link flexibilities have also been extensively studied in the literature of robotic manipulator control [88, 89]. The concept of virtual rigid link was introduced to derive the dynamic model of a flexible-link manipulator in [90]. Then, the virtual rigid manipulator was introduced for the control of a flexible-link manipulator [91]. Flexible joint robotic systems have been studied in the robotics literature for more than twenty years. To address the control problem of flexible joint robotic manipulators, nonlinear adaptive output feedback control is adopted to control the quasi-steady-state robot model in [92]. A regressor-free adaptive controller for flexible joint robot has been developed [57, 58, 93]. However, the assumption that the joint flexibility should be known is usually not available. Other techniques such as nonlinear H_∞ control [94] and the combination of neural networks and linear regulators [95] are difficult to implement. Input shaping is an open-loop vibration suppressor by creating a command signal that cancels its own vibration [96-101]. It is particularly effective for suppressing motion-induced vibration in flexible systems. However, the open-loop structure of input shaping is prone to environmental disturbances. Closed-loop input shaping is good at disturbance rejection, but it requires exact knowledge of the system to be controlled. Using approximate models may reduce the effectiveness of vibration suppression or even destabilize the system.

2.3 Proposed Methodology

The PBVS is adopted for the robotic control due to the intuitive relationship between the target and the robotic manipulator in the 3D Cartesian space. As per [48], the extended Kalman filter (EKF) is the most widely adopted nonlinear state estimation algorithm in the robotic vision applications. The EKF is a first order linearization based filtering method. It may lead to particularly poor performance, especially over a long time period, due to the linearizing approximation of the process and/or measurement models with high nonlinearity. In order to improve the performance when highly nonlinear models are involved, adaptive mechanisms, i.e., an adaptive extended Kalman Filter (AEKF), should be introduced to the distribution of noises as a compensation of the linearization error [40, 102, 103]. Although the process and measurement noises are usually assumed to be independent, it is hard to distinguish them in the process covariance matrix from the measurement covariance matrix [104]. Thus, the AEKF that attempts to correct both covariance matrices may not be robust [105]. It is well known that KF-based algorithms require initial conditions[2] and measurements over the time. Although the initial conditions do not affect the convergence property of KF-based algorithms, they do affect the performance of the estimation. For a non-cooperative target, the initial conditions and the kinematic model of the target are unknown in advance. To address this challenge, the current work combines the KF based algorithms with other estimation approach, such as photogrammetry, for better performance.

In practice, many robotic manipulators, especially those performed in space, adopted stepping motors as the joint actuators for the sake of high load capacity and positioning accuracy, where joint position is considered as the control input of the robotic manipulator. The load capacity of a robotic manipulator is considered sufficient by properly sizing the stepper motors and the torque limits are not an issue. Thus, it is relatively intuitive to consider the speed limits as the main constraints of the actuators in the current work. With this in mind, it is reasonable and acceptable to consider the robotic control from the kinematics aspect.

[2] Initial values of the states, distribution of noises, covariance of the models, etc.

Chapter 3 VISION-BASED TARGET ESTIMATION

Summary: In this chapter, the pinhole camera model is introduced and the vision-based target estimation approaches, such as photogrammetry, EKF and AEKF are developed accordingly. The target estimation approaches have been validated experimentally and the experimental results have been published in reference A and reference B.

3.1 Pinhole Camera Model

The vision-based estimation is based on the pinhole camera model. For the ease of description and without loss of generality, the global frame, denoted by $\mathcal{F}_G$, is fixed in the inertia space. The camera frame, denoted by $\mathcal{F}_C$, is fixed to the image plane center of the camera. The target frame, denoted by $\mathcal{F}_T$, is fixed to the target body. The position and orientation of $\mathcal{F}_T$ described in $\mathcal{F}_C$ are denoted by ${}^{C}\mathbf{X}_{To} = \left\{ {}^{C}x_{To}, {}^{C}y_{To}, {}^{C}z_{To} \right\}^T$ and ${}^{C}\mathbf{\Omega}_T = \left\{ {}^{C}\theta_{Tx}, {}^{C}\theta_{Ty}, {}^{C}\theta_{Tz} \right\}^T$, respectively. Further assume the non-cooperative target contains κ identified feature points in the field of view (FOV) of the camera. The positions of the feature points in $\mathcal{F}_T$ are denoted by ${}^{T}\mathbf{X}_i = \left\{ {}^{T}x_i, {}^{T}y_i, {}^{T}z_i \right\}^T$, where the subscript $i = 1, 2, \cdots, \kappa$. Based on the above definitions and assumptions, the position of *i-th* feature point in $\mathcal{F}_C$, denoted by ${}^{C}\mathbf{X}_i = \left\{ {}^{C}x_i, {}^{C}y_i, {}^{C}z_i \right\}^T$, can be expressed as

$$ {}^{C}\mathbf{X}_i = \mathcal{R}_{T,C}\,{}^{T}\mathbf{X}_i + {}^{C}\mathbf{X}_{To} \tag{3.1} $$

where $\mathcal{R}_{T,C} \in \mathbf{R}^{3\times3}$ is the rotation matrix from $\mathcal{F}_T$ to $\mathcal{F}_C$ formed by the trigonometric functions of elements in ${}^{C}\mathbf{\Omega}_T$. It should be noted that both ${}^{C}\mathbf{X}_{To}$ and ${}^{C}\mathbf{\Omega}_T$ in Eq. (3.1) are unknown and need to be estimated.

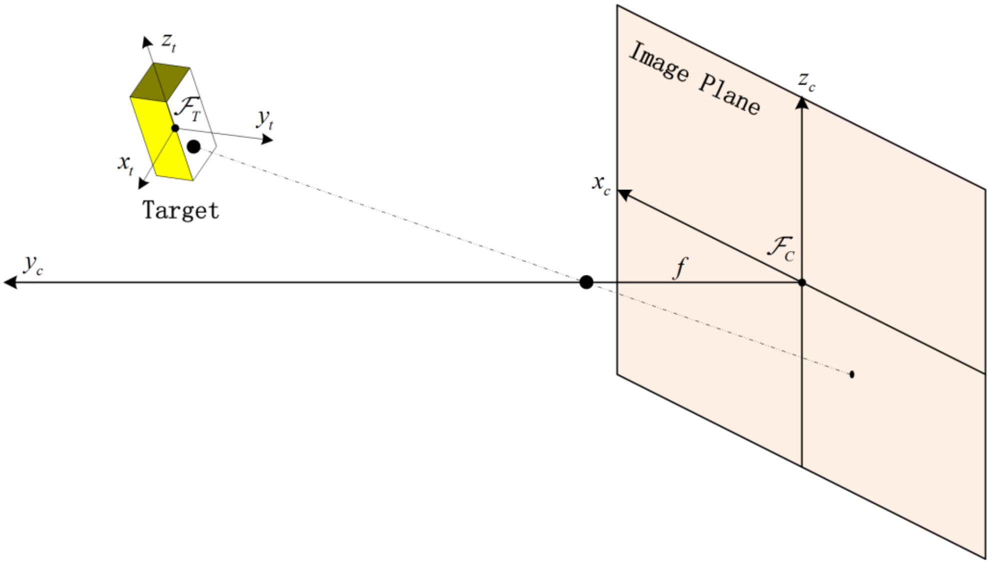


Figure 3.1 Illustration of the pinhole camera projection

As shown in Figure 3.1, the feature point is projected onto the image plane of the pinhole camera. Assume the image measurements of the *i-th* feature point, denoted by $\mathbf{Z}_i = \left\{ {}^{M}x_i, {}^{M}z_i \right\}^T$, can be obtained by image processing. According to the collinearity condition, the pinhole camera projection of the *i-th* feature point can be described by [106]

$$\left\{{}^{M}x_i, {}^{M}z_i\right\}^{T} = -\frac{f}{{}^{C}y_i - f}\left\{{}^{C}x_i, {}^{C}z_i\right\}^{T} \tag{3.2}$$

where f denotes the focal length of the camera. A generalized form of Eq. (3.2) is concisely written as

$$\mathbf{Z}_i = \mathcal{P}\left({}^{C}\mathbf{X}_i, f\right) \tag{3.3}$$

Substituting Eq. (3.1) into Eq. (3.3), we obtain the expression of image measurements in 2D, with respect to the corresponding feature points in $\mathcal{F}_T$, in terms of the six degrees of freedom — the unknown position and orientation of $\mathcal{F}_T$ described in $\mathcal{F}_C$, such that

$$\mathbf{Z}_i = \mathcal{P}\left(\mathcal{R}_{T,C}\,{}^{T}\mathbf{X}_i + {}^{C}\mathbf{X}_{To}, f\right) \tag{3.4}$$

3.2 Photogrammetry

In order to estimate the unknown position and orientation of $\mathcal{F}_T$ described in $\mathcal{F}_C$, photogrammetry [32], an iterative least squares based approach, is introduced in this section.

Rearranging Eq. (3.2) and defining functions F_i and G_i for concise expression, such that

$$\begin{cases} 0 = {}^{M}x_i\,{}^{C}y_i - {}^{M}x_i f + {}^{C}x_i f \triangleq F_i \\ 0 = {}^{M}z_i\,{}^{C}y_i - {}^{M}z_i f + {}^{C}z_i f \triangleq G_i \end{cases} \tag{3.5}$$

Since ${}^{C}\mathbf{X}_i = \left\{ {}^{C}x_i, {}^{C}y_i, {}^{C}z_i \right\}^T$ is function of ${}^{C}\mathbf{X}_{To}$ and ${}^{C}\mathbf{\Omega}_T$ as per Eq. (3.1), F_i and G_i are functions of elements in ${}^{C}\mathbf{X}_{To}$ and ${}^{C}\mathbf{\Omega}_T$ as well. Linearizing F_i and G_i by Taylor's expansion in the vicinity of an arbitrary initial guess of unknown variables leads to

$$\begin{cases} 0 = F_i \approx F_{i0} + \left(\dfrac{\partial F_i}{\partial {}^{C}x_{To}} \right)_0 d\,{}^{C}x_{To} + \left(\dfrac{\partial F_i}{\partial {}^{C}y_{To}} \right)_0 d\,{}^{C}y_{To} + \left(\dfrac{\partial F_i}{\partial {}^{C}z_{To}} \right)_0 d\,{}^{C}z_{To} \\ \qquad + \left(\dfrac{\partial F_i}{\partial {}^{C}\theta_{Tx}} \right)_0 d\,{}^{C}\theta_{Tx} + \left(\dfrac{\partial F_i}{\partial {}^{C}\theta_{Ty}} \right)_0 d\,{}^{C}\theta_{Ty} + \left(\dfrac{\partial F_i}{\partial {}^{C}\theta_{Tz}} \right)_0 d\,{}^{C}\theta_{Tz} \\ 0 = G_i \approx G_{i0} + \left(\dfrac{\partial G_i}{\partial {}^{C}x_{To}} \right)_0 d\,{}^{C}x_{To} + \left(\dfrac{\partial G_i}{\partial {}^{C}y_{To}} \right)_0 d\,{}^{C}y_{To} + \left(\dfrac{\partial G_i}{\partial {}^{C}z_{To}} \right)_0 d\,{}^{C}z_{To} \\ \qquad + \left(\dfrac{\partial G_i}{\partial {}^{C}\theta_{Tx}} \right)_0 d\,{}^{C}\theta_{Tx} + \left(\dfrac{\partial G_i}{\partial {}^{C}\theta_{Ty}} \right)_0 d\,{}^{C}\theta_{Ty} + \left(\dfrac{\partial G_i}{\partial {}^{C}\theta_{Tz}} \right)_0 d\,{}^{C}\theta_{Tz} \end{cases} \tag{3.6}$$

where F_{i0} and G_{i0} are evaluated at the current image measurements of the *i-th* feature point and the initial guess of the unknown variables.

For a concise format, Eq. (3.6) is rewritten as

$$[\mathbf{V}_0]_{2\times 1} = [\mathbf{J}_\mathbf{V}]_{2\times 6} [\mathbf{\Delta}]_{6\times 1} \tag{3.7}$$

where the subscripts indicate the dimensions of matrices enclosed by square brackets and the matrices are defined as

$$\mathbf{V}_0 = -\left\{ F_{i0}, G_{i0} \right\}^T, \quad \mathbf{\Delta} = \left\{ d\,{}^{C}x_{To}, d\,{}^{C}y_{To}, d\,{}^{C}z_{To}, d\,{}^{C}\theta_{Tx}, d\,{}^{C}\theta_{Ty}, d\,{}^{C}\theta_{Tz} \right\}^T,$$

$$\mathbf{J}_\mathbf{V} = \begin{bmatrix} \left(\dfrac{\partial F_i}{\partial {}^{C}x_{To}} \right)_0 & \left(\dfrac{\partial F_i}{\partial {}^{C}y_{To}} \right)_0 & \left(\dfrac{\partial F_i}{\partial {}^{C}z_{To}} \right)_0 & \left(\dfrac{\partial F_i}{\partial {}^{C}\theta_{Tx}} \right)_0 & \left(\dfrac{\partial F_i}{\partial {}^{C}\theta_{Ty}} \right)_0 & \left(\dfrac{\partial F_i}{\partial {}^{C}\theta_{Tz}} \right)_0 \\ \left(\dfrac{\partial G_i}{\partial {}^{C}x_{To}} \right)_0 & \left(\dfrac{\partial G_i}{\partial {}^{C}y_{To}} \right)_0 & \left(\dfrac{\partial G_i}{\partial {}^{C}z_{To}} \right)_0 & \left(\dfrac{\partial G_i}{\partial {}^{C}\theta_{Tx}} \right)_0 & \left(\dfrac{\partial G_i}{\partial {}^{C}\theta_{Ty}} \right)_0 & \left(\dfrac{\partial G_i}{\partial {}^{C}\theta_{Tz}} \right)_0 \end{bmatrix}.$$

It should be noted that Eq. (3.7) is for the *i-th* feature point. If κ feature points are known, Eq. (3.7) can be easily extended as

$$[\mathbf{V}_0]_{2\kappa\times 1} = [\mathbf{J}_\mathbf{V}]_{2\kappa\times 6}[\mathbf{\Delta}]_{6\times 1} \tag{3.8}$$

Generally, for each of the feature points, two equations with six unknowns can be obtained. Therefore, a minimum of three feature points, i.e., $\kappa=3$, should be available for solving the six unknowns. In order to eliminate the ambiguity caused by the periodic solutions of trigonometric functions, four feature points, i.e., $\kappa=4$, are widely adopted in the literature [106, 107]. Solving Eq. (3.8) based on the definition of pseudo inverse, we have the correction of the unknown variables as

$$\mathbf{\Delta} = \left(\mathbf{J}_\mathbf{V}{}^T\mathbf{J}_\mathbf{V}\right)^{-1}\mathbf{J}_\mathbf{V}{}^T\mathbf{V}_0 \tag{3.9}$$

The correction obtained by Eq. (3.9) is added to the initial guess of unknown variables in order to form a new guess. Then the procedure is iteratively performed to correct previous guess until the correction is less than a pre-defined tolerance.

The main shortcoming of the photogrammetry is that it does not estimate the motion directly, which is regarded as an important information for robotic control in a dynamic environment. Furthermore, because of purely relying on the current measurements, the photogrammetry is memoryless to the historical measurements and thus prone to the image noise. In addition, due to iteration, the computational cost of photogrammetry varies, which is not desirable in real time control systems. To address these challenges, the estimation algorithm of photogrammetry is combined with the filtering methods described as follows.

3.3 EKF Estimation

The EKF requires a priori knowledge of state variables, the process model and measurement model for the target estimation. For simplicity, we assume the unknown variables to be estimated are the position and orientation of the $\mathcal{F}_T$ described in $\mathcal{F}_C$ as well as their first order derivatives.

Define the state variable as

$$\mathbf{S}=\left\{{}^{C}x_{To},{}^{C}\dot{x}_{To},{}^{C}y_{To},{}^{C}\dot{y}_{To},{}^{C}z_{To},{}^{C}\dot{z}_{To},{}^{C}\theta_{Tx},{}^{C}\dot{\theta}_{Tx},{}^{C}\theta_{Ty},{}^{C}\dot{\theta}_{Ty},{}^{C}\theta_{Tz},{}^{C}\dot{\theta}_{Tz}\right\}^{T} \qquad (3.10)$$

By neglecting the third and higher order time derivatives and considering the second order time derivatives as process noise, a linear process model is adopted, which can be written as

$$\mathbf{S}_{k+1}=\mathbf{A}\mathbf{S}_{k}+\mathbf{B}\boldsymbol{\omega} \qquad (3.11)$$

where the subscripts k and $k+1$ indicate the current and next sampling steps, $\mathbf{A}$ and $\mathbf{B}$ are the coefficient matrices formed by functions of sampling time, $\boldsymbol{\omega}$ is the process noise obeying the normal distribution with constant mean $\mathbf{q}$ and covariance $\mathbf{Q}$, such that

$$\boldsymbol{\omega}=\left\{{}^{C}\ddot{x}_{To},{}^{C}\ddot{y}_{To},{}^{C}\ddot{z}_{To},{}^{C}\ddot{\theta}_{Tx},{}^{C}\ddot{\theta}_{Ty},{}^{C}\ddot{\theta}_{Tz}\right\}^{T},\quad \boldsymbol{\omega}\sim\mathcal{N}\left(\mathbf{q},\mathbf{Q}\right) \qquad (3.12)$$

From Eq. (3.4), the measurement model is defined as

$$\mathbf{Z}_k = \mathcal{H}\left(\mathbf{S}_k\right) + \boldsymbol{\mu}, \quad \mathcal{H}\left(\mathbf{S}_k\right) = \mathcal{P}\left(\mathcal{R}_{T,C}{}^{T}\mathbf{X}_i + {}^{C}\mathbf{X}_{To}, f\right) \tag{3.13}$$

where $\boldsymbol{\mu}$ stands for the measurement noise of the camera and is assumed to obey the normal distribution with zero mean and constant covariance matrix, i.e., $\boldsymbol{\mu} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}\right)$.

Based on the process model defined in Eq. (3.11), the state variable $\mathbf{S}$ and its covariance matrix $\mathbf{P}$ in the next step can be predicted by

$$\mathbf{S}_{k+1|k} = \mathbf{A}\mathbf{S}_{k|k} + \mathbf{B}\mathbf{q} \tag{3.14}$$

$$\mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^T + \mathbf{B}\mathbf{Q}\mathbf{B}^T \tag{3.15}$$

where the subscript $\alpha|\beta$ denotes the variables estimated at step α with respect to step β.

The Kalman gain is then given by

$$\mathbf{K}_g = \mathbf{P}_{k+1|k}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k+1|k}\mathbf{H}_k^T + \mathbf{R}\right)^{-1} \tag{3.16}$$

where $\mathbf{H}_k = \partial\mathcal{H}\left(\mathbf{S}\right)/\partial\mathbf{S}\big|_{\mathbf{S}=\mathbf{S}_{k+1|k}}$ denotes the Jacobian of the measurement model.

Once the current measurement $\mathbf{Z}_k$ is obtained, the predictions made by Eq. (3.14) and Eq. (3.15) are updated, such that

$$\mathbf{S}_{k+1|k+1} = \mathbf{S}_{k+1|k} + \mathbf{K}_g\left(\mathbf{Z}_k - \mathcal{H}\left(\mathbf{S}_{k+1|k}\right)\right) \tag{3.17}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_g\mathbf{H}_k\mathbf{P}_{k+1|k} \tag{3.18}$$

By iterating the predict-update procedure defined by Eq. (3.14) – Eq.

(3.18), the state variable is estimated for each measurement obtained. Since the initial conditions are not available, the EKF algorithm is initialized by the photogrammetry in order to accelerate the convergence rate.

3.4 AEKF Estimation

The AEKF compensates the linearization error of EKF by introducing an adaptive mechanism to estimate and update distribution parameters of the noises. Since the measurement noise of the camera can be determined in advance experimentally, only the distribution of process noise is adapted. Accordingly, the process model and noise distribution is modified slightly based on Eq. (3.11), such that

$$\mathbf{S}_{k+1} = \mathbf{A}\mathbf{S}_k + \mathbf{B}\boldsymbol{\omega}_k, \quad \boldsymbol{\omega}_k \sim \mathcal{N}\left(\mathbf{q}_k, \mathbf{Q}_k\right) \tag{3.19}$$

Once initialized, following similar predict-update procedure defined by Eq. (3.14) – Eq. (3.18) yields

$$\begin{cases} \mathbf{S}_{k+1|k} = \mathbf{A}\mathbf{S}_{k|k} + \mathbf{B}\mathbf{q}_k \\ \mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^T + \mathbf{B}\mathbf{Q}_k\mathbf{B}^T \\ \mathbf{K}_g = \mathbf{P}_{k+1|k}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k+1|k}\mathbf{H}_k^T + \mathbf{R}\right)^{-1} \\ \mathbf{S}_{k+1|k+1} = \mathbf{S}_{k+1|k} + \mathbf{K}_g\left(\mathbf{Z}_k - \mathcal{H}\left(\mathbf{S}_{k+1|k}\right)\right) \\ \mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_g\mathbf{H}_k\mathbf{P}_{k+1|k} \end{cases} \tag{3.20}$$

In order to estimate and update the distribution of process noise for the next iteration, an intuitive approximation of the mean is defined as [102]

$$\hat{\mathbf{q}}_{j+1} = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\left(\mathbf{S}_{j+1|j+1} - \mathbf{A}\mathbf{S}_{j|j}\right) \tag{3.21}$$

By further assuming that the process noise is independent and uniformly distributed over N sampling steps, the unbiased estimate of the mean and corresponding covariance matrix can be calculated by

$$\mathbf{q}_k = \frac{1}{N}\sum_{j=k-N+1}^{k}\hat{\mathbf{q}}_j = \mathbf{q}_{k-1} + \frac{1}{N}\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N}\right) \tag{3.22}$$

$$\mathbf{Q}_k = \mathbf{Q}_{k-1} + \frac{1}{N-1}\begin{pmatrix} \left(\hat{\mathbf{q}}_k - \mathbf{q}_k\right)\left(\hat{\mathbf{q}}_k - \mathbf{q}_k\right)^T - \left(\hat{\mathbf{q}}_{k-N} - \mathbf{q}_k\right)\left(\hat{\mathbf{q}}_{k-N} - \mathbf{q}_k\right)^T \\ + \frac{1}{N}\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N}\right)\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N}\right)^T + \frac{N-1}{N}\left(\mathbf{\Lambda}_{k-N} - \mathbf{\Lambda}_k\right) \end{pmatrix} \tag{3.23}$$

where $\mathbf{\Lambda}_k = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\left(\mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^T - \mathbf{P}_{k+1|k+1}\right)\mathbf{B}\left(\mathbf{B}^T\mathbf{B}\right)^{-1}$.

The adapted distribution of the process noise in Eq. (3.22) and Eq. (3.23) is then adopted for the next iteration in Eq. (3.20).

3.5 Coordinate System Transformation

It should be noted that the estimated position and velocity of the target, ${}^C\mathbf{X}_{To}$ and ${}^C\dot{\mathbf{X}}_{To}$, are in $\mathcal{F}_C$. Define $\mathbf{X}_T$ and $\dot{\mathbf{X}}_T$ as the position and velocity of the target in $\mathcal{F}_G$, such that

$$\mathbf{X}_T = \mathcal{R}_{C,G}\,{}^C\mathbf{X}_{To} + {}^G\mathbf{X}_{Co} \tag{3.24}$$

where ${}^G\mathbf{X}_{Co}$ denotes the position of $\mathcal{F}_C$ described in $\mathcal{F}_G$, and $\mathcal{R}_{C,G}$ denotes the rotational matrix from $\mathcal{F}_C$ to $\mathcal{F}_G$.

Since the configuration of the camera is known, ${}^{G}\mathbf{X}_{Co}$ and $\mathcal{R}_{C,G}$ should be easily obtained based on the configuration of the robotic system. It is worth pointing out that for the eye-in-hand configuration, ${}^{G}\mathbf{X}_{Co}$ and $\mathcal{R}_{C,G}$ are formed by functions of joint configuration, while for the eye-to-hand configuration, both ${}^{G}\mathbf{X}_{Co}$ and $\mathcal{R}_{C,G}$ are constant. By defining

$$\mathcal{T} = \begin{bmatrix} \mathcal{R}_{C,G} & {}^{G}\mathbf{X}_{Co} \\ \mathbf{O} & 1 \end{bmatrix} \tag{3.25}$$

Equation (3.24) can be rewritten in a homogeneous format, such as

$$\begin{Bmatrix} \mathbf{X}_T \\ 1 \end{Bmatrix} = \mathcal{T} \begin{Bmatrix} {}^{C}\mathbf{X}_{To} \\ 1 \end{Bmatrix} \tag{3.26}$$

Taking the first order time derivative on both sides of Eq. (3.26) yields the velocity of the target in $\mathcal{F}_G$ as

$$\begin{Bmatrix} \dot{\mathbf{X}}_T \\ 0 \end{Bmatrix} = \mathcal{T} \begin{Bmatrix} {}^{C}\dot{\mathbf{X}}_{To} \\ 0 \end{Bmatrix} + \dot{\mathcal{T}} \begin{Bmatrix} {}^{C}\mathbf{X}_{To} \\ 1 \end{Bmatrix} \tag{3.27}$$

Hereto, the target motion estimates in both $\mathcal{F}_C$ and $\mathcal{F}_G$ are obtained.

Chapter 4 KINEMATICS-BASED ROBOTIC CONTROL

Summary: In this chapter, the forward kinematics and inverse kinematics are introduced first. Then two kinematics-based robotic control approaches, interception-point-aimed and target-aimed, are developed. The effectiveness of the proposed robotic control approaches have been validated by hardware-in-the-loop simulations with eye-to-hand configuration. The theories and simulation results have been published in reference C and reference D.

4.1 Robotic Kinematics

Without consideration of the applied forces, robotic kinematics studies the movement of robotic manipulators analytically based on the geometry of the mechanical configuration [108]. Unless otherwise stated, the motion of the end-effector is considered in $\mathcal{F}_G$ by default.

Analyzing the motion of the end-effector in terms of joint configuration of the robotic manipulator is called forward kinematics. A common approach for the derivation of forward kinematics is the Denavit-Hartenberg (DH) method [109]. In the DH convention, as shown in Figure 4.1, coordinate frames are attached to each of the joints according to the following principles:

(i) The z-axis is in the direction of the joint axis.

(ii) The x-axis is perpendicular to z-axis of previous frame.

(iii) The x-axis intersects z-axis of previous frame.

(iv) The y-axis follows the right-handed convention.

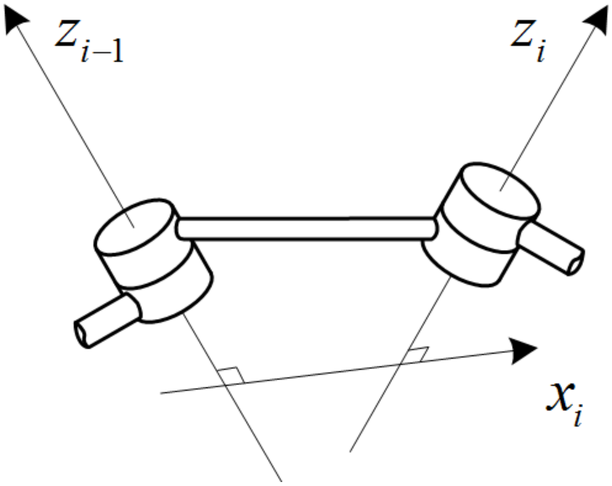


Figure 4.1 DH convention

Table 4.1 DH parameters

Parameters	Definitions
d	Offset distance along z_{i-1} from x_{i-1} to x_i
θ	Rotation angle about z_{i-1} from x_{i-1} to x_i
a	Offset distance along x_i from z_{i-1} to z_i
α	Rotation angle about x_i from z_{i-1} to z_i

By defining four parameters in Table 4.1, known as DH parameters

[110], each homogeneous transformation of the current frame with respect to the previous frame can be represented as a product of four basic rotational and translational transformations, such that

$$\mathcal{T}_{i-1,i} = \mathcal{R}_{z,\theta_i}\mathcal{D}_{z,d_i}\mathcal{D}_{x,a_i}\mathcal{R}_{x,\alpha_i} \tag{4.1}$$

$$\mathcal{R}_{z,\theta_i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

$$\mathcal{D}_{z,d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

$$\mathcal{D}_{x,a_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

$$\mathcal{R}_{x,\alpha_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

Thus,

$$\mathcal{T}_{i-1,i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

According to the DH-based homogeneous transformation, the position of the end-effector can be easily obtained by a sequence of transformations in terms of the geometric parameters i.e., link lengths, mechanical configuration, etc., as well as trigonometric functions of the joint variables. For a concise expression, the position-level forward kinematics is written as

$$\mathbf{X}_E = \mathbf{f}\left(\mathbf{\Theta}\right) \tag{4.7}$$

where $\mathbf{\Theta} \in \mathbf{R}^n$ denotes the joint variable vector of robotic manipulator, $\mathbf{X}_E \in \mathbf{R}^m$ denotes the position of the end-effector in $\mathscr{F}_G$.

Taking the first order time derivative of Eq. (4.7) leads to

$$\dot{\mathbf{X}}_E = \mathbf{J}_r \dot{\mathbf{\Theta}} \tag{4.8}$$

where $\mathbf{J}_r \in \mathbf{R}^{m \times n}$ stands for the Jacobian matrix of the robotic manipulator that is defined by the partial differential of Eq. (4.7), such as, $\mathbf{J}_r = \partial \mathbf{f} / \partial \mathbf{\Theta}$.

In contrast, the inverse kinematics is to solve for the configuration of robotic manipulator according to the motion of end-effector. Due to the periodicity of trigonometric functions and the redundant geometric configuration of the robotic manipulator, multiple solutions are frequently encountered in the inverse of Eq. (4.7). In order to obtain a unique solution, extra constraints are usually required. Then based on the definition of pseudo inverse and null space theory, the solution of the velocity-level inverse kinematics is written as

$$\dot{\mathbf{\Theta}} = \mathbf{J}_r^{\dagger} \dot{\mathbf{X}}_E + \left(\mathbf{I} - \mathbf{J}_r^{\dagger} \mathbf{J}_r\right) \boldsymbol{\xi} \tag{4.9}$$

where $\mathbf{J}_r^{\dagger}$ denotes the pseudo inverse of the Jacobian matrix, $\mathbf{I}$ denotes the identity matrix and $\boldsymbol{\xi}$ is an arbitrary vector that projected onto the null space of the Jacobian. For selected $\boldsymbol{\xi}$, additional kinematic objectives can be achieved.

Since the Jacobian matrix is usually assumed as full row rank in robotics, the calculation of the pseudo inverse can be simplified by the right inverse as

$$\mathbf{J}_r^{\dagger} = \mathbf{J}_r^T \left(\mathbf{J}_r \mathbf{J}_r^T \right)^{-1} \tag{4.10}$$

By substituting Eq. (4.10) into Eq. (4.9) and neglecting the null space solution leads to

$$\dot{\boldsymbol{\Theta}} = \mathbf{J}_r^T \left(\mathbf{J}_r \mathbf{J}_r^T \right)^{-1} \dot{\mathbf{X}}_E \tag{4.11}$$

4.2 Interception-Point-Driven Approach

Assume the configuration of the robotic manipulator can be detected by the optical shaft encoders mounted in joints, and the target motion is detected and estimated by the vision system. Thus, configuration of the robotic manipulator and target motion are adopted as feedback information. Since stepper motors have great potential in space robotic systems due to the high load capacity and high positioning accuracy, incremental joint angles are considered as control input and joint velocity limits are regarded as the main constraints. We further assume the target is always reachable by the robotic manipulator and the speed of the end-effector is always greater than the speed of the target in order to ensure the superior maneuverability of the robotic manipulator over the target.

An interception-point-driven (IPD) robotic control approach is develop as follows. Consider a successful capture process, the interception point can be approximately estimated as

$$\mathbf{X}_C = \mathbf{X}_T + \dot{\mathbf{X}}_T t \tag{4.12}$$

where $\mathbf{X}_C$ denotes certain interception point, t denotes the elapsed time of the capture process. It should be noted that the both $\mathbf{X}_C$ and t are unknown and need to be determined.

Meanwhile, the end-effector has to reach the same interception point after the same time period, such that

$$\mathbf{X}_C = \mathbf{X}_E + \dot{\mathbf{X}}_E t \tag{4.13}$$

Since $\mathbf{X}_T$ and $\dot{\mathbf{X}}_T$ are the estimated position and velocity of the target in $\mathcal{F}_G$ obtained in section 3.5, and $\mathbf{X}_E$ can be obtained by forward kinematics in Eq. (4.7), the velocity of the end-effector can be determined based on Eq. (4.12) and Eq. (4.13) as

$$\dot{\mathbf{X}}_E = \frac{1}{t}\left\{ \mathbf{X}_T + t\dot{\mathbf{X}}_T - \mathbf{X}_E \right\} \tag{4.14}$$

Substituting Eq. (4.14) into Eq. (4.11) yields

$$\dot{\mathbf{\Theta}} = \frac{1}{t}\mathbf{J}_r^{\dagger}\left\{\mathbf{X}_T + t\dot{\mathbf{X}}_T - \mathbf{X}_E\right\} \tag{4.15}$$

Let $\dot{\theta}_i$, p_{ij}, x_{Tj}, $\dot{x}_{Tj}$ and x_{Ej} denote the elements of $\dot{\mathbf{\Theta}}$, $\mathbf{J}_r^{\dagger}$, $\mathbf{X}_T$, $\dot{\mathbf{X}}_T$ and $\mathbf{X}_E$, respectively, Eq. (4.15) can be decomposed into n equations, such as

$$\dot{\theta}_i = \frac{1}{t}\sum_{j=1}^{m} p_{ij}\left(x_{Tj} + t\dot{x}_{Tj} - x_{Ej}\right), \quad i = 1, 2, \cdots, n. \tag{4.16}$$

The joint velocity limits of each actuators can be easily determined by the physical limits and the transmission mechanisms, defined as

$$\dot{\mathbf{\Theta}}_{\max} = \left(\dot{\theta}_{\max 1}, \dot{\theta}_{\max 2}, \cdots, \dot{\theta}_{\max n}\right)^T \tag{4.17}$$

By assuming that the joint velocity has the same limit in both forward and reverse revolute directions, each of the right-hand sides of Eq. (4.16) is bounded by the closed interval $\left[-\dot{\theta}_{\max i}, \dot{\theta}_{\max i}\right]$, such that

$$-\dot{\theta}_{\max i} \le \frac{1}{t}\sum_{j=1}^{m} p_{ij}\left(x_{Tj} + t\dot{x}_{Tj} - x_{Ej}\right) \le \dot{\theta}_{\max i}, \quad i = 1, 2, \cdots, n. \tag{4.18}$$

The inequalities in Eq. (4.18) are solved for t with the consideration of $t > 0$. Then intersect each of the solution intervals, denoted by $t_1, t_2, \cdots, t_n$, and take the lower bound of the intersection as the solution for t, such as

$$t = \min\left\{t_1 \cap t_2 \cap \cdots \cap t_n\right\} \tag{4.19}$$

Once t is determined, evaluating Eq. (4.15) yields the joint velocity of the end-effector. The control input for the robotic manipulator in the next sampling step is obtained by the product of the joint velocity and the sampling time.

After the end-effector moves one step towards the predicted interception point, the procedure is repeated until the capture requirements achieved. It is worth pointing out that as the end-effector approaches the dynamically predicted interception point, the prediction error induced by the approximation of uniform motion in Eq. (4.12) and Eq. (4.13) will be diminished eventually.

4.3 Target-Driven Approach

An alternative target-driven (TD) robotic control approach is developed under the same assumptions for the same problem defined in section 4.2.

According to the estimated position and velocity of the target in $\mathcal{F}_G$, denoted by $\mathbf{X}_T$ and $\dot{\mathbf{X}}_T$, respectively, obtained in section 3.5, if the sampling time t_s is sufficient small, the position of the target in the next time instant can be estimated by

$$\mathbf{X}_{T'} = \mathbf{X}_T + \dot{\mathbf{X}}_T t_s \tag{4.20}$$

In order to track the target promptly, the instantaneous velocity of the end-effector can be set to align with the vector defined by $\mathbf{X}_{T'} - \mathbf{X}_E$. Normalizing the vector yields

$$\mathbf{n}_{ET'} = \frac{\mathbf{X}_{T'} - \mathbf{X}_E}{\left|\mathbf{X}_{T'} - \mathbf{X}_E\right|} \tag{4.21}$$

Define a scalar factor $\lambda > 0$, such that the instantaneous velocity of the end-effector can be expressed as

$$\dot{\mathbf{X}}_E = \lambda \mathbf{n}_{ET'} \tag{4.22}$$

Substituting Eq. (4.22) into Eq. (4.11) leads to

$$\dot{\mathbf{\Theta}} = \lambda \mathbf{J}_r{}^{\dagger} \mathbf{n}_{ET'} \tag{4.23}$$

Let $\dot{\theta}_i$, p_{ij} and $n_{ET'j}$ be the elements of $\dot{\mathbf{\Theta}}$, $\mathbf{J}_r{}^{\dagger}$ and $\mathbf{n}_{ET'}$, respectively. Then Eq. (4.23) can be decomposed into n equations, such that

$$\dot{\theta}_i = \lambda \sum_{j=1}^{m} p_{ij} n_{ET'j}, \quad i = 1, 2, \cdots, n. \tag{4.24}$$

Applying the joint velocity limits defined in Eq. (4.17) to each of the right-hand sides in Eq. (4.24) leads to

$$-\dot{\theta}_{\max i} \le \lambda \sum_{j=1}^{m} p_{ij} n_{ET'j} \le \dot{\theta}_{\max i}, \quad i = 1, 2, \cdots, n. \tag{4.25}$$

The inequalities in Eq. (4.25) are solved for λ with the consideration of $\lambda > 0$. Then intersect each of the solution intervals, denoted by $\lambda_1, \lambda_2, \cdots, \lambda_n$, and take the upper bound of the intersection as the solution for λ, such as

$$\lambda = \max\{\lambda_1 \cap \lambda_2 \cap \cdots \cap \lambda_n\} \tag{4.26}$$

Once λ is determined, evaluating Eq. (4.23) yields the joint velocity of the end-effector. The control input for the robotic manipulator in the next sampling step is obtained by the product of the joint velocity and sampling time, the same as in the interception-point-driven approach.

After the end-effector moves one step towards the target, the procedure is repeated until the capture requirements are achieved. It is worth pointing out that since the instantaneous velocity of the end-effector is always pointing to the target in the next time instant, the possibility of losing track of the target by the vision system is greatly reduced.

4.4 Stability Analysis

Define the tracking error of the robotic capture as

$$\mathbf{E} = \mathbf{X}_T - \mathbf{X}_E \tag{4.27}$$

$$\dot{\mathbf{E}} = \dot{\mathbf{X}}_T - \dot{\mathbf{X}}_E \tag{4.28}$$

Further define the Lyapunov function as

$$\mathbf{V} = \frac{1}{2}\mathbf{E}^T\mathbf{E} \tag{4.29}$$

Then taking the first order time derivative of Eq. (4.29) leads to

$$\dot{\mathbf{V}} = \mathbf{E}^T\dot{\mathbf{E}} \tag{4.30}$$

4.4.1 Stability of the IPD approach

From Eq. (4.12) and Eq. (4.13) we have

$$\mathbf{X}_E + t\dot{\mathbf{X}}_E = \mathbf{X}_T + t\dot{\mathbf{X}}_T \tag{4.31}$$

Rearranging Eq. (4.31) and plugging in Eq. (4.27) and Eq. (4.28) yield

$$\dot{\mathbf{E}} = -\frac{1}{t}\mathbf{E} \tag{4.32}$$

Substituting Eq. (4.32) into Eq. (4.30) leads to

$$\dot{\mathbf{V}} = -\frac{1}{t}\mathbf{E}^T\mathbf{E} \le 0 \tag{4.33}$$

Because t is the elapsed time of the capture process and is always positive, the IPD approach is asymptotically stable.

4.4.2 Stability of the TD approach

The geometry of the TD approach is shown in Figure 4.2. The angle between the vector direction of $\mathbf{E} = \mathbf{X}_T - \mathbf{X}_E$ and the vector direction of $\dot{\mathbf{X}}_T$ is defined as φ, and the angle between the vector direction of $\mathbf{E} = \mathbf{X}_T - \mathbf{X}_E$ and the vector direction of $\mathbf{X}_{T'} - \mathbf{X}_E$ is defined as ϕ.

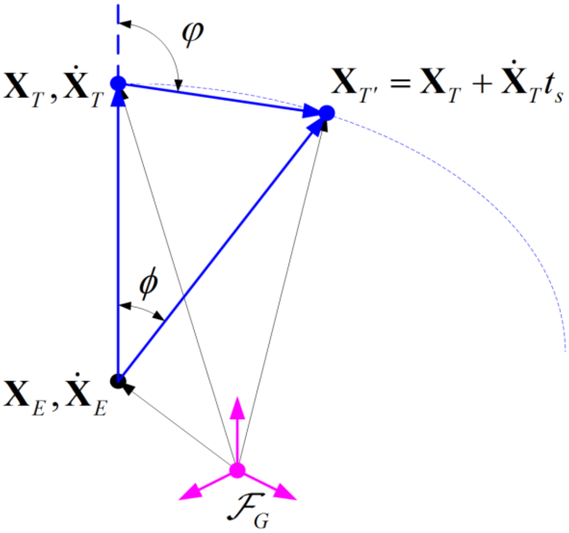

φ
$\mathbf{X}_T, \dot{\mathbf{X}}_T$
$\mathbf{X}_{T'} = \mathbf{X}_T + \dot{\mathbf{X}}_T t_s$
ϕ
$\mathbf{X}_E, \dot{\mathbf{X}}_E$
$\mathcal{F}_G$

Figure 4.2 Geometry of the TD approah

It should be noted that φ is an exterior angle of a triangle and ϕ is an interior angle of the triangle. Thus, with the consideration that the sampling time t_s is sufficiently small, the following relationships are true

$$0 < \phi < \pi/2 \tag{4.34}$$

$$0 < \varphi < \pi \tag{4.35}$$

$$\phi < \varphi \tag{4.36}$$

According to the monotonically decreasing property of cosine functions in the interval $[0, \pi]$ we obtain

$$\cos\varphi < \cos\phi \tag{4.37}$$

$$\cos\phi > 0 \tag{4.38}$$

From Eq. (4.21) and Eq. (4.22) we have

$$\dot{\mathbf{X}}_E = \frac{\lambda}{\left|\mathbf{X}_{T'} - \mathbf{X}_E\right|}\left(\mathbf{X}_{T'} - \mathbf{X}_E\right) \tag{4.39}$$

Substituting Eq. (4.39) into Eq. (4.28) yields

$$\dot{\mathbf{E}} = \dot{\mathbf{X}}_T - \frac{\lambda}{\left|\mathbf{X}_{T'} - \mathbf{X}_E\right|}\left(\mathbf{X}_{T'} - \mathbf{X}_E\right) \tag{4.40}$$

Further substituting Eq. (4.40) into Eq. (4.30) leads to

$$\dot{\mathbf{V}} = \mathbf{E}^T\dot{\mathbf{E}} = \mathbf{E}^T\dot{\mathbf{X}}_T - \frac{\lambda}{\left|\mathbf{X}_{T'} - \mathbf{X}_E\right|}\mathbf{E}^T\left(\mathbf{X}_{T'} - \mathbf{X}_E\right) \tag{4.41}$$

According to the definition of dot product, we have

$$\mathbf{E}^T\dot{\mathbf{X}}_T = \left|\mathbf{E}\right|\left|\dot{\mathbf{X}}_T\right|\cos\varphi \tag{4.42}$$

$$\mathbf{E}^T\left(\mathbf{X}_{T'} - \mathbf{X}_E\right) = \left|\mathbf{E}\right|\left|\mathbf{X}_{T'} - \mathbf{X}_E\right|\cos\phi \tag{4.43}$$

Substituting Eq. (4.42) and Eq. (4.43) into Eq. (4.41) yields

$$\dot{\mathbf{V}} = \left|\mathbf{E}\right|\left|\dot{\mathbf{X}}_T\right|\cos\varphi - \left|\mathbf{E}\right|\lambda\cos\phi \tag{4.44}$$

Note that the term $\left|\dot{\mathbf{X}}_T\right|\cos\varphi$ is the modulus of the speed projection of the target along the vector $\mathbf{E} = \mathbf{X}_T - \mathbf{X}_E$, and the term $\lambda\cos\phi$ is the modulus of the speed projection of the end-effector along the same vector, respectively.

The assumption that the speed of the end-effector is always greater than the speed of the target can be interpreted as

$$\left|\dot{\mathbf{X}}_T\right| < \lambda \tag{4.45}$$

From Eq. (4.37), Eq. (4.38) and Eq. (4.45) we obtain

$$\left|\dot{\mathbf{X}}_T\right|\cos\varphi < \left|\dot{\mathbf{X}}_T\right|\cos\phi < \lambda\cos\phi \tag{4.46}$$

Thus,

$$\dot{\mathbf{V}} = \left|\mathbf{E}\right|\left|\dot{\mathbf{X}}_T\right|\cos\varphi - \left|\mathbf{E}\right|\lambda\cos\phi \le 0 \tag{4.47}$$

Therefore, the TD approach is asymptotically stable.

Chapter 5 EXPERIMENTAL SYSTEM

Summary: In order to validate the proposed estimation algorithms of the non-cooperative target experimentally as well as the kinematics-based incremental control approaches, experimental systems of the non-cooperative target and robotic manipulator are custom designed and built. The supporting hardware and chip software of the robotic manipulator, as well as the computer software of the target-manipulator system for the experimental validation are designed and developed from sketches. The experimental validations of the proposed visual servo robotic control strategies have been performed on the customized target-manipulator system with eye-in-hand configuration. Theories and experimental results have been included in reference E and reference F.

5.1 Non-cooperative Target System

The target system had been designed and built by [107]. As shown in Figure 5.1, the target is anchored on the ceiling at the top left corner and driven by the stepper motor fixed at the top right corner. By independently operating the target system, a non-cooperative target is generated for experimental validation. The stepper motor is driven by the Little Step-U controller, a product from TLA Microsystems Ltd. A USB to serial adapter cable is then adopted for the connection interface conversion.

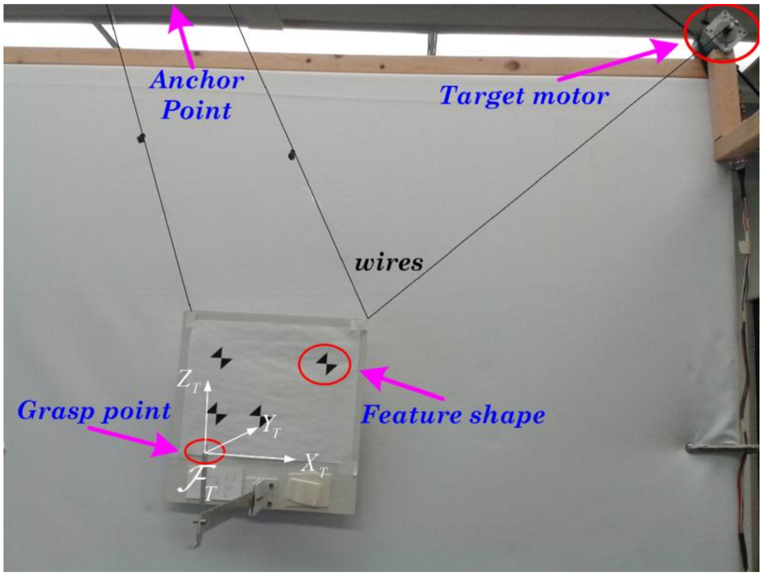


Figure 5.1 Non-cooperative target system

5.2 Robotic Manipulator System

The customized robotic manipulator is a Pieper configuration [111] with an eye-in-hand camera mounted closely to the end-effector. The mechanical design, machining and assembly had been accomplished by [107]. As shown in Figure 5.2, the robotic manipulator consists of three main revolute joints driven by stepper motors and a dexterous end-effector driven by servo motors. The main joints, namely torso, shoulder and elbow, are denoted by θ_1, θ_2 and θ_3, respectively.

As a mechatronic system, electronic hardware plays an important role in the robotic manipulator. Electronic devices, such as the self-developed shaft encoder readers, stepper motor drivers, servo motor driver, etc., require communication with desktop computer. Since each of the devices may have different connection interfaces, they are all converted to the standard USB interface for a concise hardware network.

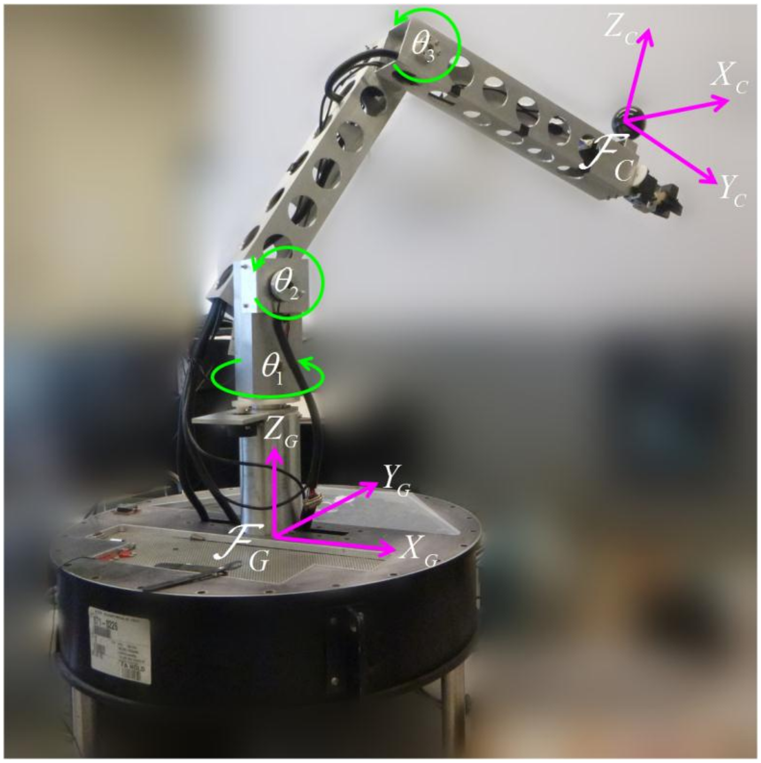


Figure 5.2 Customized robotic manipulator

5.2.1 Shaft encoder reading circuit

In order to measure the joint configuration of the robotic manipulator in real time, three optical shaft encoders are mounted in the main joints. The adopted encoder is the E8P-400-250-D-H-M-B as shown in Figure 5.3, a product from

US Digital. The resolution is $0.9°$, or 400 cycles per revolution (CPR), and the output are two TTL signals, A and B, with $90°$ phase difference. Viewed from the cover side of the encoder, A leads B indicates a clockwise (CW) rotation while B leads A indicates a counterclockwise (CCW) rotation.

Figure 5.3 E8P Shaft encoder, a product from US Digital

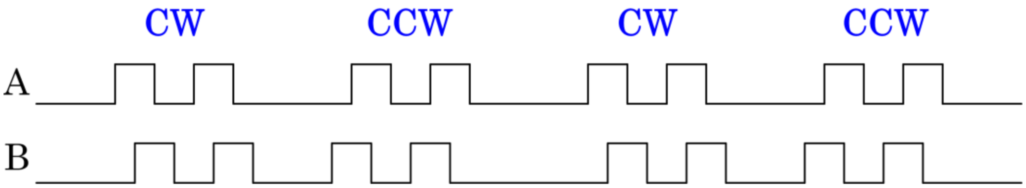


Figure 5.4 Definition of the rotation direction of the encoder

In order to obtain the rotation angle from the encoder, an intuitive way is to record the number of pulses of the output signals by an up/down counter. As a result, the up/down counting flag and the counting clock have to be determined in advance. For ease of description, the counter is set to count up on a CW rotation, and count down on a CCW rotation. The schematic circuit adopts an edge-triggered D flip-flop to obtain the up/down flag, As illustrated in Figure 5.5, where D stands for data input, CLK stands for the clock input,

U/D stands for the up/down counting direction, Q is the output, and $\bar{Q}$ is the inverse of Q. In order to increase the timing margin, the output signals of the encoder are treated by an AND gate and an OR gate beforehand.

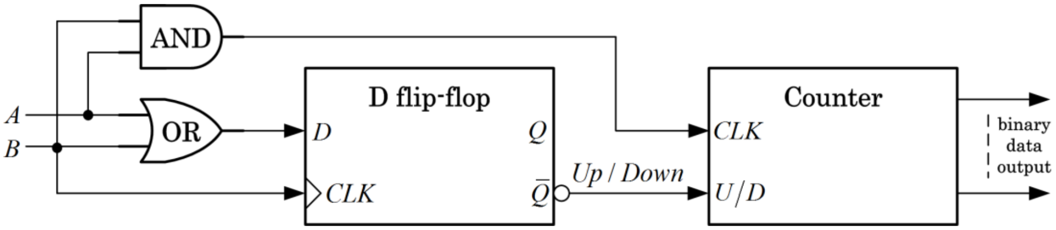


Figure 5.5 Schematic diagram of D flip-flop and Up/Down counter

Table 5.1 Truth table of a typical edge-triggered D flip-flop

D	CLK	Q	$\bar{Q}$
0	Trigger edge	0	1
1	Trigger edge	1	0
X	0	No change	No change
X	1	No change	No change

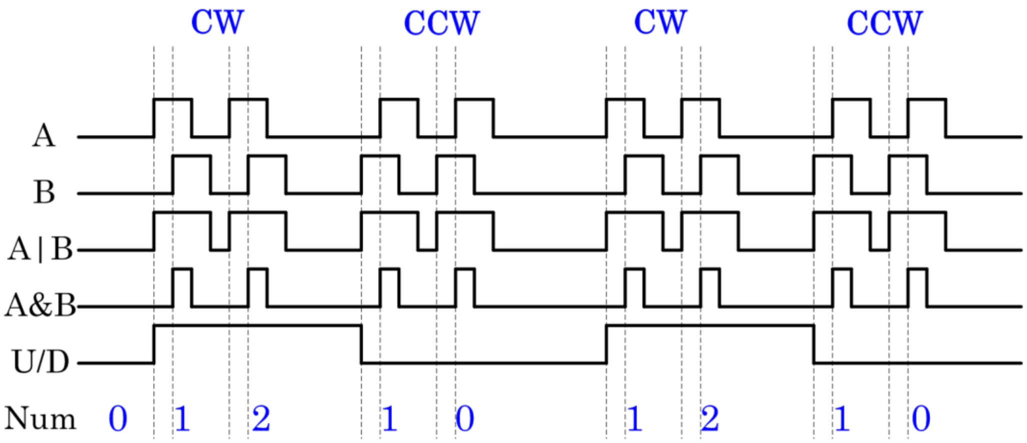


Figure 5.6 Ideal timing diagram of the counting circuit

According to the Schematic diagram of the counting circuit and the truth table of a typical edge-triggered D flip-flop, as shown in Table 5.1, the ideal timing diagram of the counting circuit is illustrated in Figure 5.6. We can see the circuit works perfect on the given output signals. However, because the resolution of the stepper motor (considering gear boxes) may be higher than the encoder, the circuit may be problematic, the timing diagram is shown in Figure 5.7. As can be seen that the error accumulates when rotation direction switches at every incomplete steps of the encoder.

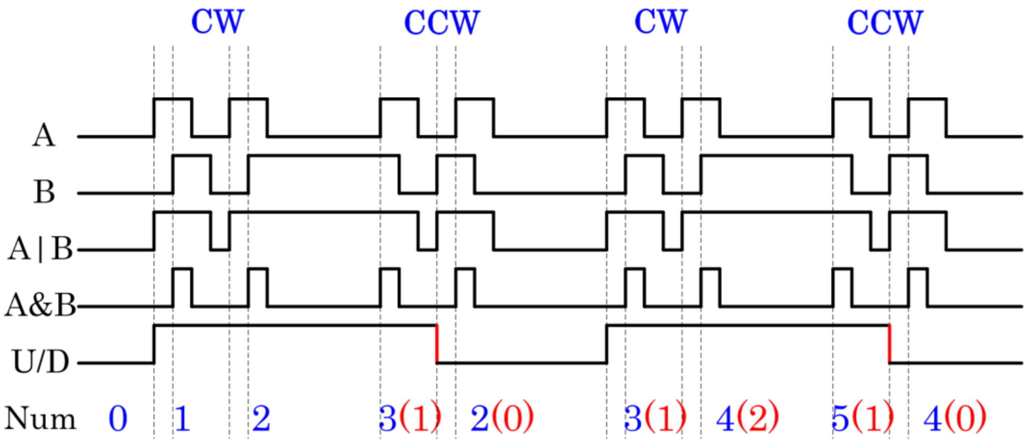


Figure 5.7 Problematic timing diagram of the counting circuit

In order to eliminate the problem and feedback the information to the desktop computer, a shaft encoder reading circuit is self-developed based on the microcontroller (ATMEGA324PA-PU). Figure 5.8 shows the block diagram of a dual-channel shaft encoder reader. The microcontroller is programmed under Atmel Studio 6.0, where the pin change interrupt (PCINT) of the microcontroller is used to detect both the rising and falling edges for the sake of doubling the CPR.

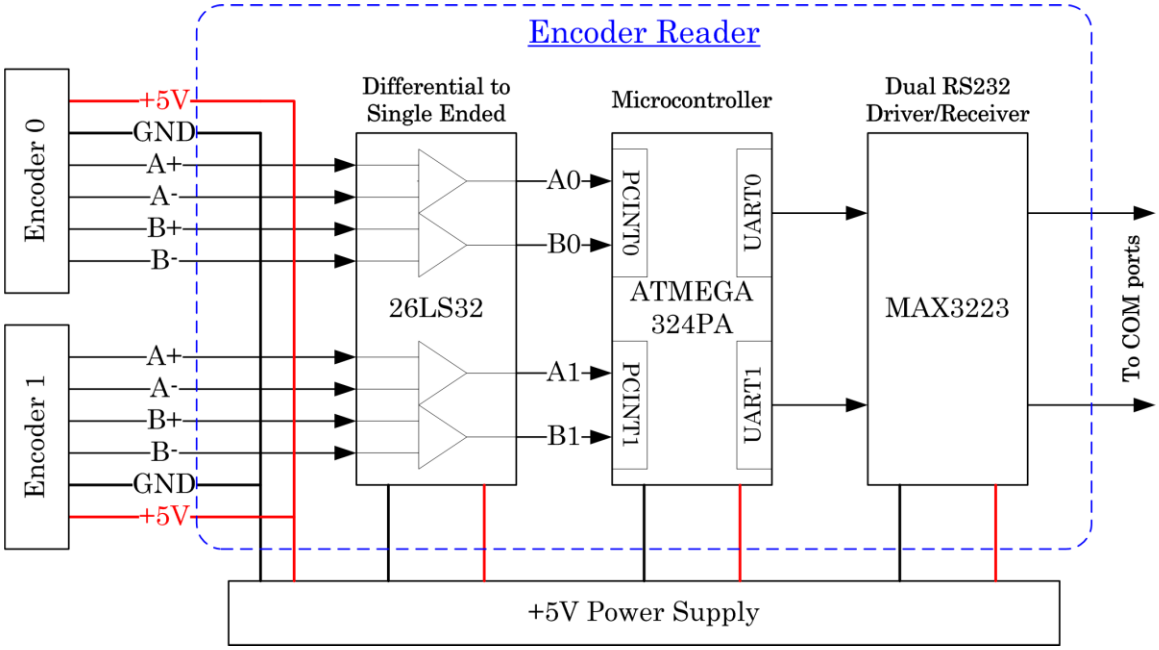


Figure 5.8 Diagram of the shaft encoder reading circuit

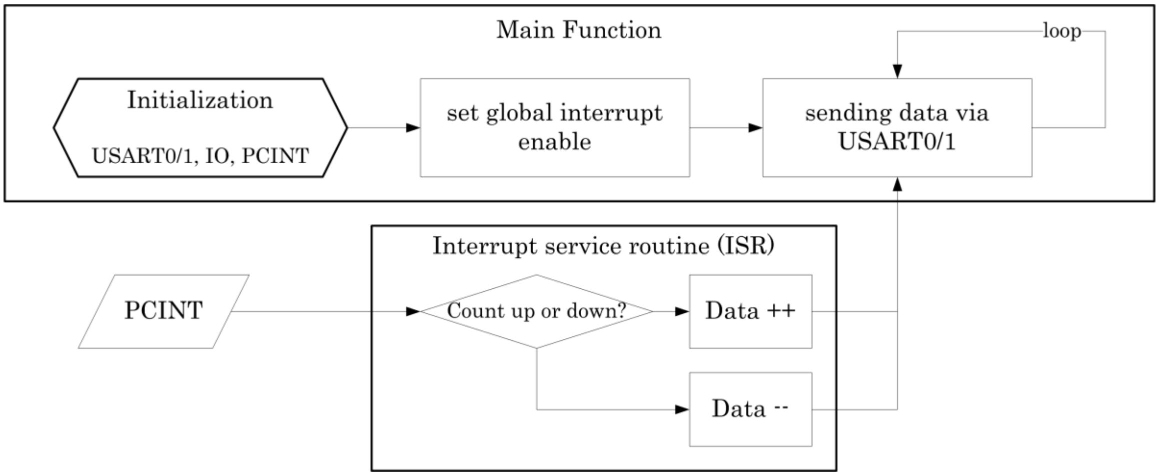


Figure 5.9 Workflow of the chip software for shaft encoder reader

The workflow of the chip software for shaft encoder reader is as shown in Figure 5.9. The rising and falling edges of A and B are detected by PCINT of the microcontroller. Once an edge change detected, the electrical level of another output signal is checked to determine the counting direction and count on edge changes at the same time. The timing diagram of the microcontroller-based encoder reading circuit is shown in Figure 5.10. By counting on both rising and falling edges, the CPR is doubled virtually by the chip software.

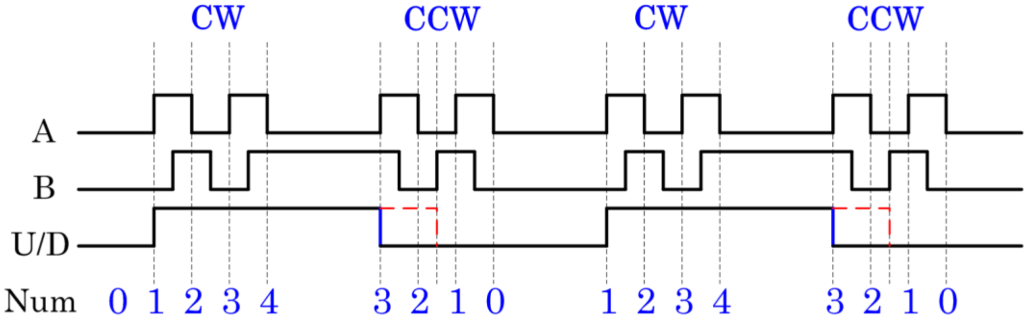


Figure 5.10 Timing diagram of double data rate

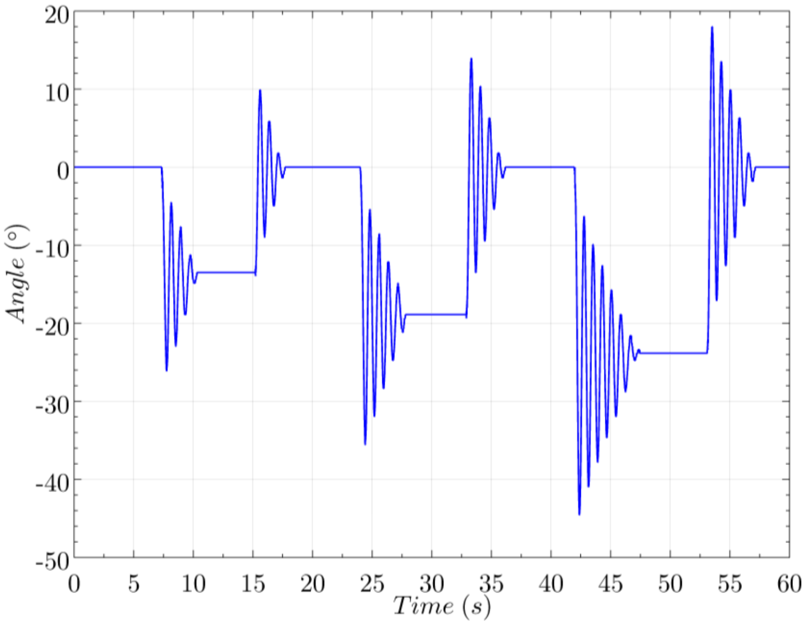


Figure 5.11 Physical test result of the encoder reader

The encoder reading circuit is physically tested on a flexible joint robotic manipulator to be described in Chapter 6. By commanding the motor to rotate certain angles, the encoder records the angle of the link caused by the flexible joint. The test result is shown in Figure 5.11.

According to Figure 5.8, the connection interface of the encoder reader is RS232. Since three shaft encoders are mounted on the joints of the robotic manipulator, a USB to four channel RS232 communication module, as shown in Figure 5.12, is selected for the interface conversion.

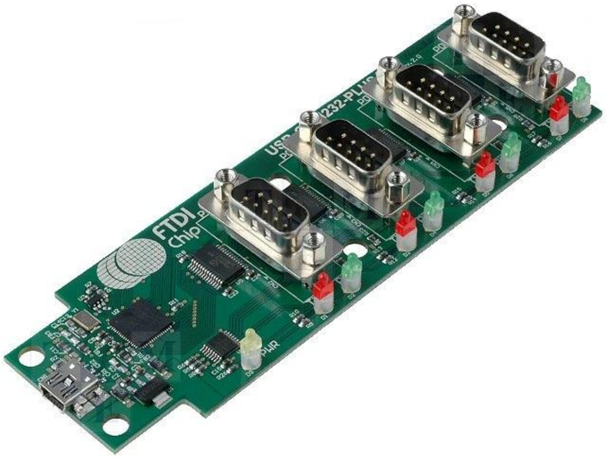


Figure 5.12 USB-COM232-PLUS4, a product from FTDI

5.2.2 Stepper motor drivers

The three stepper motors in the main joints are separately driven by Ocean Controls KT-5196 bipolar stepper motor controllers. A microcontroller (ATMEGA324PA-PU) based circuit with a RS232 interface is self-developed for bridging the connection. The chip is programmed in order to perform the incremental angular position control. A diagram of the chip software can be seen in Figure 5.13. The circuits of the encoder readers and the connection bridge for stepper motor controllers are built as shown in Figure 5.14.

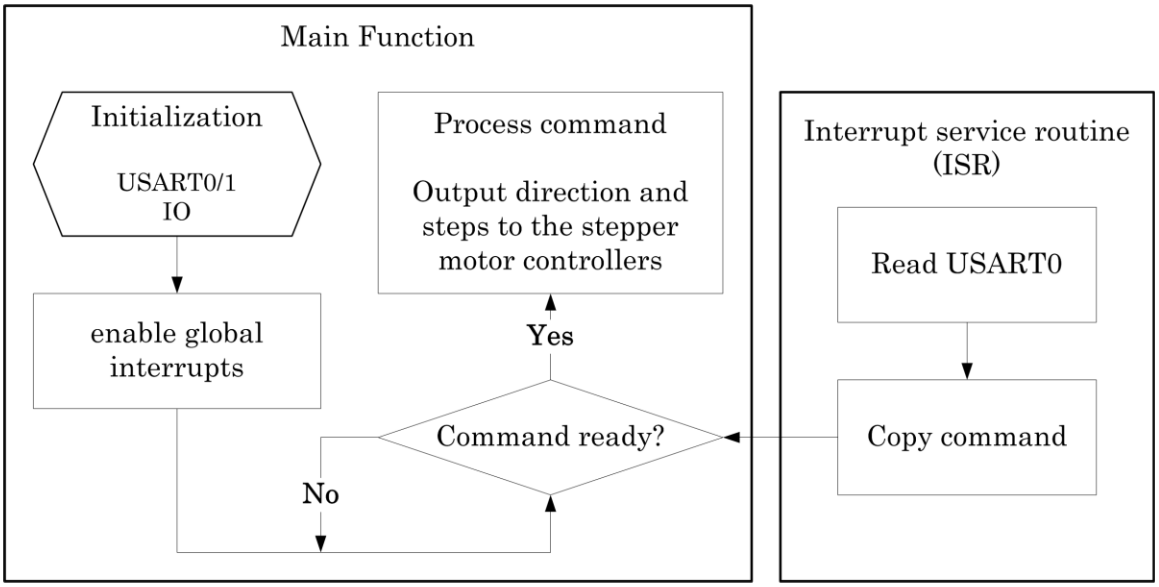


Figure 5.13 Diagram of the chip software for connection bridge

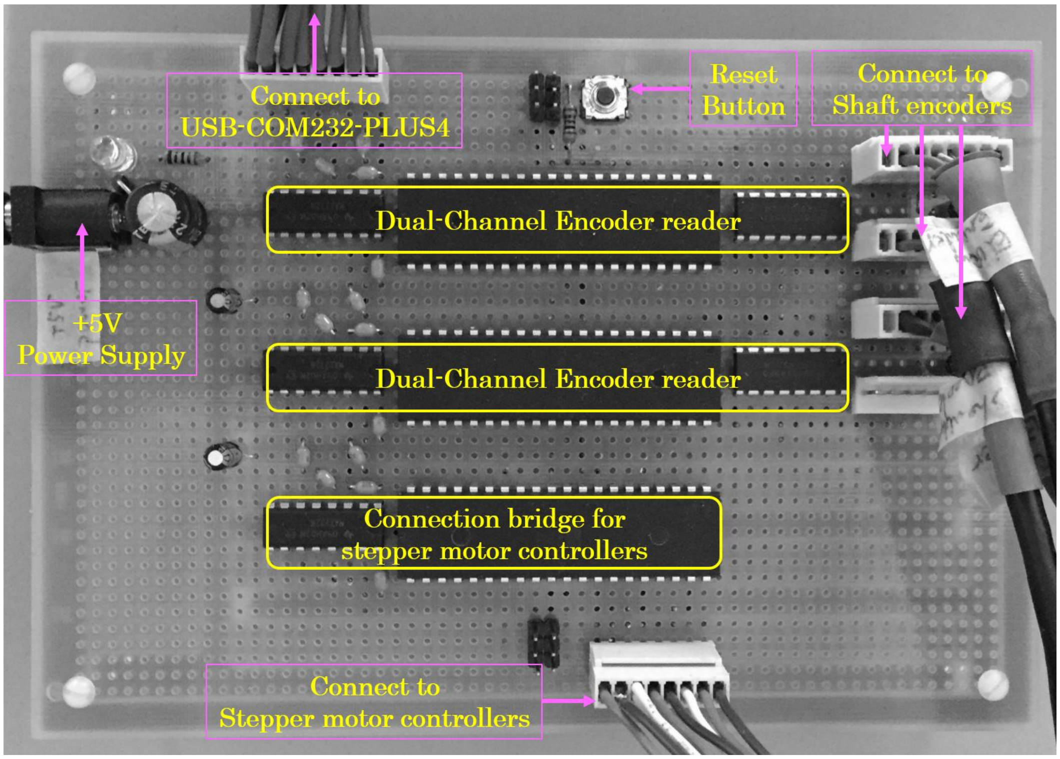
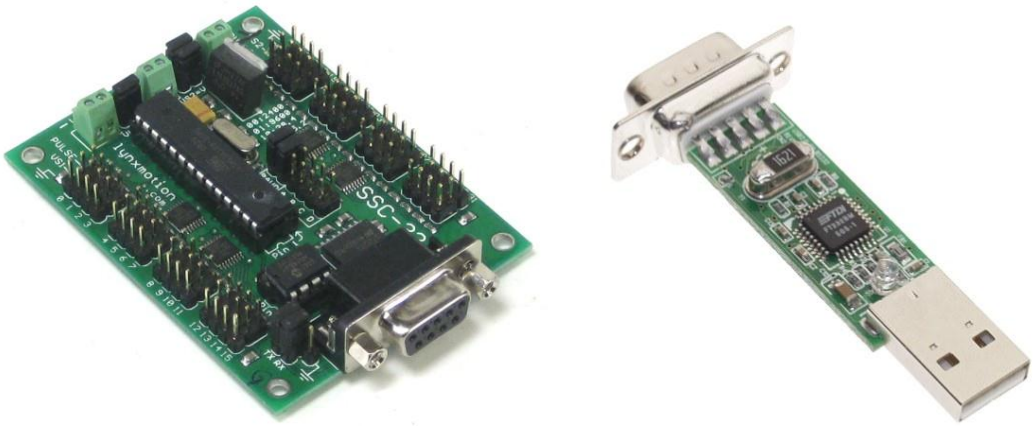


Figure 5.14 Circuits of encoder readers and connection bridge

5.2.3 Servo motor driver

As shown in Figure 5.15(a), the servo motors of the end-effector are driven by a 32-channel servo motor controller SSC-32, a product from Lynxmotion, Inc [112], with a DB9 RS232 connection port. The interface is directly converted by a single channel USB to serial adapter shown in Figure 5.15(b).



(a) SSC-32 servo motor controller (b) USB to serial adapter

Figure 5.15 Servo motor dirver

5.2.4 Hardware connection and communication

Except the adopted camera, QuickCam Orbit AF (model number: V-UCC22), a product from Logitech, which comes with corded USB, hereto all the electronic devices are converted into USB connection interface. A USB hub, DUB-H7, a product from D-link, is then adopted for bridging the devices with the desktop computer. A complete hardware connection network of the target-manipulator system is as shown in Figure 5.16.

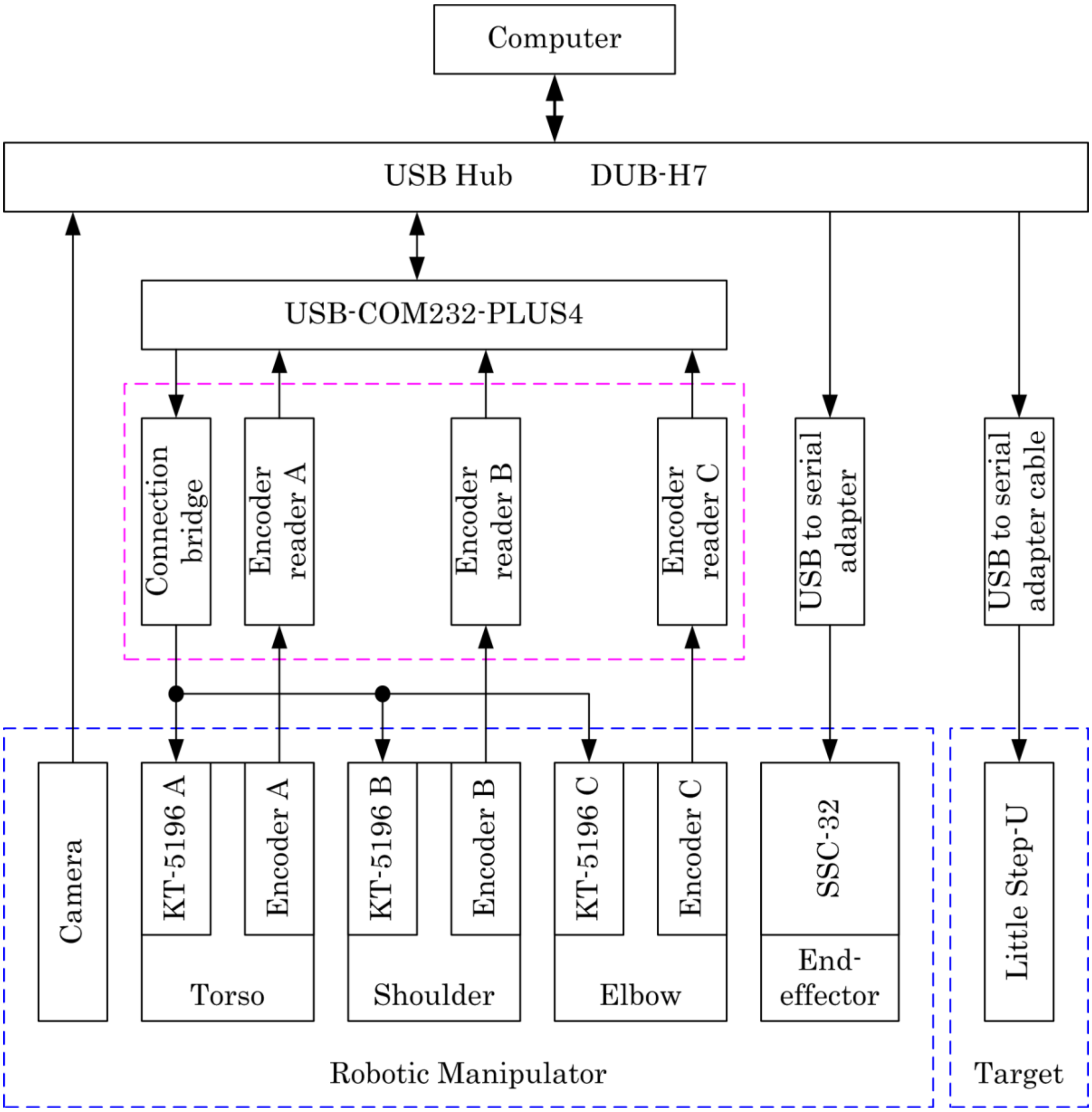


Figure 5.16 Hardware network of the target-manipulator system

From Figure 5.16 we can see, except the camera, the communication between all other devices and the computer is asynchronous half duplex in accordance with the RS232 protocol. We preset the data format as 8 bits data, non-parity and one stop bit for all the devices. The baud rates of the devices are listed in Table 5.2.

Table 5.2 Settings of baud rate

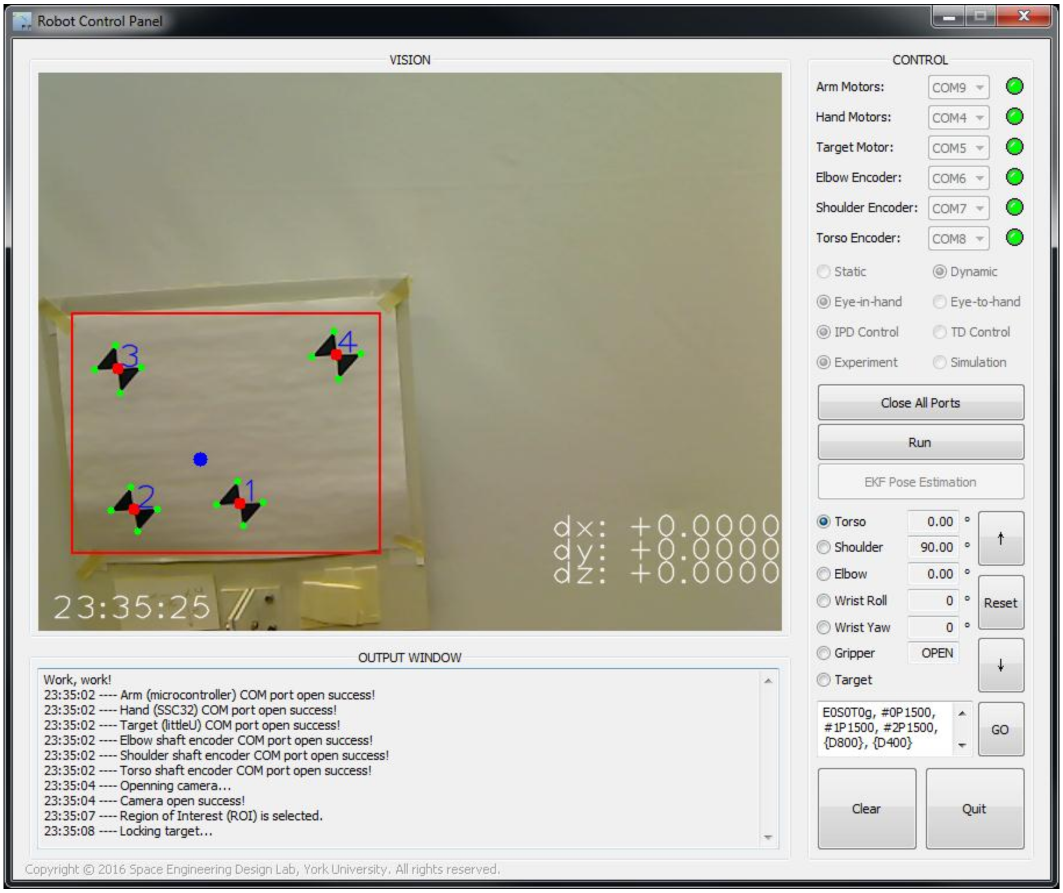
Devices	Baud Rate (bps)
Connection bridge for stepper motor driver	9600
Servo motor driver (SSC-32)	115200
Target motor driver (Little Step-U)	2400
Encoder readers	9600

5.3 Computer Software Design

In order to provide a user-friendly interface, the windows-based computer software is developed under the Microsoft Visual Studio 2012 based on the Microsoft Foundation Classes (MFC).

5.3.1 User interface

The user interface of the computer software, as shown in Figure 5.17, mainly consists of three subareas, Vision, Output and Control. As can be seen, the subarea of Vision is assigned for displaying the image acquired by the eye-in-hand camera, as well as providing interactive functions, such as, selecting the region of interest (ROI) of the image by mouse dragging, confirming the selection of ROI by single hit of the space key, displaying the ROI in red rectangle, displaying corners within the ROI, displaying system time, and other useful information during the experiments. Then the subarea of Output is used to display debug information and prompt messages of hardware status, workflow steps, and so on. Finally the subarea of Control conducts the main operations of the target-manipulator system. The functions include but are not limited to serial port mapping, hardware initialization, experimental settings, program run/stop, joint angles measured by shaft encoder readers, status of the end-effector, reset and debug operations of each motors, software termination, etc.



Arm Motors: COM9 ●
Hand Motors: COM4 ●
Target Motor: COM5 ●
Elbow Encoder: COM6 ●
Shoulder Encoder: COM7 ●
Torso Encoder: COM8 ●

○ Static ● Dynamic
● Eye-in-hand ○ Eye-to-hand
● IPD Control ○ TD Control
● Experiment ○ Simulation

● Torso 0.00 °
○ Shoulder 90.00 °
○ Elbow 0.00 °
○ Wrist Roll 0 °
○ Wrist Yaw 0 °
○ Gripper OPEN
○ Target

```
E0S0T0g, #0P1500,
#1P1500, #2P1500,
{D800}, {D400}
```

```
Work, work!
23:35:02 ---- Arm (microcontroller) COM port open success!
23:35:02 ---- Hand (SSC32) COM port open success!
23:35:02 ---- Target (littleU) COM port open success!
23:35:02 ---- Elbow shaft encoder COM port open success!
23:35:02 ---- Shoulder shaft encoder COM port open success!
23:35:02 ---- Torso shaft encoder COM port open success!
23:35:04 ---- Openning camera...
23:35:04 ---- Camera open success!
23:35:07 ---- Region of Interest (ROI) is selected.
23:35:08 ---- Locking target...
```

Copyright © 2016 Space Engineering Design Lab, York University. All rights reserved.

Figure 5.17 User interface of the computer software

5.3.2 Workflows

A block diagram of the computer software is shown in Figure 5.18. Once the software starts to run, the user interface window is created simultaneously. The serial port should be manually mapped to the corresponding electronic devices then. By a single click on the Initialization button, the USB and serial devices will be initialized and enabled accordingly. If failure occurs on the enabling of any devices, the serial ports mapping and connections of hardware devices should be checked until all the devices are successfully activated. Intuitively, each of the serial ports also has a status indicator. The colors of the indicator, dark green, green and red, indicate the serial port is disabled, enabled and failure, respectively. Then manual control of the target-manipulator system, such as commanding each of the motors and resetting the system, as well as experimental settings, such as camera configuration and control method selection, will be enabled if the devices are open successfully in initialization. Meanwhile, an interrupt service routine (ISR) for receiving data from serial ports of encoder readers, and a new thread for the visual servo robotic control are created as well. By clicking the Run/Stop button, the global flags will be set to enable/disable the visual servo robotic control. The computer software can be terminated at any time for safe operation of the mechatronic system by a single click on the Quit button.

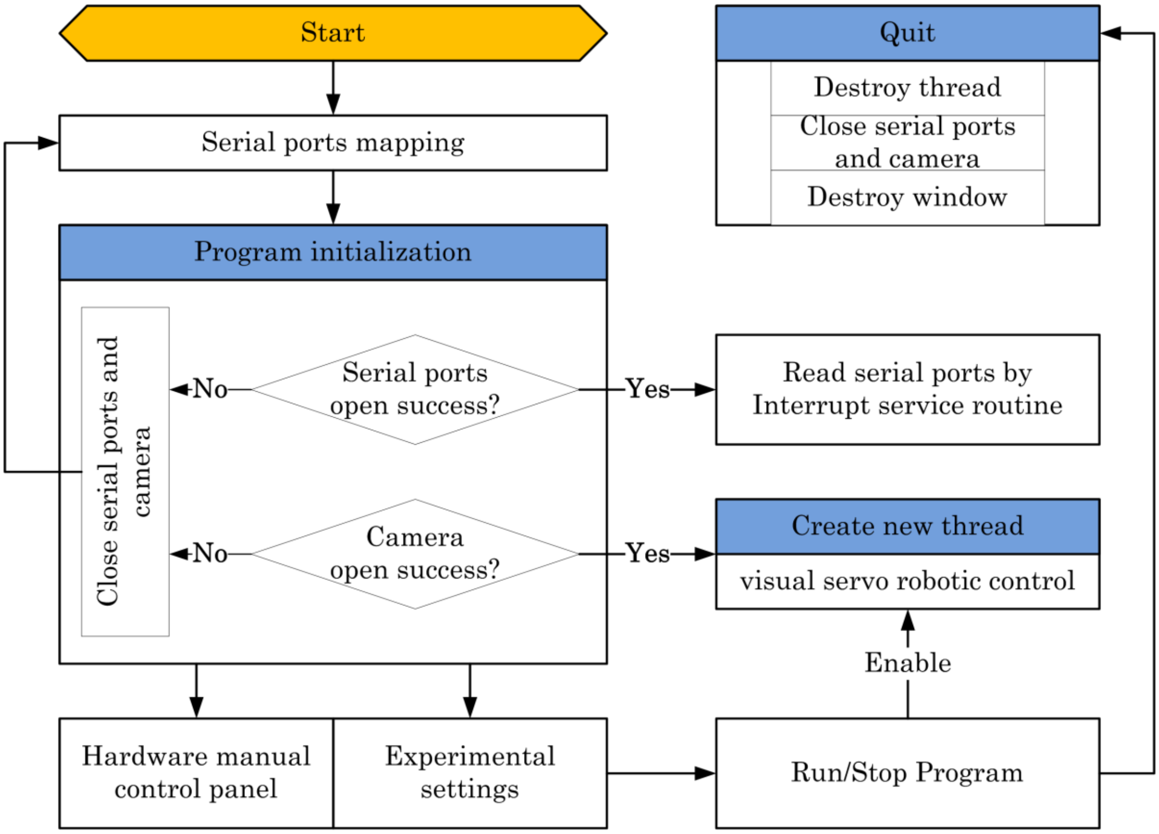

Start
Serial ports mapping
Program initialization
Close serial ports and camera
Serial ports open success?
No
Yes
Camera open success?
No
Yes
Quit
Destroy thread
Close serial ports and camera
Destroy window
Read serial ports by Interrupt service routine
Create new thread
visual servo robotic control
Enable
Hardware manual control panel
Experimental settings
Run/Stop Program

Figure 5.18 Block diagram of the computer software

To introduce the key part in details, the workflow of the visual servo robotic control thread is concisely shown in Figure 5.19. After the thread being created, the target tracker is a looped function used to detect the corners within ROI, track the target motion and output image coordinates of the feature centers for target estimation. By applying the target estimation algorithm and the robotic control approach, if enabled, the incremental joint position is generated for the robotic manipulator. Considering the transmission mechanism and the resolution of stepper motors, the incremental joint position is converted to a number of steps and rotational directions. The commands are then executed via writing to the serial port. It should be noted that if the estimation or control is not enabled, or the ROI is not selected yet, the thread will only loop on displaying the image for visual observation at the moment of experimental setting and/or manual control of the target-manipulator system being performed.

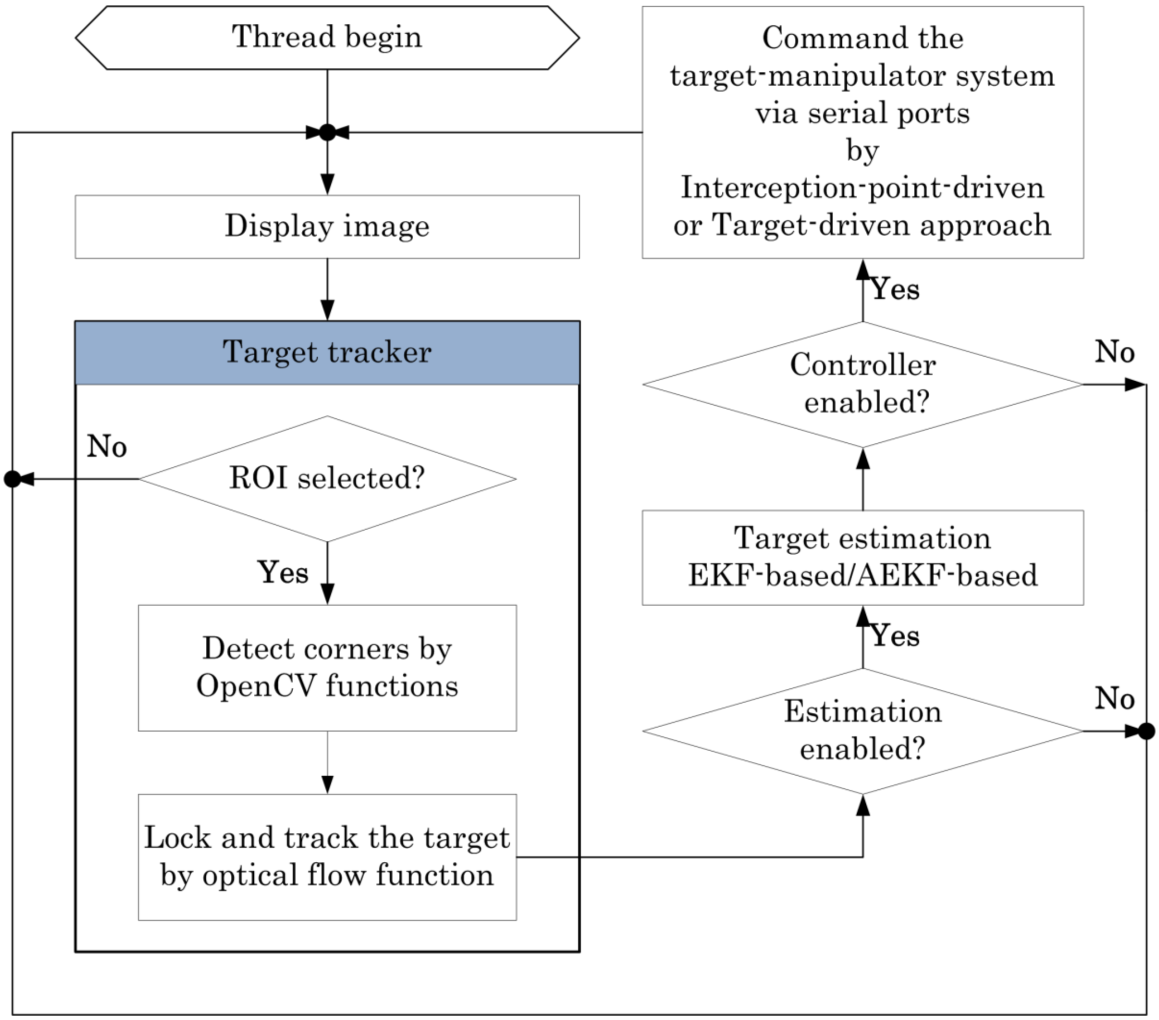

Thread begin
Display image
Target tracker
ROI selected?
No
Yes
Detect corners by OpenCV functions
Lock and track the target by optical flow function
Estimation enabled?
Yes
No
Target estimation EKF-based/AEKF-based
Controller enabled?
Yes
No
Command the target-manipulator system via serial ports by Interception-point-driven or Target-driven approach

Figure 5.19 Workflow of the visual servo robotic control thread

Chapter 6 FLEXIBLE JOINT ROBOTIC MANIPULATOR

Summary: In this chapter, the modeling and control of a single rigid-link flexible joint robotic manipulator is investigated. To validate the proposed control method, hardware and software of a flexible-joint testbed is self-developed. The experimental results and discussion are presented accordingly.

6.1 Flexible Joint Robotic Modeling

In order to adapt the misalignment of shafts, a flexible shaft coupling is adopted in the first joint of the robotic manipulator described in the previous chapter. Therefore, the flexible joint robotic control is investigated here for future extension of the robotic control.

Because the joint position is considered as the control input of robotic manipulator throughout this research, the single rigid link flexible joint robotic manipulator with consideration of damping and friction can be expressed by

$$J\ddot{q}+D\dot{q}+K\left(q-\theta\right)=-M\frac{\dot{q}}{\left|\dot{q}\right|} \tag{6.1}$$

where J denotes the inertia of the link, K denotes the stiffness of the flexible joint, D stands for the damping factor, M stands for the friction factor, q and θ denote rotational angle of the link and motor with respect to the base, respectively. Notes that K, D and M are unknown and should be determined, except J that can be calculated according to the shape and mass distribution of the link. Assume the joint flexibility is induced by two identical springs mounted symmetrically, as illustrated in Figure 6.1.

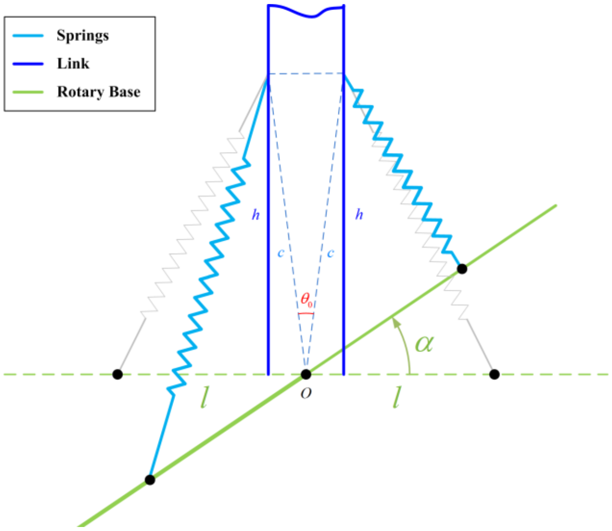


Figure 6.1 Geometry of the flexible joint robotic manipulator

Based on the geometry, the joint stiffness in terms of the angle difference defined by $\alpha = \theta - q$ is obtained by

$$K = -\frac{s_0 k_s h\left(l\cos\alpha - c\sin\dfrac{\theta_0}{2}\right)}{\alpha\sqrt{l^2 + c^2 - 2lc\sin\left(\dfrac{\theta_0}{2} - \alpha\right)}} + \frac{s_0 k_s h\left(l\cos\alpha - c\sin\dfrac{\theta_0}{2}\right)}{\alpha\sqrt{l^2 + c^2 - 2lc\sin\left(\dfrac{\theta_0}{2} + \alpha\right)}} \tag{6.2}$$

where l, c, h, θ_0 are parameters of the spring installation, s_0 denotes the original length of the spring and k_s denotes the stiffness of the spring.

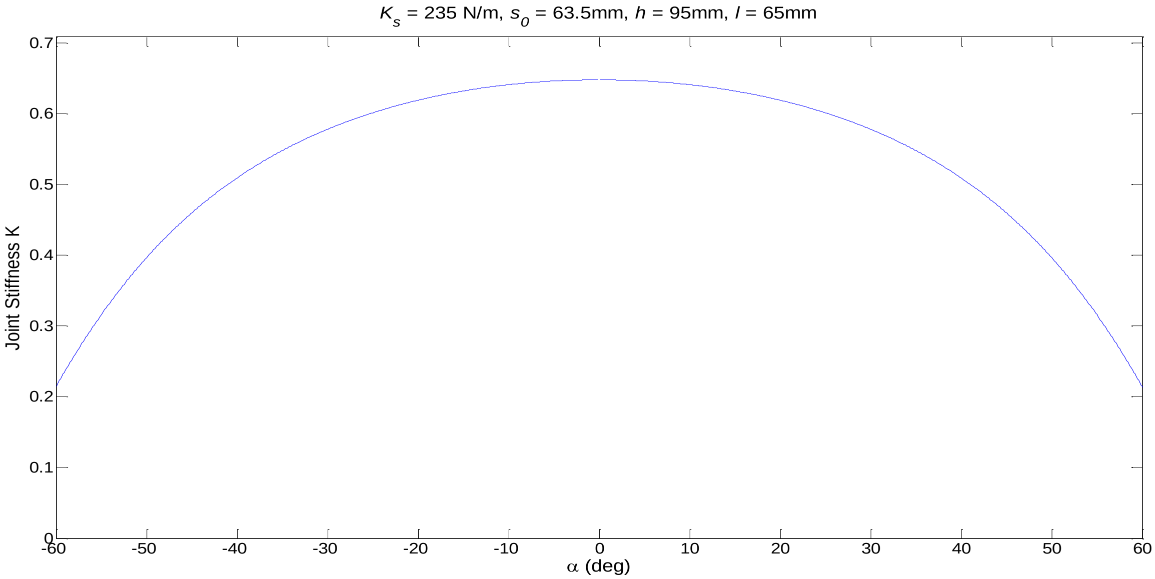


Figure 6.2 Profile of the nonlinear joint stiffness

The profile of the nonlinear joint stiffness versus α is shown in Figure 6.2. Since D and M are related to the physical configuration, they are determined after the testbed is physically built by tuning the parameters in simulation to match the experimental results.

6.2 Flexible Joint Robotic Control

If the natural frequency ϖ and damping ratio ζ of the flexible joint caused vibrations are known, then the percentage residual vibration of input shaping can be described by

$$V(\varpi,\zeta)=e^{-\zeta\varpi t_n}\sqrt{C(\varpi,\zeta)^2+S(\varpi,\zeta)^2} \tag{6.3}$$

$$C(\varpi,\zeta)=\sum_{i=1}^{n}A_i e^{\zeta\varpi t_i}\cos\left(\varpi\sqrt{1-\zeta^2}t_i\right) \tag{6.4}$$

$$S\left(\varpi,\zeta\right)=\sum_{i=1}^{n}A_i e^{\zeta\varpi t_i}\sin\left(\varpi\sqrt{1-\zeta^2}t_i\right) \tag{6.5}$$

where A_i and t_i denote the amplitude and the applied time of a sequence of control input impulses.

According to Figure 6.2, because the joint stiffness is nonlinear and varies to α, the ϖ of the induced vibration will also be a variable. For simplicity, we take the average value of maximum and minimum, denoted by ϖ_0, into account to determine the A_i and t_i for a predefined $V\left(\varpi,\zeta\right)$. A two-hump EI shaper is obtained as per [97]

$$T_d=\frac{2\pi}{\varpi_0\sqrt{1-\zeta^2}} \tag{6.6}$$

$$A_1=A_4=\frac{3\left(\sqrt[3]{V^2\left(\sqrt{1-V^2}+1\right)}\right)^2+2\sqrt[3]{V^2\left(\sqrt{1-V^2}+1\right)}+3V^2}{16\sqrt[3]{V^2\left(\sqrt{1-V^2}+1\right)}} \tag{6.7}$$

$$A_2=A_3=\frac{1}{2}-A_1 \tag{6.8}$$

Therefore, the two-hump EI shaper, denoted by P, is given by

$$P=\begin{bmatrix} A_1 & A_2 & A_3 & A_4 \\ 0 & 0.5T_d & T_d & 1.5T_d \end{bmatrix} \tag{6.9}$$

In order to address the shortcoming of open-loop input shaping, an impedance controller in joint space is developed without the consideration of external force. Define the objective impedance function as

$$m(\ddot{q}_d - \ddot{q}) + b(\dot{q}_d - \dot{q}) + k(q_d - q) = 0 \tag{6.10}$$

where m, b, k are control parameters, and q_d denotes the desired link angle after the treatment of input shaping.

Solving for $\ddot{q}$ from Eq. (6.10) and substituting into Eq. (6.1) lead to

$$\theta = q + \frac{1}{K}\left\{ J\left(\ddot{q}_d + \frac{b}{m}(\dot{q}_d - \dot{q}) + \frac{k}{m}(q_d - q) \right) + D\dot{q} + M\frac{\dot{q}}{|\dot{q}|} \right\} \tag{6.11}$$

Equation (6.11) defines the control input of motor angle. The block diagram of the flexible joint robotic control is shown in Figure 6.3. The desired link angle is treated by the input shaper first, then a closed-loop impedance controller is adopted to generate the joint position control input for the actuator.

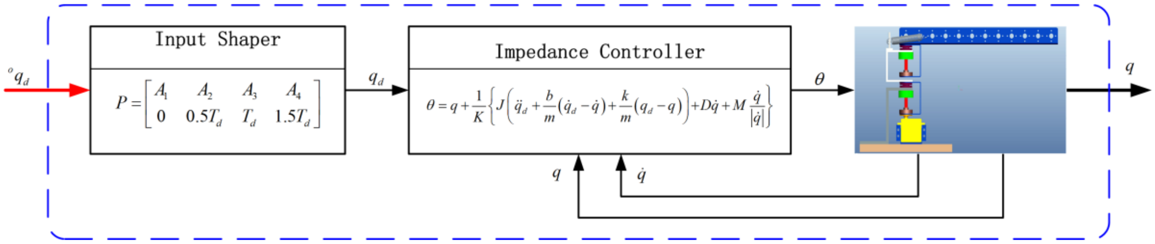

$^{o}q_d$
Input Shaper
$P = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 \\ 0 & 0.5T_d & T_d & 1.5T_d \end{bmatrix}$
q_d
Impedance Controller
$\theta = q + \frac{1}{K}\left\{ J\left(\ddot{q}_d + \frac{b}{m}(\dot{q}_d - \dot{q}) + \frac{k}{m}(q_d - q) \right) + D\dot{q} + M\frac{\dot{q}}{|\dot{q}|} \right\}$
θ
q
q
$\dot{q}$

Figure 6.3 Diagram of the flexible joint robotic control

6.3 Experimental setup

A single rigid-link flexible-joint robotic manipulator is illustrated in Figure 6.4. A servo motor is mounted on the base board, and is employed as the driver of the rotary base. A bearing is mounted between the rotary base and the link, which makes the link only be driven by the two springs. Two shaft encoders are also adopted to record the angular position of motor and the error position between link and rotary base. Because the flexible joint is modeled by two extension springs, the joint flexibility is subject to the error between link angle and motor angle.

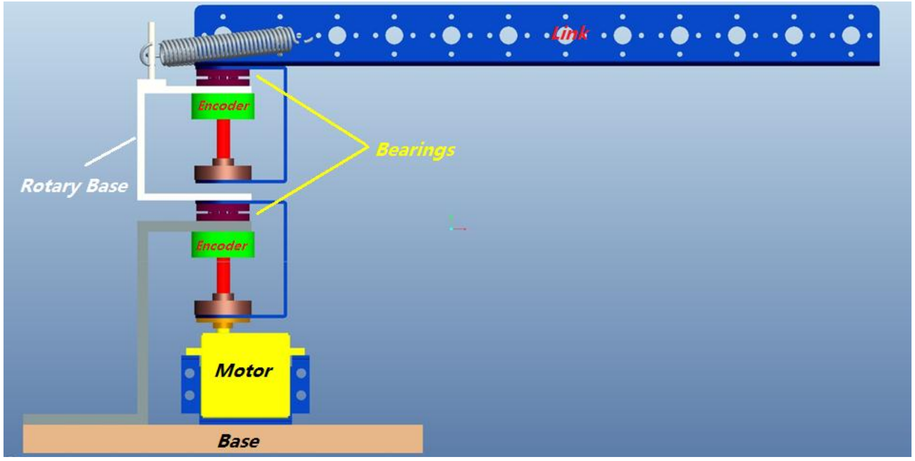


Figure 6.4 PorE model of the flexible joint robotic manipulator

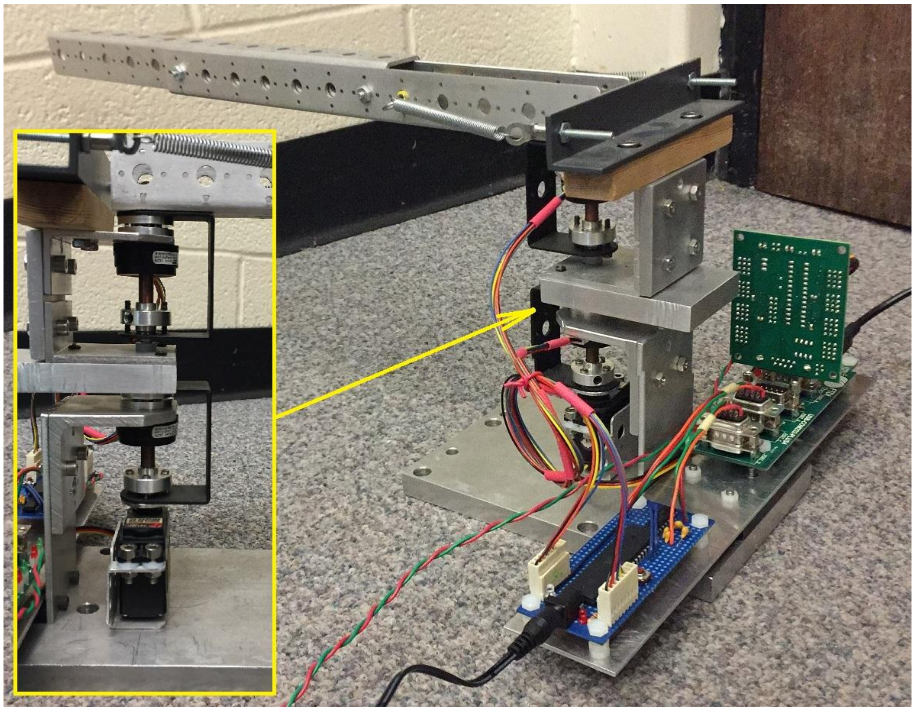


Figure 6.5 Testbed of flexible joint robotic manipulator

According to the mechanical design shown in Figure 6.4, the flexible joint robotic manipulator is then machined and assembled, as shown in Figure 6.5. The hardware and computer software are also developed. The properties of the customized single rigid-link flexible joint robotic manipulator is as listed in Table 6.1.

Table 6.1 Properties of the flexible joint robotic manipulator.

Properties	Values
Maximum torque of the motor	17 kg./cm.
Mass of the rotational base	376.5g
Mass of link	249g
Length of the link	0.4318 m ~ 0.508m
Resolution of the encoders	400 CPR
Baud rate of encoder readers	9600
Baud rate of motor driver (SSC-32)	115200

6.4 Experimental results

Experiments are carried out on the testbed. It can be seen from Figure 6.6 that the flexible joint caused oscillation of the link angle is successfully suppressed by the proposed two-part control scheme. The desired angle of the link with respect to the base is achieved within one second, which is much smaller than the stabilization time of free oscillation. The experimental result demonstrates the effectiveness of the flexible joint robotic control approach.

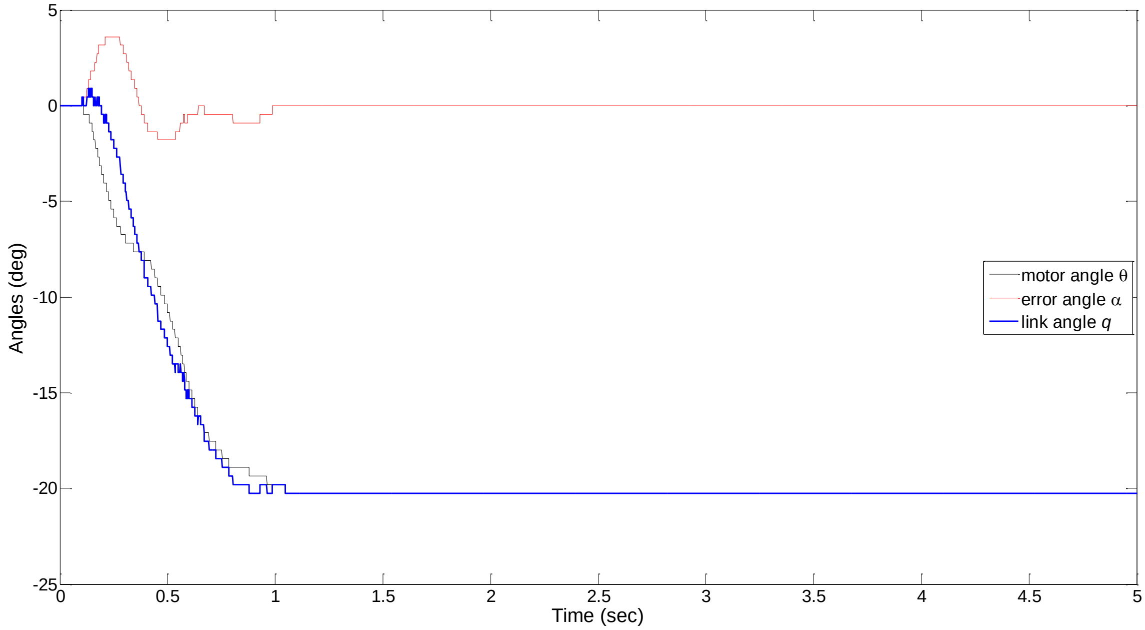


Figure 6.6 Experimental result of flexible joint robotic control

Chapter 7 CONCLUSIONS

Summary: This chapter summarizes the contributions and future research directions for the continuation of the current work.

7.1 Summary of Contributions

This work in this dissertation is focused on the development of vision-based robotic manipulator control for autonomous capture of non-cooperative targets. The contributions are summarized as follows.

7.1.1 Vision-based target estimation

The current work develops an integrated algorithm of EKF and photogrammetry for the position and velocity estimation of non-cooperative targets based on the pinhole camera model. The computational time of the EKF-based approach in each cycle is nearly constant, which is desired in the real time control since this time will affect the sampling rate. However, due to the first order approximation of the target motion, the EKF-based approach may lead to poor performance if the target motion in the camera frame is highly nonlinear. In order to compensate the error in EKF-based approach, an AEKF-based approach is adopted and tested with an eye-in-hand configuration robotic manipulator system, where the estimated motion of the target is coupled with the motion of the robotic manipulator and is highly nonlinear with respect to the camera. The largest estimation errors of the EKF-based approach in the experiment were around 10 centimeters in position and 40 degrees in rotation, where the target motion was generated manually. As a comparison, the AEKF-based approach improves the accuracy to less than 1 centimeter in position and 1 degree in rotation.

7.1.2 Robotic control theory

Two kinematics-based incremental control approaches are proposed for the robotic capture, IPD and TD, to avoid the inverse kinematics in position level by considering the incremental joint position as control input and angular velocity limits as constraints. The IPD approach directly adopts the dynamically predicted interception point as the desired position of the end-effector. It ensures a minimum time capture by moving the end-effector towards the target along a shortest path within each time interval. In contrast, the TD approach adopts the target position in the next time instant as the desired position. It can greatly reduce the possibility of target loss relative to the IPD, when the robotic manipulator equipped with eye-in-hand camera.

7.1.3 Validation

The proposed estimation algorithms and robotic control approaches are validated by hardware-in-the-loop simulation as well as experiments. A custom eye-in-hand configured robotic manipulator is upgraded. Reading circuit of shaft encoders are designed and added to measure the joint position. Furthermore, the electronic hardware and chip software of the robotic manipulator are completely upgraded. A Windows-based computer software with user-friendly operation interface is developed and implemented.

7.1.4 Flexible joint modeling and control

The modeling and control of flexible joint robotic manipulator is investigated for future research of robotic control with joint flexibility. The flexible joint caused oscillation of the link angle is successfully suppressed by the proposed two-part control scheme. In the experiment, a desired link angle of 20 degrees is achieved at around one second, which is much smaller than the stabilization time of free oscillation (around four seconds).

7.2 Future Work

Based on current study, the following research is suggested to continue and expand the current work.

(i) A hybrid eye-in-hand and eye-to-hand configuration is suggested to maximize the advantages of each while overcoming the limitations.

Accordingly, the target estimation can be performed in switch mode or calibration mode for the sake of improvement of accuracy and robustness.

(ii) Free-floating-based robotic manipulator system for space application.

(iii) Post-capture robotic control for stabilizing the target-manipulator system is another important topic in robotic capture.

(iv) Control schemes that consider the flexible links and/or flexible joints to improve the control accuracy and robustness.

(v) Obstacle avoidance and trajectory planning to ensure a safe approach and capture.

Bibliography

[1] D. Rising, "Satellite hits Atlantic—but what about next one?," *Seattle Times*, November 11, 2013.

[2] "Union of Concerned Scientists, Satellite Database," May 01, 2016; http://ucsusa.org/satellites.

[3] K. Wormnes, R. Le Letty, L. Summerer, R. Schonenborg, O. Dubois-Matra, E. Luraschi, A. Cropp, H. Krag, and J. Delaval, "ESA technologies for space debris remediation." pp. 3-4.

[4] M. H. Shan, J. Guo, and E. Gill, "Review and comparison of active space debris capturing and removal methods," *Progress in Aerospace Sciences,* vol. 80, pp. 18-32, Jan, 2016.

[5] M. Jankovic, J. Paul, and F. Kirchner, "GNC architecture for autonomous robotic capture of a non-cooperative target: preliminary concept design," *Advances in Space Research,* vol. 57, no. 8, pp. 1715-1736, 2016.

[6] L. T. DeLuca, F. Bernelli, F. Maggi, P. Tadini, C. Pardini, L. Anselmo, M. Grassi, D. Pavarin, A. Francesconi, F. Branz, S. Chiesa, N. Viola, C.

Bonnal, V. Trushlyakov, and I. Belokonov, “Active space debris removal by a hybrid propulsion module,” *Acta Astronautica,* vol. 91, pp. 20-33, Oct-Nov, 2013.

[7] R. Dudziak, S. Tuttle, and S. Barraclough, “Harpoon technology development for the active removal of space debris,” *Advances in Space Research,* vol. 56, no. 3, pp. 509-527, Aug 1, 2015.

[8] R. Zhong, and Z. H. Zhu, “Dynamics of Nanosatellite Deorbit by Bare Electrodynamic Tether in Low Earth Orbit,” *Journal of Spacecraft and Rockets,* vol. 50, no. 3, pp. 691-700, May-Jun, 2013.

[9] E. T. Lu, and S. G. Love, “Gravitational tractor for towing asteroids,” *Nature,* vol. 438, no. 7065, pp. 177-8, Nov 10, 2005.

[10] D. Reintsema, J. Thaeter, A. Rathke, W. Naumann, P. Rank, and J. Sommer, "DEOS–the German robotics approach to secure and de-orbit malfunctioned satellites from low earth orbits." pp. 244-251.

[11] T. Boge, T. Wimmer, O. Ma, and M. Zebenay, "EPOS− A Robotics-Based Hardware-in-the-Loop Simulator for Simulating Satellite RvD Operations."

[12] T. J. Debus, and S. P. Dougherty, "Overview and performance of the front-end robotics enabling near-term demonstration (FREND) robotic arm." pp. 1-12.

[13] A. Ellery, “A robotics perspective on human spaceflight,” *Earth, Moon,*

and Planets, vol. 87, no. 3, pp. 173-190, 1999.

[14] B. Bischof, L. Kerstein, J. Starke, H. Guenther, and W. P. Foth, "Roger - Robotic geostationary orbit restorer," *Space Debris and Space Traffic Management Symposium 2003,* vol. 109, pp. 183-193, 2004.

[15] P. Huang, J. Cai, Z. Meng, Z. Hu, and D. Wang, "Novel method of monocular real-time feature point tracking for tethered space robots," *Journal of Aerospace Engineering,* vol. 27, no. 6, pp. 04014039, 2013.

[16] G. Dong, and Z. H. Zhu, "Position-based visual servo control of autonomous robotic manipulators," *Acta Astronautica,* vol. 115, pp. 291-302, Oct-Nov, 2015.

[17] M. Robotics, "Mobile Servicing System-Data Sheet," *Brampton, Ontario, Canada: MD Robotics*, 2002.

[18] T. Matsueda, F. Kuwao, S. Motohasi, and R. Okamura, "Development of Japanese experiment module remote manipulator system," 1994.

[19] R. Boumans, and C. Heemskerk, "The European Robotic Arm for the International Space Station," *Robotics and Autonomous Systems,* vol. 23, no. 1-2, pp. 17-27, Mar, 1998.

[20] "Jet Propulsion Laboratory, Viking Fact Sheet," Feburary 3, 2016; http://www.jpl.nasa.gov/news/fact_sheets/viking.pdf.

[21] E. Tunstel, M. Maimone, A. Trebi-Ollennu, J. Yen, R. Petras, and R. Willson, "Mars Exploration Rover mobility and robotic arm operational

performance." pp. 1807-1814.

[22] R. Billing, and R. Fleischner, "Mars Science Laboratory robotic arm."

[23] C. Covault, “Curiosity's MISSION to MARS,” *Aerospace America,* vol. 49, no. 1, pp. 28-34, Jan, 2011.

[24] "Canadian Space Agency, The structure of Canadarm," Feburary 3, 2016; http://www.asc-csa.gc.ca/eng/canadarm/description.asp.

[25] "Defense Advanced Research Projects Agency, Orbital Express Fact Sheet," Feburary 3, 2016; http://archive.darpa.mil/orbitalexpress/pdf/oe_fact_sheet_final.pdf.

[26] G. Hirzinger, B. Brunner, K. Landzettel, N. Sporer, J. Butterfass, and M. Schedl, “Space robotics - DLR's telerobotic concepts, lightweight arms and articulated hands,” *Autonomous Robots,* vol. 14, no. 2-3, pp. 127-145, Mar-May, 2003.

[27] G. Dong, and Z. H. Zhu, “Control of Space Robotic Manipulators with Joint Flexibility,” *Canadian Society for Mechanical Engineering Bulletin*, pp. 11-16, 2013.

[28] K. Yoshida, “Achievements in Space Robotics Expanding the Horizons of Service and Exploration,” *IEEE Robotics & Automation Magazine,* vol. 16, no. 4, pp. 20-28, Dec, 2009.

[29] C. W. Bac, E. J. van Henten, J. Hemming, and Y. Edan, “Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges

Ahead," *Journal of Field Robotics,* vol. 31, no. 6, pp. 888-911, Nov-Dec, 2014.

[30] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences,* vol. 68, pp. 1-26, Jul, 2014.

[31] J. A. Smith, J. Jivraj, R. Wong, and V. Yang, "30 Years of Neurosurgical Robots: Review and Trends for Manipulators and Associated Navigational Systems," *Ann Biomed Eng*, pp. 1-11, Oct 14, 2015.

[32] B. P. Larouche, and Z. H. Zhu, "Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control," *Autonomous Robots,* vol. 37, no. 2, pp. 157-167, Aug, 2014.

[33] G. J. Garcia, J. A. Corrales, J. Pomares, and F. Torres, "Survey of visual and force/tactile control of robots for physical interaction in Spain," *Sensors (Basel),* vol. 9, no. 12, pp. 9689-733, 2009.

[34] 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," *Advanced Robotics,* vol. 18, no. 2, pp. 175-198, 2004.

[35] W. F. Xu, B. Liang, C. Li, and Y. S. Xu, "Autonomous rendezvous and robotic capturing of non-cooperative target in space," *Robotica,* vol. 28, no. 05, pp. 705-718, Sep, 2010.

[36] S. K. Wong, “Non-cooperative target recognition in the frequency domain,” *Iee Proceedings-Radar Sonar and Navigation,* vol. 151, no. 2, pp. 77-84, Apr, 2004.

[37] N. Inaba, M. Oda, and M. Asano, “Rescuing a stranded satellite in space - Experimental robotic capture of non-cooperative satellites,” *Transactions of the Japan Society for Aeronautical and Space Sciences,* vol. 48, no. 162, pp. 213-220, Feb, 2006.

[38] C. Martinez, I. F. Mondragon, M. A. Olivares-Mendez, and P. Campoy, “On-board and Ground Visual Pose Estimation Techniques for UAV Control,” *Journal of Intelligent & Robotic Systems,* vol. 61, no. 1-4, pp. 301-320, Mar, 2011.

[39] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain," *Robotics Research*, pp. 201-212: Springer, 2011.

[40] F. Janabi-Sharifi, and M. Marey, “A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing,” *Ieee Transactions on Robotics,* vol. 26, no. 5, pp. 939-947, Oct, 2010.

[41] G. Caron, A. Dame, and E. Marchand, “Direct model based visual tracking and pose estimation using mutual information,” *Image and Vision Computing,* vol. 32, no. 1, pp. 54-63, Jan, 2014.

[42] G. Schweighofer, and A. Pinz, “Robust pose estimation from a planar target,” *IEEE Trans Pattern Anal Mach Intell,* vol. 28, no. 12, pp. 2024-

30, Dec, 2006.

[43] F. Blais, J.-A. Beraldin, S. F. El-Hakim, and L. Cournoyer, "Real-time geometrical tracking and pose estimation using laser triangulation and photogrammetry." pp. 205-212.

[44] T. Clarke, and X. Wang, “The control of a robot end-effector using photogrammetry,” *International Archives of Photogrammetry and Remote Sensing,* vol. 33, no. B5/1; PART 5, pp. 137-142, 2000.

[45] B. Moghaddam, and A. Pentland, "Probabilistic visual learning for object detection." pp. 786-793.

[46] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking."

[47] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of basic Engineering,* vol. 82, no. 1, pp. 35-45, 1960.

[48] S. Y. Chen, “Kalman Filter for Robot Vision: A Survey,” *Ieee Transactions on Industrial Electronics,* vol. 59, no. 11, pp. 4409-4420, Nov, 2012.

[49] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot manipulator control: theory and practice*: CRC Press, 2003.

[50] N. Hogan, "Impedance control: An approach to manipulation." pp. 304-313.

[51] N. Hogan, “Impedance control: An approach to manipulation: Part II—

Implementation," *Journal of dynamic systems, measurement, and control,* vol. 107, no. 1, pp. 8-16, 1985.

[52] S. Jung, T. C. Hsia, and R. G. Bonitz, "Force tracking impedance control for robot manipulators with an unknown environment: Theory, simulation, and experiment," *International Journal of Robotics Research,* vol. 20, no. 9, pp. 765-774, Sep, 2001.

[53] F. Almeida, A. Lopes, and P. Abreu, "Force-impedance control: a new control strategy of robotic manipulators," *Recent advances in Mechatronics*, pp. 126-37, 1999.

[54] C.-C. Cheah, and D. Wang, "Learning impedance control for robotic manipulators," *Robotics and Automation, IEEE Transactions on,* vol. 14, no. 3, pp. 452-465, 1998.

[55] S. Chan, B. Yao, W. Gao, and M. Cheng, "Robust impedance control of robot manipulators," *International Journal of Robotics & Automation,* vol. 6, no. 4, pp. 220-227, 1991.

[56] R. J. Anderson, and M. W. Spong, "Hybrid Impedance Control of Robotic Manipulators," *Ieee Journal of Robotics and Automation,* vol. 4, no. 5, pp. 549-556, Oct, 1988.

[57] M. C. Chien, and A. C. Huang, "Adaptive control of electrically-driven robot without computation of regressor matrix," *Journal of the Chinese Institute of Engineers,* vol. 30, no. 5, pp. 855-862, Jul, 2007.

[58] A.-C. Huang, and M.-C. Chien, *Adaptive control of robot manipulators: a unified regressor-free approach*: World Scientific, 2010.

[59] B. Yao, and M. Tomizuka, "Adaptive robust control of robot manipulators: Theory and comparative experiments." pp. 442-447.

[60] A. Merabet, and J. Gu, "Advanced Nonlinear Control of Robot Manipulators," Chapter, 2010.

[61] J. Yim, and J. H. Park, "Nonlinear H∞ control of robotic manipulator." pp. 866-871.

[62] Y. Miyasato, "Nonlinear adaptive H∞ control of constrained robotic manipulators with input nonlinearity." pp. 2000-2005.

[63] M. G. Ortega, M. Vargas, C. Vivas, and F. R. Rubio, "Robustness improvement of a nonlinearH∞ controller for robot manipulators via saturation functions," *Journal of Robotic Systems,* vol. 22, no. 8, pp. 421-437, 2005.

[64] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, "State-Space Solutions to Standard H-2 and H-Infinity Control-Problems," *Ieee Transactions on Automatic Control,* vol. 34, no. 8, pp. 831-847, Aug, 1989.

[65] C. Wang, C. Y. Lin, and M. Tomizuka, "Statistical Learning Algorithms to Compensate Slow Visual Feedback for Industrial Robots," *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme,* vol. 137, no. 3, pp. 031011, Mar, 2015.

[66] W. T. Miller, F. H. Glanz, and L. G. Kraft, "Application of a General Learning Algorithm to the Control of Robotic Manipulators," *International Journal of Robotics Research,* vol. 6, no. 2, pp. 84-98, Sum, 1987.

[67] H. Miyamoto, M. Kawato, T. Setoyama, and R. Suzuki, "Feedback-Error-Learning Neural Network for Trajectory Control of a Robotic Manipulator," *Neural Networks,* vol. 1, no. 3, pp. 251-265, 1988.

[68] S. Arimoto, "Learning control theory for robotic motion," *International Journal of Adaptive Control and Signal Processing,* vol. 4, no. 6, pp. 543-564, 1990.

[69] W. He, Y. T. Dong, and C. Y. Sun, "Adaptive Neural Impedance Control of a Robotic Manipulator With Input Saturation," *Ieee Transactions on Systems Man Cybernetics-Systems,* vol. 46, no. 3, pp. 334-344, Mar, 2016.

[70] D. Kragic, and H. I. Christensen, "Survey on visual servoing for manipulation," *Computational Vision and Active Perception Laboratory, Fiskartorpsv,* vol. 15, 2002.

[71] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *Ieee Transactions on Robotics and Automation,* vol. 12, no. 5, pp. 651-670, Oct, 1996.

[72] F. Chaumette, and S. Hutchinson, "Visual servo control. I. Basic approaches," *Robotics & Automation Magazine, IEEE,* vol. 13, no. 4, pp.

82-90, 2006.

[73] F. Chaumette, and S. Hutchinson, “Visual servo control - Part II: Advanced approaches,” *Ieee Robotics & Automation Magazine,* vol. 14, no. 1, pp. 109-118, Mar, 2007.

[74] A. De Luca, M. Ferri, G. Oriolo, and P. R. Giordano, "Visual servoing with exploitation of redundancy: An experimental study." pp. 3231-3237.

[75] P. Gasbarri, M. Sabatini, and G. B. Palmerini, “Ground tests for vision based determination and control of formation flying spacecraft trajectories,” *Acta Astronautica,* vol. 102, pp. 378-391, Sep-Oct, 2014.

[76] M. Sabatini, R. Monti, P. Gasbarri, and G. Palmerini, “Deployable space manipulator commanded by means of visual-based guidance and navigation,” *Acta Astronautica,* vol. 83, pp. 27-43, Feb-Mar, 2013.

[77] M. Sabatini, R. Monti, P. Gasbarri, and G. B. Palmerini, “Adaptive and robust algorithms and tests for visual-based navigation of a space robotic manipulator,” *Acta Astronautica,* vol. 83, pp. 65-84, Feb-Mar, 2013.

[78] H. S. Wang, Y. H. Liu, and W. D. Chen, “Visual tracking of robots in uncalibrated environments,” *Mechatronics,* vol. 22, no. 4, pp. 390-397, Jun, 2012.

[79] C. Cai, E. Dean-León, N. Somani, and A. Knoll, "3D image-based dynamic visual servoing with uncalibrated stereo cameras." pp. 1-6.

[80] M. Sabatini, P. Gasbarri, R. Monti, and G. B. Palmerini, “Vibration control of a flexible space manipulator during on orbit operations,” *Acta Astronautica,* vol. 73, pp. 109-121, Apr-May, 2012.

[81] E. Malis, F. Chaumette, and S. Boudet, “2½D visual servoing,” *Robotics and Automation, IEEE Transactions on,* vol. 15, no. 2, pp. 238-250, 1999.

[82] S. Cocuzza, I. Pretto, and S. Debei, “Least-Squares-Based Reaction Control of Space Manipulators,” *Journal Of Guidance Control And Dynamics,* vol. 35, no. 3, pp. 976-986, May-Jun, 2012.

[83] Y. Wang, J. Luo, W. Li, H. Tan, and Y. Zhang, "Inverse-free D1G1 solution to acceleration-level inverse kinematics of redundant robot manipulators." pp. 280-284.

[84] S. Chiaverini, “Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators,” *Ieee Transactions on Robotics And Automation,* vol. 13, no. 3, pp. 398-410, Jun, 1997.

[85] Y. Umetani, and K. Yoshida, “Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix,” *Ieee Transactions on Robotics and Automation,* vol. 5, no. 3, pp. 303-314, Jun, 1989.

[86] Z. Vafa, and S. Dubowsky, "On the dynamics of manipulators in space using the virtual manipulator approach." pp. 579-585.

[87] B. Liang, Y. S. Xu, and M. Bergerman, “Mapping a space manipulator to a dynamically equivalent manipulator,” *Journal of Dynamic Systems*

Measurement and Control-Transactions of the Asme, vol. 120, no. 1, pp. 1-7, Mar, 1998.

[88] S. Ozgoli, and H. D. Taghirad, “A survey on the control of flexible joint robots,” *Asian Journal of Control,* vol. 8, no. 4, pp. 332-344, Dec, 2006.

[89] A. Benosman, and G. Le Vey, “Control of flexible manipulators: A survey,” *Robotica,* vol. 22, no. 05, pp. 533-545, Sep-Oct, 2004.

[90] T. Yoshikawa, and K. Hosoda, “Modeling of flexible manipulators using virtual rigid links and passive joints,” *International Journal of Robotics Research,* vol. 15, no. 3, pp. 290-299, Jun, 1996.

[91] K. Senda, and Y. Murotsu, “Methodology for control of a space robot with flexible links,” *Iee Proceedings-Control Theory and Applications,* vol. 147, no. 6, pp. 562-568, Nov, 2000.

[92] S. Ulrich, J. Z. Sasiadek, and I. Barkana, “Nonlinear Adaptive Output Feedback Control of Flexible-Joint Space Manipulators with Joint Stiffness Uncertainties,” *Journal of Guidance Control and Dynamics,* vol. 37, no. 6, pp. 1961-1975, Nov, 2014.

[93] M. C. Chien, and A. C. Huang, “Adaptive impedance controller design for flexible-joint electrically-driven robots without computation of the regressor matrix,” *Robotica,* vol. 30, no. 01, pp. 133-144, Jan, 2012.

[94] H. Taghirad, and M. Shaterian, "Nonlinear H∞ controller design for flexible joint robots." pp. 1391-1396.

[95] Z. H. Jiang, and S. Higaki, “Control of flexible joint robot manipulators using a combined controller with neural network and linear regulator,” *Proceedings of the Institution of Mechanical Engineers Part I-Journal of Systems and Control Engineering,* vol. 225, no. I6, pp. 798-806, Sep, 2011.

[96] N. C. Singer, and W. P. Seering, “Preshaping Command Inputs to Reduce System Vibration,” *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme,* vol. 112, no. 1, pp. 76-82, Mar, 1990.

[97] W. E. Singhose, “Command generation for flexible systems,” Citeseer, 1997.

[98] N. C. Singer, “Residual vibration reduction in computer controlled machines,” 1989.

[99] T. Singh, and W. Singhose, "Input shaping/time delay control of maneuvering flexible structures." pp. 1717-1731.

[100] J. Vaughan, and W. Singhose, "Input shapers for reducing overshoot in human-operated flexible systems." pp. 178-183.

[101] K. Kozak, W. Singhose, and I. Ebert-Uphoff, “Performance measures for input shaping and command generation,” *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme,* vol. 128, no. 3, pp. 731-736, Sep, 2006.

[102] K. A. Myers, and B. D. Tapley, “Adaptive Sequential Estimation with

Unknown Noise Statistics," *Ieee Transactions on Automatic Control,* vol. 21, no. 4, pp. 520-523, 1976.

[103] V. Lippiello, B. Siciliano, and L. Villani, "Adaptive extended Kalman filtering for visual motion estimation of 3D objects," *Control Engineering Practice,* vol. 15, no. 1, pp. 123-134, Jan, 2007.

[104] I. Blanchet, C. Frankignoul, and M. A. Cane, "A comparison of adaptive kalman filters for a tropical Pacific Ocean model," *Monthly Weather Review,* vol. 125, no. 1, pp. 40-58, Jan, 1997.

[105] Q. Song, and J.-D. Han, "An adaptive UKF algorithm for the state and parameter estimations of a mobile robot," *Acta Automatica Sinica,* vol. 34, no. 1, pp. 72-79, 2008.

[106] T. Schenk, "Introduction to photogrammetry," *The Ohio State University, Columbus*, 2005.

[107] B. P. Larouche, *Autonomous intelligent robotic manipulator for on-orbit servicing*: York University, 2012.

[108] S. Kucuk, and Z. Bingul, *Robot kinematics: forward and inverse kinematics*: INTECH Open Access Publisher, 2006.

[109] J. Denavit, "A kinematic notation for lower-pair mechanisms based on matrices," *Trans. of the ASME. Journal of Applied Mechanics,* vol. 22, pp. 215-221, 1955.

[110] M. W. Spong, and M. Vidyasagar, *Robot dynamics and control*: John

Wiley & Sons, 2008.

[111] J. J. Craig, *Introduction to robotics: mechanics and control*: Pearson Prentice Hall, 2005.

[112] "Lynxmotion Inc. SSC-32 Servo Controller," May 31, 2016; http://www.lynxmotion.com/p-395-ssc-32-servo-controller.aspx.

Reference Paper A

Acta Astronautica 115 (2015) 291–302

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Position-based visual servo control of autonomous robotic manipulators

Gangqi Dong, Z.H. Zhu*

Department of Earth and Space Science and Engineering, York University, 4700 Keele Street, Toronto, Ontario, Canada M3J 1P3

ARTICLE INFO

Article history:
Received 22 December 2014
Received in revised form
27 April 2015
Accepted 22 May 2015
Available online 7 June 2015

Keywords:
Autonomous capture
Non-cooperative target
Extend Kalman filter
Robotic manipulator
Position-based visual servo
On-orbit servicing

ABSTRACT

This paper concerns the position-based visual servo control of autonomous robotic manipulators in space. It focuses on the development of a real-time vision-based pose and motion estimation algorithm of a non-cooperative target by photogrammetry and extended Kalman filter for robotic manipulators to perform autonomous capture. Optical flow algorithm is adopted to track the target features in order to improve the image processing efficiency. Then, a close-loop position-based visual servo control strategy is devised to determine the desired pose of the end-effector at the rendezvous point based on the estimated pose and motion of the target. The corresponding desired joint angles of the robotic manipulator in the joint space are derived by the inverse kinematics of the robotic manipulator. The developed algorithm and position-based visual servo control strategy are validated experimentally by a custom built robotic manipulator with an eye-in-hand configuration. The experimental results demonstrate the proposed estimation algorithm and control scheme are feasible and effective.

1. Introduction

Robotic manipulators have been widely used in space for docking, assembling, repairing and other on-orbit servicing operations [1–5]. For instance, Mobile Servicing System (MSS) or Canadarm2 [6], Japanese Experiment Module Remote Manipulator System (JEMRMS) [7] and European Robotic Arm (ERA) [8] are typical examples of robotic manipulators performing assembly, maintenance, and payloads exchanging tasks on International Space Station. These operations were conducted either autonomously or by human astronauts. Robotic manipulators mounted on Mars exploration rovers, such as, Viking 1 and 2 [9], Spirits and Opportunity [10], Phoenix [11] and Curiosity [12], were designed to collect soil samples and/or place instruments on a target. These tasks were performed by preprogrammed commands and controlled from the Earth directly or relayed by the Mars Orbiter. Cameras were used in these missions to monitor the movements of the manipulators and take photographs of the surroundings. Robotic manipulators of orbital docking systems, such as the Shuttle Remote Manipulator System [13] and Orbital Express [14], performed tasks of grappling, docking, refueling, repairing and/or servicing another spacecraft. Pure experimental systems, such as, ROTEX (Robot Technology Experiment) and ETS-VII (Engineering Test Satellite) [15] demonstrated the operations of assembling, grasping, docking and exchanging orbit replaceable units by robotic manipulators. Most of these missions employed human-in-the-loop control. Manual control from the Earth may result in long time delay, while sending astronauts into space to perform the tasks suffers higher cost and the possibility of life loss. To address these challenges, autonomous control is required and becomes a research highlight in the field of robotic technology [16,17].

* Corresponding author. Tel.: +1 416 7362100x77729.
E-mail address: gzhu@yorku.ca (Z.H. Zhu).

http://dx.doi.org/10.1016/j.actaastro.2015.05.036
0094-5765/© 2015 IAA. Published by Elsevier Ltd. All rights reserved.

Nomenclature

$\boldsymbol{A}$	system transformation matrix
$\boldsymbol{B}$	process noise coefficient matrix
$\boldsymbol{E}$	residual error in the measurement
f	focal length of the camera
$\boldsymbol{H}$	Jacobian matrix of measurement model
$\boldsymbol{I}$	identity matrix
$\boldsymbol{J_c}$	Jacobian matrix of the pin-hole camera model
$\boldsymbol{J_r}$	Jacobian matrix of the robotic manipulator
$\boldsymbol{P}$	covariance matrix of system state variable
$\boldsymbol{Q}$	covariance matrix of process noise
$\boldsymbol{R}$	covariance matrix of measurement noise
t	sample time
$\boldsymbol{X}$	state variable
$\boldsymbol{Z}$	measurement vector
$(\cdot)^{\boldsymbol{c}}$	control vector
$(\cdot)^{\boldsymbol{d}}$	desired vector
$(\cdot)^{\boldsymbol{t}}$	vector of the target
$(\bullet)_C$	coordinates in camera frame
$(\bullet)_m$	image coordinates
$(\bullet)_T$	coordinates in target frame
$(\bullet)_{To}$	coordinates of target frame origin in camera frame
d_6	variable of robotic gripper
$\boldsymbol{dX_p}$	correction term of the photogrammetry
$\boldsymbol{K_g}$	Kalman gain
$\boldsymbol{R_{TC}}$	rotational matrix from target frame to camera frame
$\boldsymbol{R_{CT}}$	rotational matrix from camera frame to target frame
$\boldsymbol{T_{cg}}$	transformation matrix from camera frame to global frame
$\boldsymbol{X_c}$	position of the grasp point in camera frame
$\boldsymbol{X_g}$	position of the grasp point in global frame
$\boldsymbol{X_p}$	photogrammetry estimated target pose in camera frame
$\boldsymbol{\delta}$	vector of image error
ε	pre-set tolerance of photogrammetry
$\boldsymbol{\mu}$	measurement noise vector
$\boldsymbol{\omega}$	process noise vector
$\boldsymbol{\Theta}$	main robotic joint angle vector
ΔT	estimated time for capture
$\theta_{1,2,3,4,5}$	angle of torso, shoulder, elbow, wrist roll and wrist yaw joints
$\theta_{x,y,z}$	target orientation refer to x, y, z axes of camera frame

Autonomous control of robotic manipulator to track and grasp a moving target requires the precise knowledge of the target's pose and motion. Because of the non-intrusive, non-damaging and non-contact nature, computer vision is favored exclusively as a sensing system to obtain the required information [2,16,18–22]. Accordingly, visual servo control system has been developed to control the pose of manipulator's end-effector with respect to the target based on the feedback of vision system. For instance, the position of a known moving object in the image plane can be tracked with a single mobile camera based on past images and past control inputs to the mobile platform [23]. The autonomous capture of a non-cooperative target by a robotic manipulator requires not only to track the motion of target [24,25] but also to predict the rendezvous point and follow a specific approaching trajectory by the end-effector based on the estimated pose and motion of the target [16,19].

The camera configuration in a visual servo robotic system can be either eye-in-hand or eye-to-hand [26]. The eye-in-hand camera is mounted on the end-effector to provide a close and precise view of the target while the eye-to-hand camera is installed beside the robot to monitor the whole workspace with a broad and relative less accurate scene of the target [27]. Based on the errors employed in control, the robotic visual servo may be categorized as: image-based, position-based, and hybrid visual servo [28,29]. The image-based visual servo (IBVS) controls robots by the error between the projected desired and actual positions in the 2D (two dimensional) image plane via an image Jacobian without reconstruction of the target. Thus, it is free from target model errors and less sensitive to camera calibration errors and measurement noise in images. Considerable efforts [30,31] have been devoted to track moving targets in 3D (three dimensional) space with eye-in-hand cameras using IBVS. Extended Kalman filter was introduced into the IBVS algorithm to address the navigation errors and actuation delays [32]. The perturbation to eye-in-hand cameras by the flexibility of robotic manipulator [33] was investigated to enhance the robustness of IBVS algorithm. However, the IBVS lacks 3D depth information of a target and additional measure is required to estimate the depth. The position-based visual servo (PBVS) controls the error between the desired and actual poses and motion of the end-effector directly in the 3D workspace. The advantage of the PBVS is that the pose of end-effector can be controlled relative to the target directly and naturally, while the drawbacks are that the pose and motion estimation is prone to camera calibration errors, target model accuracy, and image measurement noise. These challenges have been successfully addressed by many researchers to eliminate image errors caused by an uncalibrated camera [34,35] and suppress the image noise due to the vibration of camera resulting from flexible manipulators [36]. Finally, the hybrid visual servo, referred as 2½D visual servo in the literature, evaluates the control errors partially in the 3D workspace and partially on the 2D image plane. Although effective, the hybrid system is generally more complex than either IBVS or PBVS for implementation. In the current work, we adopted a single, calibrated and eye-in-hand camera with PBVS to simplify the system configuration and implementation in autonomous capture of non-cooperative targets.

The key issue in the autonomous capture of non-cooperative targets by PBVS robotic manipulator is the estimation of target's pose and motion with visual feedback to reconstruct the target in 3D space. The pose and motion estimated by the eye-in-hand camera in PBVS are prone to image jittering, residual vibration and unexpected disturbances of camera or the end-effector. Considerable efforts have been devoted to extract information from visual images in the past decades [37–40]. Different methodologies have been developed, such as, analytic or geometric method, learning-based method, offline method and filtering method. The geometric method, such as the photogrammetry, is widely used when the camera is properly calibrated and the target is known. It extracts six degrees of freedom information (pose) of the target from 2D images. However, the geometric method is memoryless and its result is noisy if the image data is not smooth either due to the jittering of image processing or the mechanical disturbance of the end-effector in case of eye-in-hand camera. The learning-based method requires a sufficient large set of images of target in different poses, which is usually not available when the target is non-cooperative. The offline methods are not suitable for the robotic manipulator to track, approach and capture the target in real-time. Another potential solution for the estimation of pose and motion is filtering. Kalman filter (KF) is a widely used filtering algorithm to estimate unknown variables based on a set of noisy measurements observed over time with initial conditions. Since being proposed in 1960 [41], the KF has been applied widely with many variations and extensions beyond the original proposed linear system [42]. Although the initial conditions do not change the convergence property of the KF, they do affect the performance of the filter, especially when dealing with the non-cooperative target in real-time where the initial conditions are unknown.

The focus of this study is to improve the estimation of pose and motion in real-time PBVS control to address the challenges of image jittering and disturbances of camera caused by the flexibility in joints, actuation delays and rough motion of stepper motor. In our previous PBVS work [16], the challenges were studied by a dual KF approach where the first KF was introduced in the image space and the second KF was used in the 3D space. The first KF not only reduces the image noise due to jittering to prevent the errors propagating into photogrammetry algorithm but also provides substitute image data momentarily to enhance the tracking robustness in case of an outrage in the vision system occurs. The second KF was designed to suppress impact of the residual vibration and unexpected distribution of the camera and the sudden motion of a target. Although effective, the approach is computational cumbersome and increases the actuation delay. To address the issue, a new methodology was developed in the current work to integrate the photogrammetry and extended Kalman filter (EKF) to estimate the target's pose and motion in real-time for PBVS. Combining with the inner closed-loop control of robotic manipulator, the visual estimation of pose and motion of a target as well as the control errors in 3D space forms an outer closed-loop control. It is worth pointing out that the effectiveness of EKF in improving the robustness of visual servo [32,33] has also been demonstrated in IBVS. For instance, the EKF introduced in the image space not only improves the estimation accuracy of target's kinematics state but also substitutes state estimates when the target is lost locking [33]. These works showed that the introduction of EKF into IBVS is effective to prevent the tracking failure in a dynamic situation due to image noises and actuation delay. The current approach is validated experimentally by a custom-built robotic manipulator [16] with an eye-in-hand camera mounted closely to the end-effector. The experimental results demonstrate the effectiveness and robustness of the proposed approach by successfully tracking, approaching and capturing a non-cooperative target autonomously.

2. Position-based visual servo of robotic manipulator

2.1. Camera model and photogrammetry

The pose of a target can be described by the Cartesian coordinates $\{x_{To}, y_{To}, z_{To}\}^T$ of a target-fixed frame origin and the Euler angles of that frame $\{\theta_x, \theta_y, \theta_z\}^T$ regarding to the camera frame. The rotational matrix from the camera frame to the target frame, $\boldsymbol{R_{CT}}$, can be developed by rotating x-axis of the camera frame by θ_x first and followed by rotating y-axis of the camera frame by θ_y and rotating z-axis of the camera frame by θ_z. Accordingly, the rotational matrix from the target frame to the camera frame can be expressed as $\boldsymbol{R_{TC}} = \boldsymbol{R_{CT}}^T$, such that

$$\boldsymbol{R_{TC}} = \begin{bmatrix} C_yC_z & -C_yS_z & S_y \\ C_xS_z + S_xS_yC_z & C_xC_z - S_xS_yS_z & -S_xC_y \\ S_xS_z - C_xS_yC_z & S_xC_z + C_xS_yS_z & C_xC_y \end{bmatrix} \tag{1}$$

Here, $S_x = \sin\theta_x$, $S_y = \sin\theta_y$, $S_z = \sin\theta_z$ $C_x = \cos\theta_x, C_y = \cos\theta_y, C_z = \cos\theta_z$.

Assume the coordinates, $\{x_T, y_T, z_T\}^T$, of a feature point on the target are known in the target frame, which implies the vision system is calibrated in advance. Then, the homogeneous relationship between the target and camera frames can be described by

$$\begin{Bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{Bmatrix} = \begin{bmatrix} & & & x_{To} \\ & \boldsymbol{R_{TC}} & & y_{To} \\ & & & z_{To} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{Bmatrix} \tag{2}$$

where $\{x_C, y_C, z_C\}^T$ is the coordinates of the same point in the camera frame.

Consider a pinhole camera model as shown in Fig. 1, denote r_{ij} for elements of $\boldsymbol{R_{TC}}$. The feature point on the target is projected onto the image plane by Eq. (3), such as

$$\begin{cases} x_m = -f\dfrac{x_C}{y_C - f} = -f\dfrac{r_{11}x_T + r_{12}y_T + r_{13}z_T + x_{To}}{r_{21}x_T + r_{22}y_T + r_{23}z_T + y_{To} - f} \\ z_m = -f\dfrac{z_C}{y_C - f} = -f\dfrac{r_{31}x_T + r_{32}y_T + r_{33}z_T + z_{To}}{r_{21}x_T + r_{22}y_T + r_{23}z_T + y_{To} - f} \end{cases} \tag{3}$$

where f stands for the focal length of the camera and $\{x_m, z_m\}^T$ denotes the projected image coordinates of the feature point. For a calibrated camera, the focal length is known in advance.

Defining $\eta = y_C - f$ in Eq. (3) leads to

$$\begin{cases} \eta x_m + x_C f = 0 \\ \eta z_m + z_C f = 0 \end{cases} \tag{4}$$

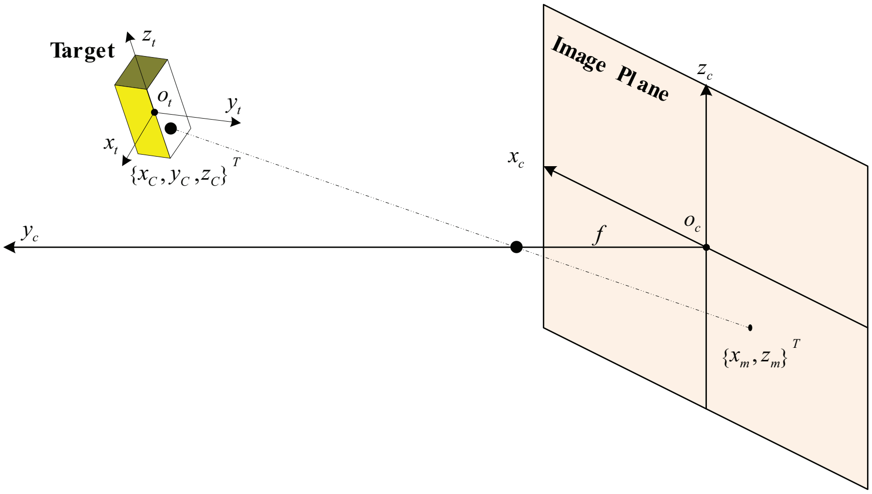


Fig. 1. Pin-hole camera model.

Further define the left side of Eq. (4) by $F=\eta x_m + x_C f$ and $G=\eta z_m + z_C f$, where F and G are functions of the target pose in the camera frame and the projected image coordinates of the feature point on the target. Linearizing F and G by the Taylor expansion in the vicinity of $\{x_m, z_m, x_{To}, y_{To}, z_{To}, \theta_x, \theta_y, \theta_z\}^T$ leads to

$$\boldsymbol{E}_{2\times1}+\boldsymbol{\delta}_{2\times1}=\boldsymbol{J_c}_{\,2\times6}\boldsymbol{dX_p}_{\,6\times1} \tag{5}$$

where

$$\boldsymbol{E}_{2\times1}=\begin{Bmatrix} dx_m \\ dz_m \end{Bmatrix}, \quad \boldsymbol{\delta}_{2\times1}=\frac{1}{\eta}\begin{Bmatrix} F_0 \\ G_0 \end{Bmatrix},$$

$$\boldsymbol{dX_p}_{\,6\times1}=\{dx_{To}, dy_{To}, dz_{To}, d\theta_x, d\theta_y, d\theta_z\}^T,$$

$$\boldsymbol{J_c}_{\,2\times6}=-\frac{1}{\eta}\begin{Bmatrix} \left(\frac{\partial F}{\partial x_{To}}\right)_0 & \left(\frac{\partial F}{\partial y_{To}}\right)_0 & \left(\frac{\partial F}{\partial z_{To}}\right)_0 & \left(\frac{\partial F}{\partial \theta_x}\right)_0 & \left(\frac{\partial F}{\partial \theta_y}\right)_0 & \left(\frac{\partial F}{\partial \theta_z}\right)_0 \\ \left(\frac{\partial G}{\partial x_{To}}\right)_0 & \left(\frac{\partial G}{\partial y_{To}}\right)_0 & \left(\frac{\partial G}{\partial z_{To}}\right)_0 & \left(\frac{\partial G}{\partial \theta_x}\right)_0 & \left(\frac{\partial G}{\partial \theta_y}\right)_0 & \left(\frac{\partial G}{\partial \theta_z}\right)_0 \end{Bmatrix}.$$

Eq. (5) contains two independent equations for six unknowns (pose of target). Theoretically, one needs only three feature points to solve for the six unknowns. However, this approach may result in four ambiguous poses [28,29]. To eliminate the ambiguity and increase the robustness, minimum four feature points are widely adopted in literature, which leads to eight equations with six unknowns, such that

$$\boldsymbol{E}_{8\times1}+\boldsymbol{\delta}_{8\times1}=\boldsymbol{J_c}_{\,8\times6}\boldsymbol{dX_p}_{\,6\times1} \tag{6}$$

The unknowns are solved by an iterative least square approach assuming the zero residual error in the measurement, such that

$$\boldsymbol{E}_{8\times1}=0, \quad \boldsymbol{dX_p}_{\,6\times1}=(\boldsymbol{J_c^T}_{\,8\times6}\boldsymbol{J_c}_{\,8\times6})^{-1}\boldsymbol{J_c^T}_{\,8\times6}\boldsymbol{\delta}_{8\times1} \tag{7}$$

By inputting the known image coordinates of feature points and an initial guess of the target pose, the algorithm iterates to correct previous guess until the correction is less than a pre-set tolerance, i.e., $\|\boldsymbol{dX_p}\| \le \varepsilon$. In practice, it is common to use the previous target pose as the initial guess to reduce the iterations. As aforementioned, the photogrammetry is memoryless and prone to the image noise, which may result in large fluctuation of estimated target pose. Thus, the computational time of photogrammetry may increase if the initial guess, which is the previous pose as mentioned, is far away from the current pose. As a result, the sampling time-step of vision system may be adjusted, which is not desirable when dealing with real-time pose estimation. Another shortcoming of the photogrammetry is that it does not estimate the motion of target directly, which is an important parameter for trajectory planning of the robotic manipulator to perform autonomous capture in a dynamic environment. To address these challenges, an extended Kalman filter (EKF) with photogrammetry is presented in the following.

2.2. *Extended Kalman filter*

The Kalman filter is an optimal estimation algorithm for a linear system with independent white noise of normal distribution [41,42]. The camera model in Eq. (3) is highly nonlinear and the extended Kalman filter has been adopted to estimate the pose and motion of a dynamic target. Assume the motion of the target is approximately constant within the sampling interval if the sampling time-step is sufficiently small. Then, the motion of the target can be approximated by the first order equation of motion. Define the system variable vector as

$$\boldsymbol{X}=\{x_{To}, \dot{x}_{To}, y_{To}, \dot{y}_{To}, z_{To}, \dot{z}_{To}, \theta_x, \dot{\theta}_x, \theta_y, \dot{\theta}_y, \theta_z, \dot{\theta}_z\}^T$$

By assuming the system's acceleration vector $\boldsymbol{\omega}=\{\ddot{x}_{To}, \ddot{y}_{To}, \ddot{z}_{To}, \ddot{\theta}_x, \ddot{\theta}_y, \ddot{\theta}_z\}^T$ as the process noise with a normal distribution with zero mean and covariance matrix $\boldsymbol{Q}$, the system model is expressed as

$$\boldsymbol{X}_k=\boldsymbol{A}\boldsymbol{X}_{k-1}+\boldsymbol{B}\boldsymbol{\omega}_{k-1} \tag{8}$$

where

$$\boldsymbol{A}=diag\begin{bmatrix}\boldsymbol{A_1} & \boldsymbol{A_1} & \boldsymbol{A_1} & \boldsymbol{A_1} & \boldsymbol{A_1} & \boldsymbol{A_1}\end{bmatrix}, \quad \boldsymbol{A_1}=\begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix},$$

$$\boldsymbol{B} = diag\left[\boldsymbol{B_1} \quad \boldsymbol{B_1} \quad \boldsymbol{B_1} \quad \boldsymbol{B_1} \quad \boldsymbol{B_1} \quad \boldsymbol{B_1}\right], \quad \boldsymbol{B_1} = \begin{Bmatrix} t^2/2 \\ t \end{Bmatrix}.$$

k is sample time-step and t is the sample time.

The measurement model is developed from the pin-hole camera model in Eq. (3) for n feature points, such that

$$\boldsymbol{Z}_k = \boldsymbol{h}(\boldsymbol{X}_k) + \boldsymbol{\mu}_k$$
$$\boldsymbol{Z} = \{x_{m1}, z_{m1}, \cdots x_{mn}, z_{mn}\}^T, \quad \boldsymbol{h}(\cdot) = \left\{h_{x1}, h_{z1}, \cdots h_{xn}, h_{zn}\right\}^T \tag{9}$$

and

$$h_{xi}(\boldsymbol{X}) = -f\frac{x_{Ci}}{\eta_i}, \quad h_{zi}(\boldsymbol{X}) = -f\frac{z_{Ci}}{\eta_i} \tag{10}$$

where $\boldsymbol{\mu}$ is the measurement noise that obeys a normal distribution with zero mean and covariance matrix $\boldsymbol{R}$.

The EKF requires initial conditions and measurements observed over time. Since initial conditions of a non-cooperative target are unknown, an inappropriate initial guess may lead to poor performance of the EKF. To improve its performance and accelerate the convergence rate, we initialize the state variable vector by the photogrammetry in the algorithm. Based on the above definitions, a recursive pose and motion estimation algorithm is derived as shown in Table 1, where $\boldsymbol{H}$ is the Jacobian matrix formed by the first order partial differential of the measurement model respect to system variable, $\boldsymbol{K_g}$ is the Kalman gain at time step k, $\boldsymbol{P}$ is the covariance matrix of the system state variable, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the process and measurement noise covariance matrices.

2.3. Kinematics of robotic manipulator

The autonomous capture will be conducted by a custom-built six degrees of freedom (6DOF) robotic manipulator with an eye-in-hand configuration as shown in Fig. 2. The robotic manipulator consists of three links and one end-effector with five revolute and one prismatic joints. The eye-in-hand camera is mounted closely to the end-effector. The first three revolute joints, namely torso (θ_1), shoulder (θ_2) and elbow (θ_3), control the position of the end-effector while the last two revolute joints and one prismatic joint, namely wrist roll (θ_4), wrist yaw (θ_5) and gripper (d_6), provide dexterous orientation and griping function for capture operation. Thus, the translation and rotation of the end-effector can be considered separately from the wrist to simplify the controller design.

The kinematics of robotic manipulator provides the forward relationship from the joint angles to the position of the end-effector (the wrist center), such that

$$\boldsymbol{X_g} = \boldsymbol{K_f}(\boldsymbol{\Theta}) \tag{11}$$

where $\boldsymbol{\Theta} = \{\theta_1, \theta_2, \theta_3\}^T$.

Table 1
Pose and motion estimation algorithm.

```
1.   Given an initial guess of target pose: $\boldsymbol{X}_p = \boldsymbol{X}_{p0}$;
2.   Input measurement;
3.   Start photogrammetry loop
4.   {
5.   evaluate pin-hole camera model Jacobian $\boldsymbol{J}_c$ and image error vector $\boldsymbol{\delta}$;
6.   calculate pseudo-inverse of Jacobian: $\boldsymbol{J}_c{}^+ = \left(\boldsymbol{J}_c{}^T\boldsymbol{J}_c\right)^{-1}\boldsymbol{J}_c{}^T$;
7.   calculate previous-guess-correction: $\boldsymbol{dX}_p = \boldsymbol{J}_c{}^+\boldsymbol{\delta}$;
8.   if ($\|\boldsymbol{dX}_p\| <$ tolerance)
9.   {break;}
10.  else
11.  {$\boldsymbol{X}_p = \boldsymbol{X}_p + \boldsymbol{dX}_P$;}
12.  }
13.  Return $\boldsymbol{X}_p$;
14.  Augment to state variable: $\boldsymbol{X}_0 = \left(\boldsymbol{X}_p, \dot{\boldsymbol{X}}_p\right)$;
15.  Initialize EKF: $\boldsymbol{X}_0$ (initial state variable),
     $\boldsymbol{P}_0$ (state variable covariance matrix),
     $\boldsymbol{Q}$ (process noise covariance matrix),
     $\boldsymbol{R}$ (measurement noise covariance matrix);
16.  Start EKF loop
17.  {
18.  estimate next state variable and covariance matrix:
     $\boldsymbol{X}_{k|k-1} = \boldsymbol{AX}_{k-1|k-1}$
     $\boldsymbol{P}_{k|k-1} = \boldsymbol{AP}_{k-1|k-1}\boldsymbol{A}^T + \boldsymbol{BQB}^T$
19.  evaluate Jacobian of measurement model: $\boldsymbol{H}_k = \partial\boldsymbol{h}(\boldsymbol{X})/\partial\boldsymbol{X}|_{\boldsymbol{X} = \boldsymbol{X}_{k|k-1}}$
20.  calculate Kalman gain:
     $\boldsymbol{K}_g = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T\left(\boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}\right)^{-1}$
21.  update state variable and covariance matrix:
     $\boldsymbol{X}_{k|k} = \boldsymbol{X}_{k|k-1} + \boldsymbol{K}_g\left[\boldsymbol{Z}_k - \boldsymbol{h}\left(\boldsymbol{X}_{k|k-1}\right)\right]$
     $\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_g\boldsymbol{H}_k\boldsymbol{P}_{k|k-1}$
22.  record and output: $\boldsymbol{X}_{k|k}$, $\boldsymbol{P}_{k|k}$;
23.  }
```

Thus, the velocity of the end-effector can be derived by taking the time derivative of Eq. (11)

$$\dot{\boldsymbol{X}}_{\boldsymbol{g}} = \boldsymbol{J}_{\boldsymbol{r}} \dot{\boldsymbol{\Theta}} \tag{12}$$

where $\boldsymbol{J}_{\boldsymbol{r}}$ is the Jacobian matrix of the robotic manipulator formed by joint angles.

Eqs. (11) and (12) form the forward kinematics of the robotic manipulator. The actuators are step motors at joints, which drive the end-effector to the desired position by achieving the desired joint angles. Thus, the transformation of position to joint angles, namely inverse kinematics, should be derived in advance. It is well known that the robotic inverse kinematics may have multiple-solutions due to the periodicity of trigonometric functions. In order to obtain a unique solution of joint angles based on Eq. (11), we introduced additional constraints, such as mechanical configuration and motion limits as per [16] in addition to the following inverse kinematics

$$\boldsymbol{\Theta} = \tilde{\boldsymbol{K}}_f^{-1} \boldsymbol{X}_{\boldsymbol{g}}, \quad \dot{\boldsymbol{\Theta}} = \boldsymbol{J}_{\boldsymbol{r}}^{-1} \dot{\boldsymbol{X}}_{\boldsymbol{g}} \tag{13}$$

2.4. *Desired state of the end-effector*

The desired state of the end-effector is defined as the pose and motion of a target in the dynamic PBVS. Since the pose and motion of a target are estimated relative to the camera frame, they must be transformed to the global frame first, such that

$$\boldsymbol{X}_{\boldsymbol{g}} = \boldsymbol{T}_{\boldsymbol{cg}} \boldsymbol{X}_{\boldsymbol{c}} \tag{14}$$

$$\dot{\boldsymbol{X}}_{\boldsymbol{g}} = \dot{\boldsymbol{T}}_{\boldsymbol{cg}} \boldsymbol{X}_{\boldsymbol{c}} + \boldsymbol{T}_{\boldsymbol{cg}} \dot{\boldsymbol{X}}_{\boldsymbol{c}} \tag{15}$$

where $\boldsymbol{T}_{\boldsymbol{cg}}$ is the transformation matrix from the camera frame to the global frame.

Denote the position and velocity vectors of a target as

$$\boldsymbol{X}_{\boldsymbol{c}}^{\boldsymbol{t}} = \{x_{To}, y_{To}, z_{To}\}^T, \quad \dot{\boldsymbol{X}}_{\boldsymbol{c}}^{\boldsymbol{t}} = \{\dot{x}_{To}, \dot{y}_{To}, \dot{z}_{To}\}^T \tag{16}$$

Then, the desired position and velocity of the end-effector in the global frame are determined by substituting Eqs. (14) and (15) into Eq. (16)

$$\boldsymbol{X}_{\boldsymbol{g}}^{\boldsymbol{d}} = \boldsymbol{T}_{\boldsymbol{cg}} \boldsymbol{X}_{\boldsymbol{c}}^{\boldsymbol{t}}, \quad \dot{\boldsymbol{X}}_{\boldsymbol{g}}^{\boldsymbol{d}} = \dot{\boldsymbol{T}}_{\boldsymbol{cg}} \boldsymbol{X}_{\boldsymbol{c}}^{\boldsymbol{t}} + \boldsymbol{T}_{\boldsymbol{cg}} \dot{\boldsymbol{X}}_{\boldsymbol{c}}^{\boldsymbol{t}} \tag{17}$$

Accordingly, the desired joint angles and angular velocities of the first three joints of the robotic manipulator are determined by substituting Eq. (17) into Eq. (13), such as

$$\left\{ \theta_1^d \quad \theta_2^d \quad \theta_3^d \right\}^T = \tilde{\boldsymbol{K}}_f^{-1} \boldsymbol{X}_{\boldsymbol{g}}^{\boldsymbol{d}}, \quad \left\{ \dot{\theta}_1^d \quad \dot{\theta}_2^d \quad \dot{\theta}_3^d \right\}^T = \boldsymbol{J}_{\boldsymbol{r}}^{-1} \dot{\boldsymbol{X}}_{\boldsymbol{g}}^{\boldsymbol{d}} \tag{18}$$

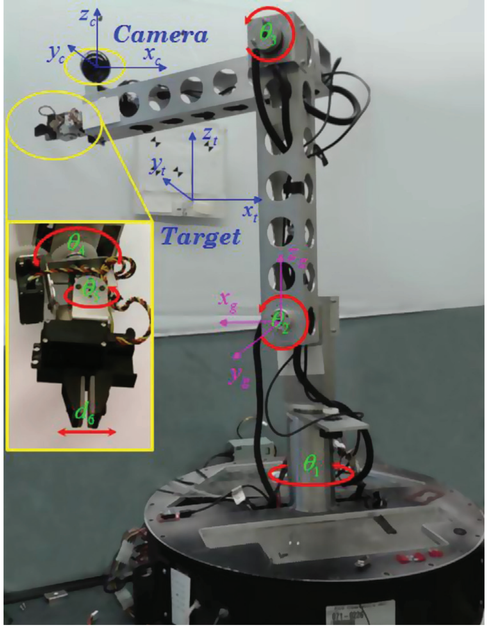


Fig. 2. The custom-built 6DOF robotic manipulator.

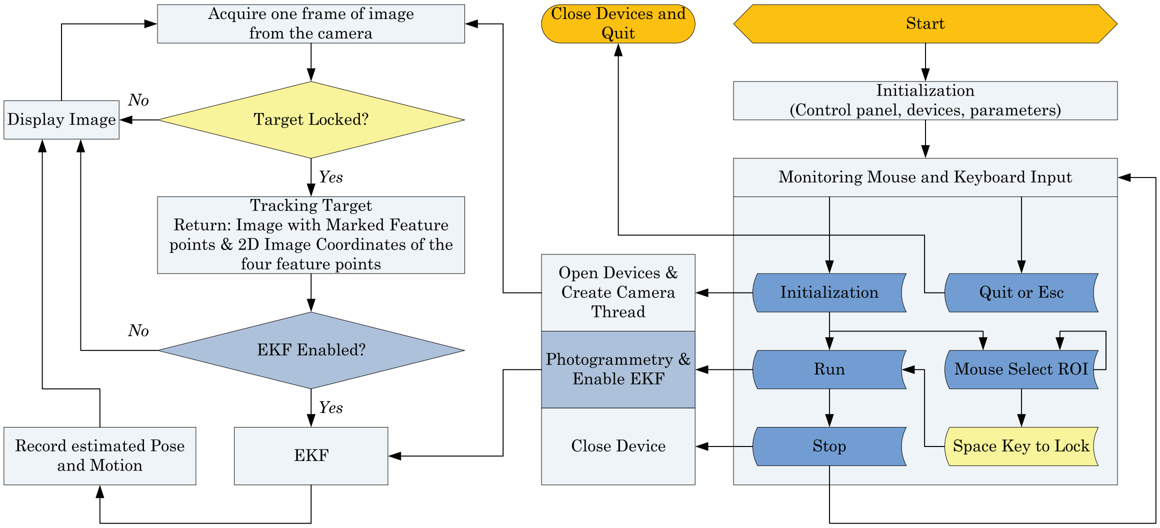

Acquire one frame of image from the camera
Target Locked?
No
Display Image
Yes
Tracking Target Return: Image with Marked Feature points & 2D Image Coordinates of the four feature points
No
EKF Enabled?
Yes
Record estimated Pose and Motion
EKF
Close Devices and Quit
Start
Initialization (Control panel, devices, parameters)
Monitoring Mouse and Keyboard Input
Open Devices & Create Camera Thread
Photogrammetry & Enable EKF
Close Device
Initialization
Quit or Esc
Run
Mouse Select ROI
Stop
Space Key to Lock

Fig. 3. Flowchart of the pose and motion estimation algorithm.

The desired joint angles and angular velocities of the wrist roll and wrist yaw are defined as

$$\left\{ \theta_4^d \quad \theta_5^d \quad \dot{\theta}_4^d \quad \dot{\theta}_5^d \right\} = \left\{ \theta_y \quad \theta_z \quad \dot{\theta}_y \quad \dot{\theta}_z \right\} \tag{19}$$

They will be determined separately by the capture strategy defined below.

2.5. Capture strategy

Once the target's pose and position are estimated by the EKF, the task of autonomous tracking, approaching and capturing of the target will be performed by an inner control loop of the robot. For the dynamic capture, the robotic manipulator should track and approach the interception point of the moving target instead of its current position [16], which is approximately estimated as

$$\boldsymbol{\Theta}^{\boldsymbol{I}} = \boldsymbol{\Theta}^{\boldsymbol{d}} + \dot{\boldsymbol{\Theta}}^{\boldsymbol{d}}(\Delta T + t) \tag{20}$$

where t is the sample time of the vision system and $\Delta T = \left(\|\boldsymbol{X}_{\boldsymbol{g}}^{\boldsymbol{d}} - \boldsymbol{X}_{\boldsymbol{g}}\| / \|\dot{\boldsymbol{X}}_{\boldsymbol{g}}\| \right) = \left(\|\boldsymbol{T}_{\boldsymbol{cg}} \boldsymbol{X}_{\boldsymbol{c}}^{\boldsymbol{t}} - \boldsymbol{K}_{\boldsymbol{f}}\| / \|\boldsymbol{J}_{\boldsymbol{r}} \dot{\boldsymbol{\Theta}}\| \right)$ is the estimated time for the end-effector to approach the desired position from its current position at its maximum allowed velocity. It should be noted that the purpose of time ΔT is to provide a lead-time for the controller to estimate the interception point and the accuracy of its estimate is not critical. This is because ΔT as well as the error in the estimation of interception point in Eq. (20) will be reduced as the robot approaches the target.

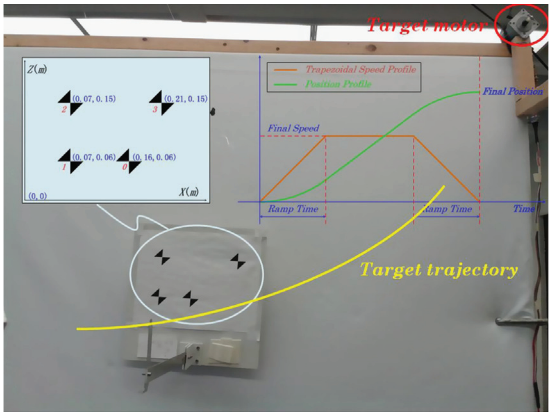


Fig. 4. Non-cooperative target system.

Thus, the control for joint actuators with respect to their current positions is obtained as

$$\Delta \boldsymbol{\Theta}^C = \boldsymbol{\Theta}^d + \dot{\boldsymbol{\Theta}}^d(\Delta T + t) - \boldsymbol{\Theta} \tag{21}$$

Once the target is within the capture region of the end-effector, the control input for the wrist roll and yaw angles is the errors between the orientations of the griper and the target, which can be simply defined as

$$\Delta\theta_{4,5}^C = \Delta\theta_{4,5}^d + \dot{\theta}_{4,5}^d t \tag{22}$$

The control strategy to activate a capture is given in [16].

3. Experimental setup

3.1. Robotic manipulator system

The proposed pose and motion estimation algorithm for the non-cooperative target and the PBVS autonomous capture by the robotic manipulator are validated

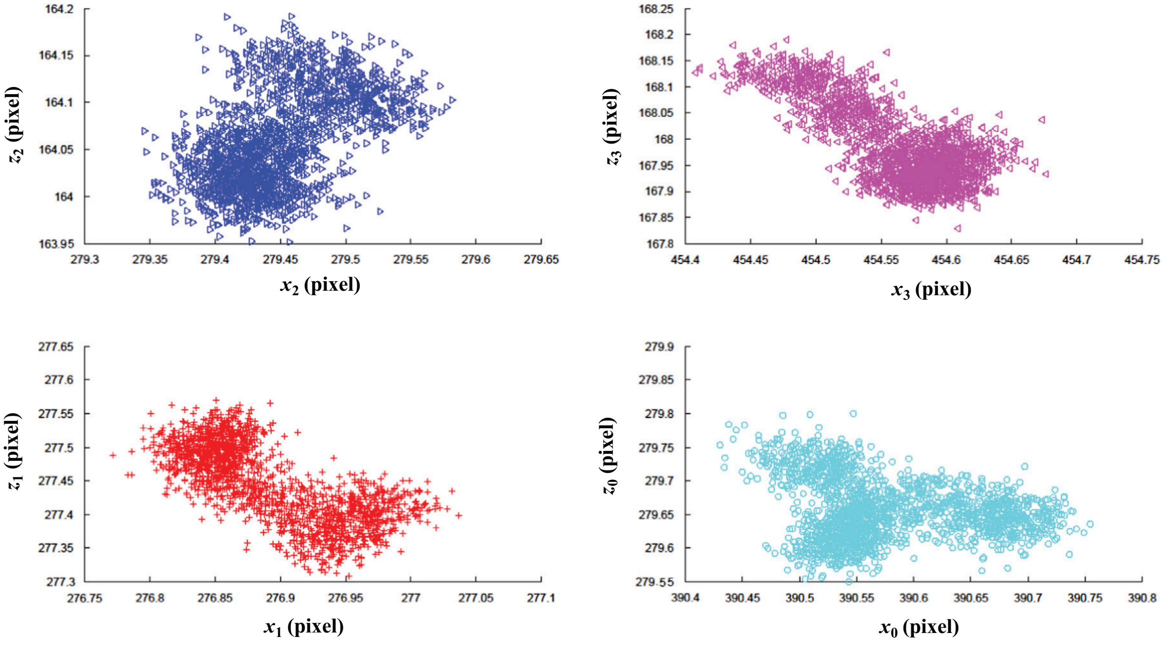


Fig. 5. The centroid coordinates of feature points of a stationary target on image plane.

experimentally. A robotic manipulator with five revolute joints, one prismatic joint and an eye-in-hand camera mounted closely to the end effector is custom built to conduct the autonomous PBVS capture operation, as shown in Fig. 2. The camera used in the experiment was a Logitech webcam with 640 × 480 resolution and 2 mm focal length. The implementation of the pose and motion estimation and PBVS control has been done by the Microsoft Fundamental Class (MFC) in order to obtain a friendly human machine interface. The flowchart of the implementation is shown in Fig. 3. Once the camera is activated, a separate thread is created simultaneously for the camera to acquire image data and display them on the computer screen. After the target is locked, the imaging processing subroutine is activated to track the motion of feature points and output their pixel coordinates in sequence.

3.2. Non-cooperative target system

A target with a known pattern of four low-optical-noise features is designed to facilitate a simple optical locking and tracking. The target system as well as configuration of feature points is shown in Fig. 4. The target is driven by a single stepper motor with programmable speed profile in the tests. By programming the speed profile of the motor and isolating the target system from the robot system, several testing scenarios for the non-cooperative target can be generated.

The task of imaging processing is to track the center point of four optical features on the target. It is done by the Open-CV (Open Source Computer Vision) software. By applying the *cvGoodFeaturesToTrack* and *cvFindCornerSubPix* functions of OpenCV library, the target tracking program groups the corners of each project feature image to extract the centroid coordinates and tracks the centers of each group by *cvCalcOpticalFlowPyrLK* function. The extracted coordinates of the center of each feature of a stationary target are shown in Fig. 5. It can be seen that the image noises of each feature points are within an area of 0.5 × 0.5 pixel due to jittering. Thus, the covariance matrix of the measurement noise in EKF can be determined accordingly as, $\boldsymbol{R}=e^2\boldsymbol{I}_{8\times 8}$, where e stands for the metric dimension of half-pixel. The covariance matrix of process noise, $\boldsymbol{Q}=10^{-5}\boldsymbol{I}_{6\times 6}$, was determined by trial-and-error.

4. Experimental results and discussion

4.1. Static pose estimation by EKF

The target was stationary in the global frame in the static pose estimation. It was placed roughly 0.5 m away from the camera. The pose estimates by the EKF, as shown in Fig. 6, are compared with the results by photogrammetry. As expected, the results of photogrammetry fluctuate around the true pose of the target due to the jittering of pixel coordinates of features. The jitter distribution in the experiments was observed as ±0.5 pixel as shown in Fig. 5. It resulted in roughly 0.1 mm translational and 0.4° rotational fluctuations in pose estimation as shown in Fig. 6(a). This is because the photogrammetry does not consider the measurement noises of images and propagates the noises down to the pose estimation. In contrast, the poses estimated by the EKF are much smoother than the photogrammetry. The EKF suppressed the spread of the measurement noises by considering the effects of measurement noises. The motion estimated by the EKF is stationary compared with the one estimated based on the numerical derivation of the photogrammetry data, which is very noisy and unsuitable for robotic control, see Fig. 6(b). This experiment demonstrates the necessity of using the EKF in the PBVS of robots.

4.2. Dynamic pose and motion estimation by EKF

The dynamic pose and motion estimation was done by manually translating and rotating the target along x, y, z axes, while the target was roughly 0.5 m away from the camera. The estimated pose and motion of the moving target are shown in Fig. 7 together with the results from the photogrammetry. Fig. 7(a) shows that the positions estimated by the EKF and photogrammetry are in good agreement. However, the orientations estimated by the photogrammetry are noisier than the EKF. It is worth pointing out that the EKF requires the Jacobian matrix of measurement model, which induces linearization error. The stronger the nonlinearity, the more error may be induced. According to Eq. (3), the nonlinearity of y, θ_x, θ_z are relatively stronger than x, z, θ_y in the measurement model. Therefore, it can be seen the test results of y, θ_x, θ_z estimated by the EKF agree with the photogrammetry better than x, z, θ_y. In the latter case, the estimates by the EKF show some delay in phase when the target experienced sudden motion. Fig. 7(b) shows the motion estimated by the EKF. As a comparison, the motion obtained by a simple numerical derivation of the photogrammetry data is also plotted. It shows clearly that the EKF results are much smoother than the photogrammetry and suitable for robotic tracking control.

4.3. Robotic autonomous capture enhanced by EKF

Once the EKF algorithm is validated, the robotic autonomous capture strategy is tested using the custom-built robotic manipulator. The robotic manipulator always starts at its home position, i.e., torso 0°, shoulder 90° and elbow 0°. The speed profile of the target motor is a trapezoidal with a ramp of two seconds in the dynamic capture testing. The maximum linear velocity of the target was set to 1.2 cm/s and the direction of the velocity vector changed in the course. Fig. 8 shows the test results where the autonomous capture process was divided into three phases. Phase 1 is the target searching and locking until the target was locked by the vision system. In phase 2, the vision system tracked the motion of the target using optical flow to increase the imaging process efficiency. The robot started to track and approach the target based on the estimated information of pose and motion of the target. The desired and actual joint angles change simultaneously as the end-effector approaches the target. This is because the position of the target with respect to the eye-in-hand camera varied continuously when the pose error between the end-effector and the target diminished.

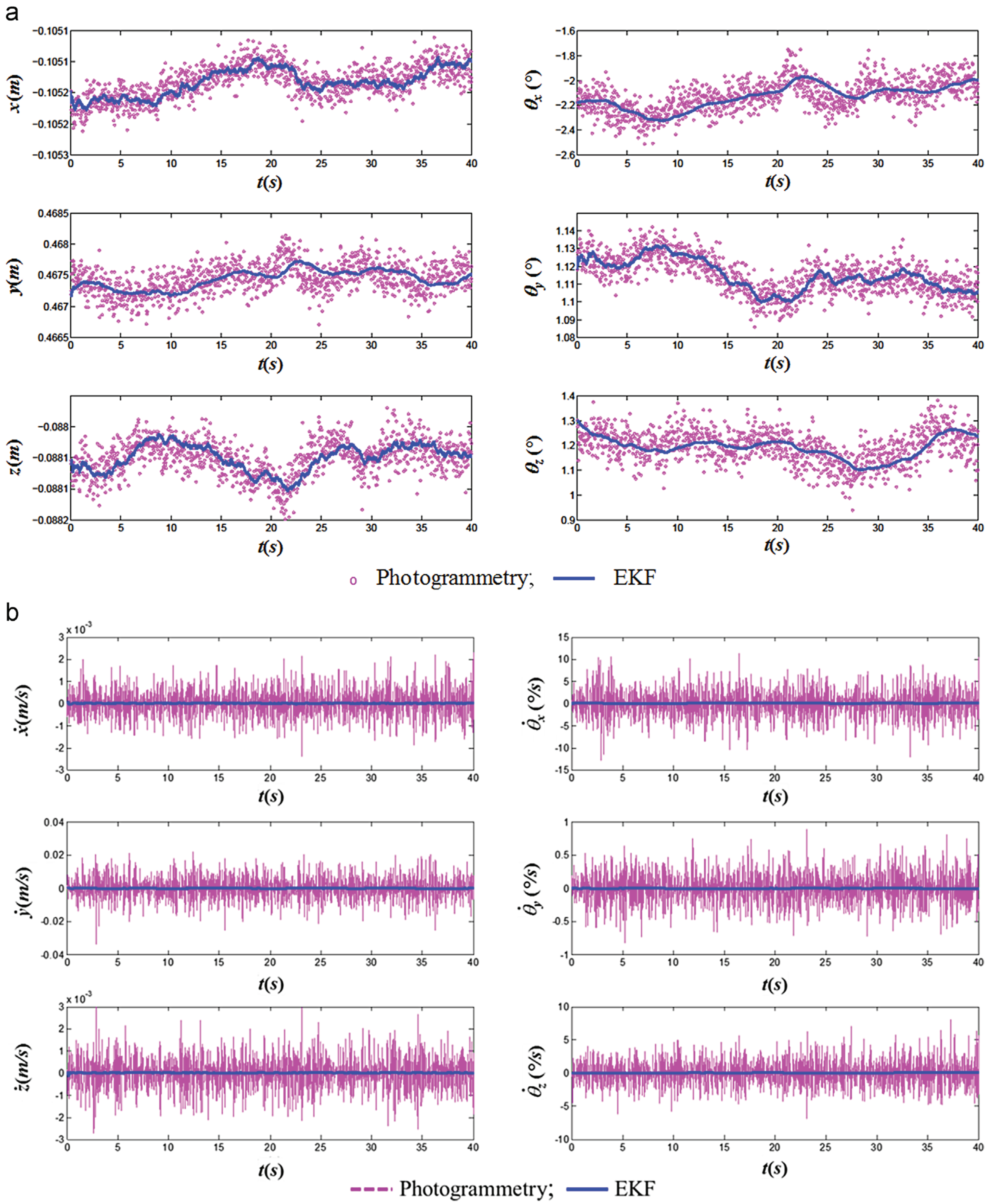


Fig. 6. Static test results: (a) pose estimation and (b) motion estimation.

However, the estimated desired joint angles were smoother than the actual joint angles due to some residual vibration in joints. Furthermore, the target motion caused the estimated desired joint angles to vary ahead of the actual joint angles as shown in Fig. 8 because the end-effector was approaching to the interception instead of the current target position. Finally in phase 3, the manipulator successfully captured the moving target.

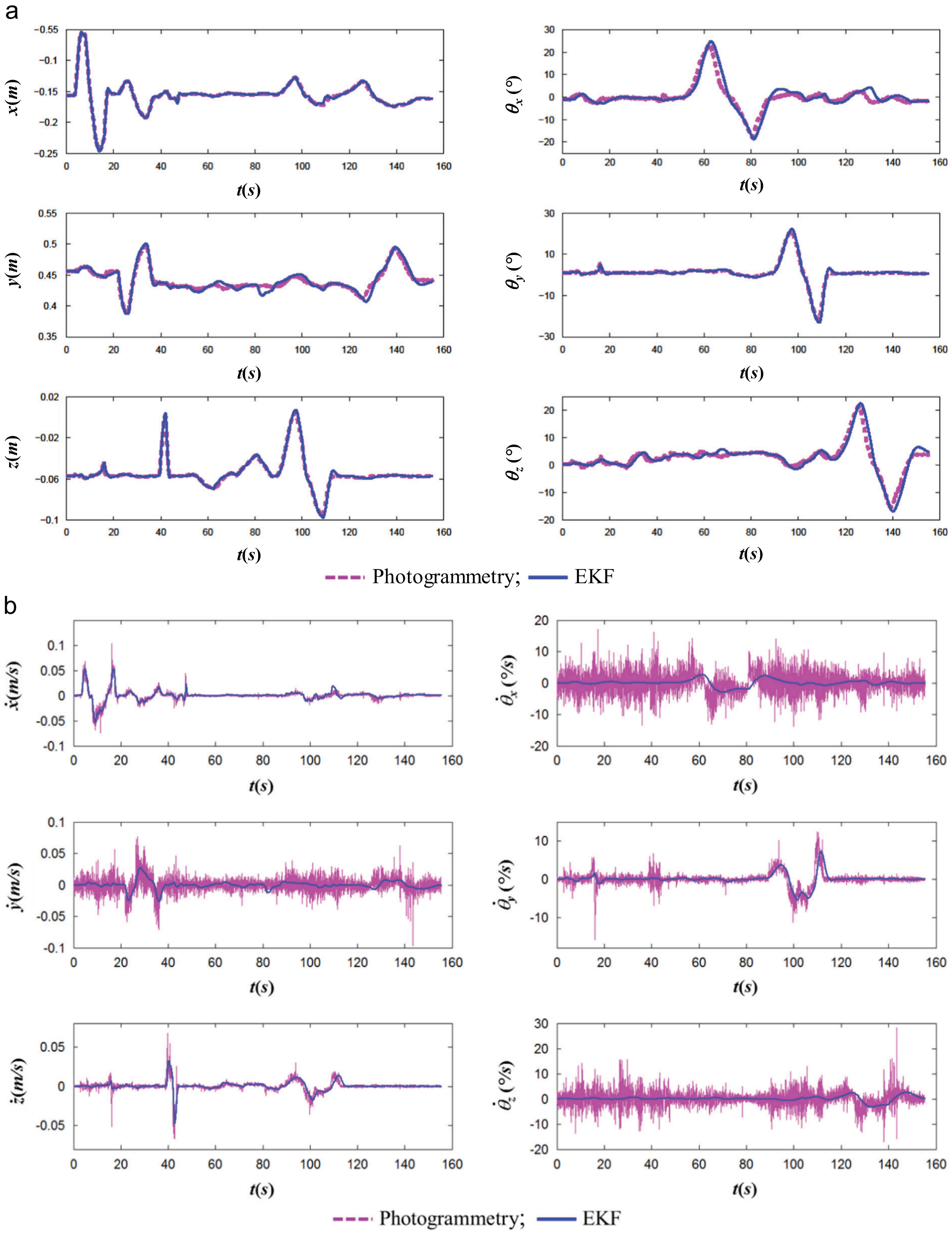


Fig. 7. Dynamic test results: (a) pose estimation and (b) motion estimation.

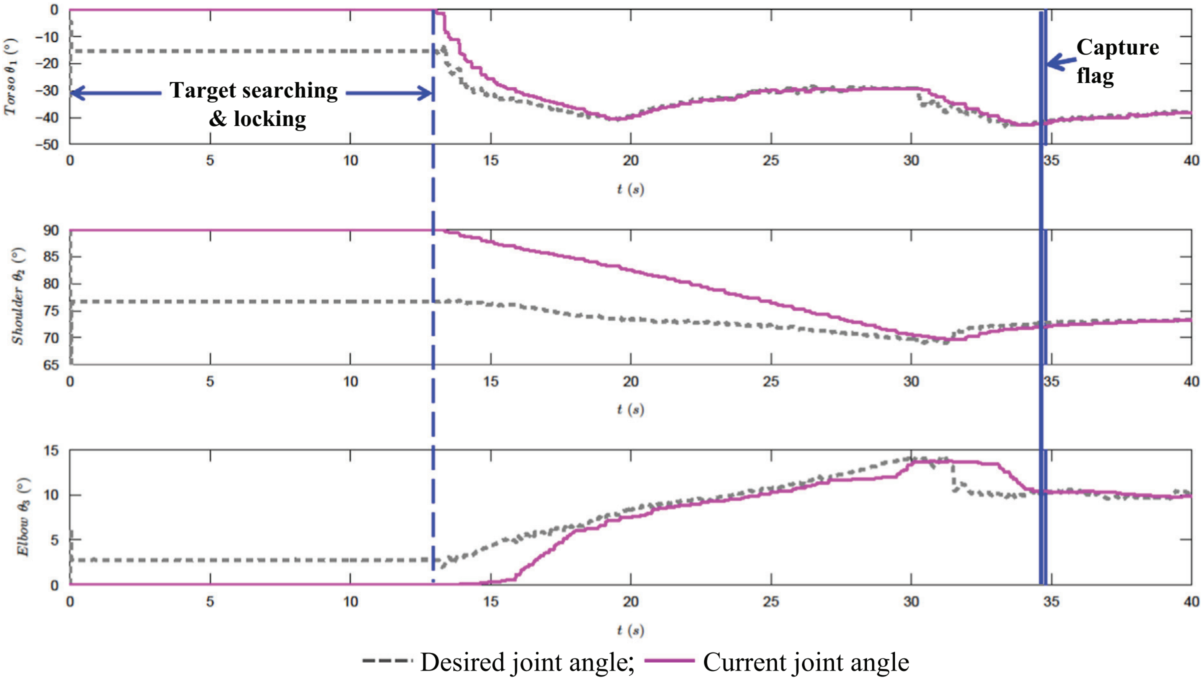


Fig. 8. Test results of autonomous capture of moving target.

5. Conclusion

This paper proposed a real-time PBVS control of autonomous robotic capture of a non-cooperative target. A vision-based pose and motion estimation algorithm of a non-cooperative target was developed by the photogrammetry and the extended Kalman filter for robotic manipulators to perform the PBVS autonomous capture. This methodology adopts the photogrammetry to initialize the EKF to improve its convergence rate when dealing with the non-cooperative target case. Optical flow algorithm is employed to track the target in order to increase the imaging processing speed for real-time pose and motion estimation of the non-cooperative target. This is beneficial for reducing the time delaying in joint actuation. A close-loop PBVS control strategy is devised to determine the desired pose and position of the end-effector at the rendezvous point once the pose and motion of the target are estimated. The proposed approach was validated experimentally on a custom-built robotic manipulator with an eye-in-hand configuration. The experimental results show that the EKF provided a smooth estimation of pose and motion of the target and enabled successful captures of a non-cooperative target. The experiments demonstrated that the effectiveness and robustness of the proposed EKF enhanced pose and motion estimation and the PBVS control strategy.

Acknowledgment

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] J. Yuh, Design and control of autonomous underwater robots: a survey, Auton. Robots 8 (2000) 7–24.

[2] S.S. Mehta, T.F. Burks, Vision-based control of robotic manipulator for citrus harvesting, Comput. Electron. Agric. 102 (2014) 146–158.

[3] S.J. Lee, S.C. Lee, H.S. Ahn, Design and control of tele-matched surgery robot, Mechatronics 24 (2014) 395–406.

[4] L. Pedersen, D. Kortenkamp, D. Wettergreen, I. Nourbakhsh, A survey of space robotics, in: Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space, 2003, pp. 19–23.

[5] K. Yoshida, Achievements in space robotics, IEEE Robot. Autom. Mag. 16 (2009) 20–28.

[6] M. Robotics, Mobile Servicing System Data Sheet, MD Robotics, Brampton, Ontario, Canada, 2002.

[7] T. Matsueda, F. Kuwao, S. Motohasi, R. Okamura, Development of Japanese experiment module remote manipulator system, in: Proceedings of JPL, Third International Symposium on Artificial Intelligence, Robotics, and Automation for Space, 1994, pp. 183–186 (SEE N 95-23672 07-63).

[8] R. Boumans, C. Heemskerk, The European robotic arm for the international space station, Robot. Auton. Syst. 23 (1998) 17–27.

[9] 〈http://www.jpl.nasa.gov/news/fact_sheets/viking.pdf〉.

[10] E. Tunstel, M. Maimone, A. Trebi-Ollennu, J. Yen, R. Petras, R. Willson, Mars exploration rover mobility and robotic arm operational performance, in: Proceedings of 2005 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2005, pp. 1807–1814.

[11] P.A.R. Billing, C.-A.R. Fleischner, Mars science laboratory robotic arm, in: Proceedings of the 14th European Space Mechanisms & Tribilogy Symposium-ESMATS, 2011.

[12] C. Covault, Curiosity's mission to Mars, Aerosp. Am. 49 (2011) 28–34.

[13] 〈http://www.ieee.ca/millennium/canadarm/canadarm_technical.html〉.

[14] 〈http://archive.darpa.mil/orbitalexpress/pdf/oe_fact_sheet_final.pdf〉.

[15] G. Hirzinger, B. Brunner, K. Landzettel, N. Sporer, J. Butterfass, M. Schedl, Space robotics—DLR's telerobotic concepts, lightweight arms and articulated hands, Auton. Robots 14 (2003) 127–145.

[16] B.P. Larouche, Z.H. Zhu, Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control, Auton. Robots 37 (2014) 157–167.

[17] G. Moustris, S. Hiridis, K. Deliparaschos, K. Konstantinidis, Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature, Int. J. Med. Robot. Comput. 7 (2011) 375–392.

[18] C. Liu, X.H. Huang, M. Wang, Target tracking for visual servoing systems based on an adaptive Kalman filter, Int. J. Adv. Robot. Syst. 9 (2012) 1–12.

[19] M. Kazemi, K. Gupta, M. Mehrandezh, Path planning for image-based control of wheeled mobile manipulators, in: Proceedings of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2012, pp. 5306–5312.

[20] S. Hutchinson, G.D. Hager, P.I. Corke, A tutorial on visual servo control, IEEE Trans. Robot. Autom. 12 (1996) 651–670.

[21] D. Kragic, H.I. Christensen, Survey on Visual Servoing for Manip-ulationComputational Vision and Active Perception Laboratory, Fiskartorpsv.

[22] B. Espiau, F. Chaumette, P. Rives, A new approach to visual servoing in robotics, IEEE Trans. Robot. Autom. 8 (1992) 313–326.

[23] J.T. Feddema, C.S.G. Lee, Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera, IEEE Trans. Syst. Man Cybern. 20 (1990) 1172–1183.

[24] M.H. Ghasemi, N. Kashiri, M. Dardel, Time-optimal trajectory planning of robot manipulators in point-to-point motion using an indirect method, Proc. Inst. Mech. Eng. C: J. Mech. Eng. Sci. 226 (2011) 473–484.

[25] Q. Zhang, S.-R. Li, X.-S. Gao, Practical smooth minimum time trajectory planning for path following robotic manipulators, in: Proceedings of the American Control Conference (ACC), IEEE, 2013, pp. 2778–2783.

[26] G. Flandin, F. Chaumette, E. Marchand, Eye-in-hand/eye-to-hand cooperation for visual servoing, in: Proceedings of IEEE International Conference on Robotics and Automation, 2000, ICRA'00, IEEE, 2000, pp. 2741–2746.

[27] G.J. Garcia, J.A. Corrales, J. Pomares, F. Torres, Survey of visual and force/tactile control of robots for physical interaction in Spain, Sensors 9 (2009) 9689–9733.

[28] F. Chaumette, S. Hutchinson, Visual servo control - Part I. Basic approaches, IEEE Robot. Autom. Mag. 13 (2006) 82–90.

[29] F. Chaumette, S. Hutchinson, Visual servo control - Part II: advanced approaches, IEEE Robot. Autom. Mag. 14 (2007) 109–118.

[30] A. De Luca, M. Ferri, G. Oriolo, P.R. Giordano, Visual servoing with exploitation of redundancy: an experimental study, in: Proceedings of IEEE International Conference on Robotics and Automation, 2008, ICRA 2008, IEEE, 2008, pp. 3231–3237.

[31] P. Gasbarri, M. Sabatini, G.B. Palmerini, Ground tests for vision based determination and control of formation flying spacecraft trajectories, Acta Astronaut. 102 (2014) 378–391.

[32] M. Sabatini, R. Monti, P. Gasbarri, G.B. Palmerini, Adaptive and robust algorithms and tests for visual-based navigation of a space robotic manipulator, Acta Astronaut. 83 (2013) 65–84.

[33] M. Sabatini, R. Monti, P. Gasbarri, G. Palmerini, Deployable space manipulator commanded by means of visual-based guidance and navigation, Acta Astronaut. 83 (2013) 27–43.

[34] H.S. Wang, Y.H. Liu, W.D. Chen, Visual tracking of robots in uncalibrated environments, Mechatronics 22 (2012) 390–397.

[35] C. Cai, E. Dean-León, N. Somani, A. Knoll, 3D image-based dynamic visual servoing with uncalibrated stereo cameras, in: Proceedings of 2013 44th International Symposium on Robotics (ISR), IEEE, 2013, pp. 1–6.

[36] M. Sabatini, P. Gasbarri, R. Monti, G.B. Palmerini, Vibration control of a flexible space manipulator during on orbit operations, Acta Astronaut. 73 (2012) 109–121.

[37] C. Martínez, I.F. Mondragón, M.A. Olivares-Méndez, P. Campoy, On-board and ground visual pose estimation techniques for UAV control, J. Intell. Robot. Syst. 61 (2011) 301–320.

[38] K. Konolige, M. Agrawal, J. Sola, Large-scale visual odometry for rough terrain M. Kanekom, Y. Nakamura, (Eds.), in: Robotics Research, Springer, 66,2011, pp. 201–212.

[39] F. Janabi-Sharifi, M. Marey, A. Kalman-Filter-Based, Method for pose estimation in visual servoing, IEEE Trans. Robot. 26 (2010) 939–947.

[40] G. Caron, A. Dame, E. Marchand, Direct model based visual tracking and pose estimation using mutual information, Image Vis. Comput. 32 (2014) 54–63.

[41] R.E. Kalman, A new approach to linear filtering and prediction problems, J. Basic Eng. 82 (1960) 35–45.

[42] S.Y. Chen, Kalman filter for robot vision: a survey, IEEE Trans. Ind. Electron. 59 (2012) 4409–4420.

Reference Paper B

Acta Astronautica 122 (2016) 209–218

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Autonomous robotic capture of non-cooperative target by adaptive extended Kalman filter based visual servo

Gangqi Dong, Zheng H. Zhu *

Department of Earth and Space Science and Engineering, York University, 4700 Keele Street, Toronto, Ontario M3J 1P3, Canada

ARTICLE INFO

Article history:
Received 1 August 2015
Received in revised form
11 January 2016
Accepted 5 February 2016
Available online 12 February 2016

Keywords:
Adaptive extend Kalman filter
Visual servo
Robotic manipulator
Autonomous capture
Non-cooperative target

ABSTRACT

This paper presents a real-time, vision-based algorithm for the pose and motion estimation of non-cooperative targets and its application in visual servo robotic manipulator to perform autonomous capture. A hybrid approach of adaptive extended Kalman filter and photogrammetry is developed for the real-time pose and motion estimation of non-cooperative targets. Based on the pose and motion estimates, the desired pose and trajectory of end-effector is defined and the corresponding desired joint angles of the robotic manipulator are derived by inverse kinematics. A close-loop visual servo control scheme is then developed for the robotic manipulator to track, approach and capture the target. Validating experiments are designed and performed on a custom-built six degrees of freedom robotic manipulator with an eye-in-hand configuration. The experimental results demonstrate the feasibility, effectiveness and robustness of the proposed adaptive extended Kalman filter enabled pose and motion estimation and visual servo strategy.

1. Introduction

Robotic manipulators have achieved great successes in industrial and space applications to perform tasks that are dangerous, complex, and even impossible to be conducted by human beings [1–4]. Specifically, the autonomous capture of non-cooperative targets by robotic manipulators is one of the research highlights in the robotic field [5–8]. For instance, the control for on-orbit space manipulators during grasping, docking and post-docking was studied in [7,8]. Regarding to non-cooperative target, one of the critical challenges arises from such a task is the accurate pose and motion estimates of the target [5,9] in order to define the end-effector's pose and trajectory [10,11]. Generally, the target is unknown to the robot and the vision system is extensively used for pose and motion estimation of the target due to its non-intrusive, non-damaging and non-contact nature, seen in [2,5,9,12–15]. The configuration of the vision sensing in a visual servo robotic system can be either eye-in-hand or eye-to-hand [16]. The eye-in-hand provides a detailed and accurate scene of the target while the eye-to-hand monitors the whole workspace with less accuracy [17]. Based on the error used in the feedback control loop, the visual servo can be cataloged as position-based, image-based and hybrid visual servo as per [18,19]. The critical issue to perform autonomous capture of non-cooperative targets by visual servo is the precise estimation of pose and motion of the target in a dynamic environment.

Considerable effort has been devoted to extract information from visual images in the past [20–23]. Different methodologies have been developed, which can be distinguished generally into four categories: analytic or geometric method, optimization-based method, offline method and filtering method. The geometric method, such as the photogrammetry, is widely adopted if the camera is

* Corresponding author. Tel.: +1 416 7362100x77729.
E-mail address: gzhu@yorku.ca (Z.H. Zhu).

http://dx.doi.org/10.1016/j.actaastro.2016.02.003
0094-5765/© 2016 IAA. Published by Elsevier Ltd. All rights reserved.

calibrated and the geometric features of target are known in advance. It extracts pose information about six degrees of freedom (DOF) of the target from two-dimensional (2D) images. However, the photogrammetry relies heavily on the accuracy of imaging processing and is prone to the errors of target feature measurements and camera calibration. Moreover, the geometric method estimates is solely based on the current measurements to make predictions for the future, which is a memoryless or Markov process. Thus, its results may be noisy if the image information is not smooth. The optimization-based method requires a sufficient large set of target images in different poses, which is not available when the target is non-cooperative. The offline method is performed afterwards, which is not applicable for real-time tracking, approaching and capturing the target. The filtering method is widely used in pose and motion estimation. Among all filters, the Kalman filter (KF) is widely used in analyzing and solving pose estimation problems based on a set of noisy observations over time. Since KF being proposed in 1960 [24], it has been adopted in almost every engineering field, especially in the robotic vision, and has been extended from the original linear system to nonlinear system with different enhancing techniques. Ref. [25] provided a detailed survey of the application of KF and its various extensions in the field of robotic vision. The survey shows evidently the extended Kalman filter (EKF) is the most widely adopted nonlinear state estimation algorithm in robot vision applications. A good performance of EKF mainly depends on the good estimation of covariance matrices of system and measurement models of the filter. The challenge arises in estimating the covariance matrix of system model in dealing with the non-cooperative target where its motion is unknown in advance and unpredictable. Furthermore, all KF algorithms require the input of initial conditions and measurements over the time. Although the initial conditions do not change the convergence property of the KF, they do affect the performance of the filter, such as the convergence speed, and its results, especially when dealing with the non-cooperative target where the initial conditions are unknown. Therefore, the focus of this study will be on the impact of pose estimation of non-cooperative targets by KF in visual servo where the relative pose and motion of a non-cooperative target with respect to the camera frame is used for real-time control of the robotic manipulator, seen with limited success in [12,14]. To achieve the objective, a new approach that combines both the adaptive extended Kalman filter and photogrammetry to obtain more robust and accurate pose and motion estimates of the non-cooperative target from noisy image data in real-time is presented. The approach has been validated experimentally using a custom-built robotic manipulator with an eye-in-hand camera mounted closely to the end-effector. The experimental results demonstrate the effectiveness and robustness of the proposed algorithm as well as the control strategy by successfully tracking, approaching and capturing a non-cooperative target autonomously in both static and dynamic scenarios.

2. Camera model

Without loss of generality, the global frame is fixed in the inertia space, the camera frame is located at the center of the image plane and the target frame is attached to the target at the rotating center. The pose of a target can be described by the target frame with respect to the camera frame, such that, $\{x_{To}, y_{To}, z_{To}, \theta_x, \theta_y, \theta_z\}^T$. Here the $\{x_{To}, y_{To}, z_{To}\}^T$ is the origin of the body-fixed target frame and $\{\theta_x, \theta_y, \theta_z\}^T$ are the Euler angles of target frame with respect to the camera frame. The singularity in the frame transformation caused by the Euler angles is assumed avoided by imposing physical limits on the joint angles of manipulator in the operation space. Then, we can easily obtain the rotational matrix from target frame to the camera frame, denoted by $\mathbf{R}_{TC}$. Assume the coordinates of a feature point on the target is known in the target frame and defined as $\{x_T, y_T, z_T\}^T$. Further define $\{x_C, y_C, z_C\}^T$ as the coordinates of the corresponding feature point in the camera frame. An augmented homogeneous relationship between the target frame and the camera frame can be established as

$$\begin{Bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{Bmatrix} = \begin{bmatrix} & & & x_{To} \\ & \mathbf{R}_{TC} & & y_{To} \\ & & & z_{To} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{Bmatrix} \tag{1}$$

Consider a pin-hole camera model, the feature point on the target is projected onto the image plane, such that

$$\begin{Bmatrix} x_m \\ z_m \end{Bmatrix} = \frac{-f}{y_C - f} \begin{Bmatrix} x_C \\ z_C \end{Bmatrix} \tag{2}$$

where f stands for the focal length of the camera, and $\{x_m, z_m\}^T$ denotes the projected image coordinates. Here, it is assumed that the y_C-axis of the camera frame is pointing from the camera towards the target.

Rewrite Eq. (2) in the following form, such that,

$$\begin{cases} x_m y_C - x_m f + x_C f = 0 \\ z_m y_C - z_m f + z_C f = 0 \end{cases} \tag{3}$$

Eq. (3) indicates that there are two independent equations for one feature point, which contains six unknowns (pose information). Theoretically, one needs only three feature points to solve for the six unknowns. To eliminate the ambiguity in pose estimation with three feature points and increase the robustness of algorithm, minimum four feature points are widely adopted in literature [26]. Consequently, there will be up to eight equations with six unknowns, which are solved by an iterative least square approach with an initial guess.

The photogrammetry is a Markov process, which means that it is based only on the current measurement and prone to the measurement noise. Moreover, the computational cost of photogrammetry may increase or converge to local minima that are not true solutions, if the initial guess is far away from the real pose. As a result, the system sampling time interval may be adjusted/reduced, which is not desirable when dealing in real-time pose estimation. Another short come of the photogrammetry is that it does not estimate the target motion directly, which is an important parameter in capturing the target by a robotic manipulator autonomously in a dynamic environment. To address these challenges, an adaptive extended Kalman filter (AEKF) is presented in the follows.

3. Adaptive extended Kalman filter

The Kalman filter is an optimal estimation algorithm for a linear system with independent white noise of normal distributions. Define the state vector of target as

$$\mathbf{X} = \left\{x_{To}, \dot{x}_{To}, y_{To}, \dot{y}_{To}, z_{To}, \dot{z}_{To}, \theta_x, \dot{\theta}_x, \theta_y, \dot{\theta}_y, \theta_z, \dot{\theta}_z\right\}^T \tag{4}$$

where the overhead dot denotes time derivative. $\{x_{To}, y_{To}, z_{To}\}^T$ is the origin of the body-fixed target frame and $\{\theta_x, \theta_y, \theta_z\}^T$ are the Euler angles of target frame with respect to the camera frame defined in Section 2.

By assuming the target motion is uniform within a sample time interval t if it is sufficiently small, then the higher order motion can be regarded as the process noise. For simplicity, we define the acceleration vector as the process noise, such that, $\boldsymbol{\omega} = \left\{\ddot{x}_{To}, \ddot{y}_{To}, \ddot{z}_{To}, \ddot{\theta}_x, \ddot{\theta}_y, \ddot{\theta}_z\right\}^T$. The system model can be defined as

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{B}\boldsymbol{\omega}_{k-1} \tag{5}$$

Notes that the subscripts k and $k-1$ indicate the current and previous sampling time. The coefficient matrix of system model $\mathbf{A}$ and the coefficient matrix of process noise $\mathbf{B}$ can be written as

$$\mathbf{A} = \mathrm{diag}\begin{bmatrix}\mathbf{a} & \mathbf{a} & \mathbf{a} & \mathbf{a} & \mathbf{a} & \mathbf{a}\end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix}1 & t \\ 0 & 1\end{bmatrix},$$
$$\mathbf{B} = \mathrm{diag}\begin{bmatrix}\mathbf{b} & \mathbf{b} & \mathbf{b} & \mathbf{b} & \mathbf{b} & \mathbf{b}\end{bmatrix}, \quad \mathbf{b} = \begin{Bmatrix}t^2/2 \\ t\end{Bmatrix}. \tag{6}$$

The measurement model is derived from the pin-hole camera model as shown in Eq. (2). Define an equation vector for one feature point, such that,

$$\mathbf{h}(\mathbf{X}) = \frac{-f}{y_C - f}\begin{Bmatrix}x_C \\ z_C\end{Bmatrix} \tag{7}$$

Then, the measurement model can be written as

$$\mathbf{Z}_k = \mathbf{h}(\mathbf{X}_k) + \boldsymbol{\mu}_k \tag{8}$$

where $\mathbf{Z}$ is the real measurement vector (image coordinates in the camera frame) of four feature points, $\mathbf{h}(\mathbf{X}_k)$ is the estimated measurement vector based on the camera model and $\boldsymbol{\mu}$ stands for the measurement noise of the camera, which may be caused by different natural conditions, such as illumination, temperature of the environment, etc. This kind of noise can be assumed as Gaussian noise and obeys the normal distribution.

While the covariance matrix of measurement noise $\mathbf{R}$ can be determined in advance based on the residual errors measured by the photogrammetry algorithm experimentally, challenge arises in determining the covariance matrix of process noise $\mathbf{Q}$ for the non-cooperative target. This is because the motion of the non-cooperative target is unknown in advance. The situation is further complicated when the eye-in-hand camera is used in the vision system, where the motion of the camera is related to the motion of the robotic manipulator and thus unknown in advance neither.

Adaptive extended Kalman filter (AEKF) addresses the challenge by updating the covariance matrices at each time step [22,27,28]. Although the process and measurement noises are usually assumed to be independent, it is hard to distinguish them in the process covariance matrix from the measurement covariance matrix [29]. Thus, the AEKF that attempts to correct both covariance matrices may not be robust [30]. As mentioned before, the statistics of measurement noise can be determined in advance, therefore, the adaptivity in AEKF is only applied to the covariance matrix of process noise in this study to compensate the errors while maintaining the robustness of the estimation algorithm.

Table 1
Outline of pose and motion estimation algorithm.

1.	Estimate the initial target pose $\mathbf{X}_p$;
2.	Augment to state variable $\mathbf{X}_0 = \left\{\mathbf{X}_p, \dot{\mathbf{X}}_p\right\}$;
3.	Initialize AEKF $\mathbf{X}_0, \mathbf{P}_0, \mathbf{q}_0, \mathbf{Q}_0, \mathbf{R}$;
4.	while (AEKF enabled)
5.	{
6.	estimate next state variable and covariance matrix $\mathbf{X}_{k\|k-1} = \mathbf{A}\mathbf{X}_{k-1\|k-1} + \mathbf{B}\mathbf{q}_{k-1}\mathbf{P}_{k\|k-1} = \mathbf{A}\mathbf{P}_{k-1\|k-1}\mathbf{A}^T + \mathbf{B}\mathbf{Q}_{k-1}\mathbf{B}^T$
7.	evaluate Jacobian of measurement model $\mathbf{H}_k = \partial\mathbf{h}(\mathbf{X})/\partial\mathbf{X}\|_{\mathbf{X} = \mathbf{X}_{k\|k-1}}$
8.	calculate Kalman gain $\mathbf{K}_g = \mathbf{P}_{k\|k-1}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k\|k-1}\mathbf{H}_k^T + \mathbf{R}\right)^{-1}$
9.	update state variable and covariance matrix $\mathbf{X}_{k\|k} = \mathbf{X}_{k\|k-1} + \mathbf{K}_g(\mathbf{Z}_k - \mathbf{h}(\mathbf{X}_{k\|k-1}))\mathbf{P}_{k\|k} = \mathbf{P}_{k\|k-1} - \mathbf{K}_g\mathbf{H}_k\mathbf{P}_{k\|k-1}$
10.	output $\mathbf{X}_{k\|k}$
11.	update process noise $\hat{\mathbf{q}}_k = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T(\mathbf{X}_{k\|k} - \mathbf{A}\mathbf{X}_{k-1\|k-1})\mathbf{q}_k = \mathbf{q}_{k-1} + \frac{1}{N}(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N})$
12.	update covariance matrix of the process noise $\boldsymbol{\Lambda}_k = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\left(\mathbf{A}\mathbf{P}_{k-1\|k-1}\mathbf{A}^T - \mathbf{P}_{k\|k}\right)\mathbf{B}\left(\mathbf{B}^T\mathbf{B}\right)^{-1}$ $\mathbf{Q}_k = \mathbf{Q}_{k-1} + \frac{1}{N-1}\begin{pmatrix}(\hat{\mathbf{q}}_k - \mathbf{q}_k)(\hat{\mathbf{q}}_k - \mathbf{q}_k)^T - (\hat{\mathbf{q}}_{k-N} - \mathbf{q}_k)(\hat{\mathbf{q}}_{k-N} - \mathbf{q}_k)^T \\ +\frac{1}{N}(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N})(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N})^T + \frac{N-1}{N}(\boldsymbol{\Lambda}_{k-N} - \boldsymbol{\Lambda}_k)\end{pmatrix}$
13.	}

Define an intuitive approximation of the process noise at time j as

$$\hat{\mathbf{q}}_j = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\left(\mathbf{X}_{j|j} - \mathbf{A}\mathbf{X}_{j-1|j-1}\right) \tag{9}$$

Assume the process noises are independent and identically distributed over N sample time intervals. Then, an unbiased estimate of process noise at time k can be evaluated as [27]

$$\mathbf{q}_k = \frac{1}{N}\sum_{j=k-N+1}^{k}\hat{\mathbf{q}}_j \tag{10}$$

Meanwhile, an unbiased estimate of the covariance matrix of process noise is given by

$$\mathbf{Q}_k = \frac{1}{N-1}\sum_{j=k-N+1}^{k}\left(\left(\hat{\mathbf{q}}_j - \mathbf{q}_k\right)\left(\hat{\mathbf{q}}_j - \mathbf{q}_k\right)^T - \frac{N-1}{N}\left(\boldsymbol{\Lambda}_j\mathbf{P}_{k-1|k-1}\boldsymbol{\Lambda}_j^T - \mathbf{P}_{k|k}\right)\right) \tag{11}$$

$$\boldsymbol{\Lambda}_j = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\left(\mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T - \mathbf{P}_{k|k}\right)\mathbf{B}\left(\mathbf{B}^T\mathbf{B}\right)^{-1} \tag{12}$$

Eqs. (10)–(12) can be further simplified to reduce computational effort, such that

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \frac{1}{N}\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N}\right) \tag{13}$$

$$\mathbf{Q}_k = \mathbf{Q}_{k-1} + \frac{1}{N-1}\Big(\left(\hat{\mathbf{q}}_k - \mathbf{q}_k\right)\left(\hat{\mathbf{q}}_k - \mathbf{q}_k\right)^T - \left(\hat{\mathbf{q}}_{k-N} - \mathbf{q}_k\right)\left(\hat{\mathbf{q}}_{k-N} - \mathbf{q}_k\right)^T + \frac{1}{N}\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N}\right)\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N}\right)^T + \frac{N-1}{N}\left(\boldsymbol{\Lambda}_{k-N} - \boldsymbol{\Lambda}_k\right)\Big) \tag{14}$$

Thus, an unbiased estimate of the process noise distribution at time k is obtained as a normal distribution with the mean $\mathbf{q}_k$ and covariance matrix $\mathbf{Q}_k$. The input to AEKF is the initial conditions and the measurements observed over the time. Since initial pose and motion of the non-cooperative target are generally unknown, an inappropriate initial guess may lead to poor convergence of the AEKF. In order to improve the performance and accelerate the convergence of the AEKF, we initialize the initial states by the photogrammetry. Therefore, a recursive pose and motion estimation algorithm of AEKF can be designed as shown in Table 1, where $\mathbf{H}$ is the Jacobian matrix formed by the first order partial differential of the measurement model with respect to system state variable, $\mathbf{K}_g$ is the Kalman gain at time step k, $\mathbf{P}$ is the covariance matrix of system state variable, $\mathbf{Q}$ and $\mathbf{R}$ are the covariance matrices of process and measurement noise.

4. Visual servo control strategy

Once the pose and motion estimates of target are obtained, they are fed into the controller of robotic manipulator to capture the target autonomously. The robotic manipulator in the study is a custom designed robotic manipulator with an eye-in-hand configuration, as shown in Fig. 1. It consists of three links and one end-effector with five revolute joints, one prismatic joint and an eye-in-hand camera mounted closely to the end-effector. The first three revolute joints, namely torso (θ_1), shoulder (θ_2) and elbow (θ_3), control the position of the end-effector while the last two revolute joints and one prismatic joint, namely wrist roll (θ_4), wrist yaw (θ_5) and gripper, provide dexterous orientation and griping function for capture, see Fig. 1. Thus, the control of the position and orientation of the end-effector can be separated in the controller design for simplicity. Furthermore, the global frame is fixed in space and located at the center of the second joint, the camera frame is at the center of the image plane and the target frame is at the rotating center of the target, as illustrated in Fig. 1.

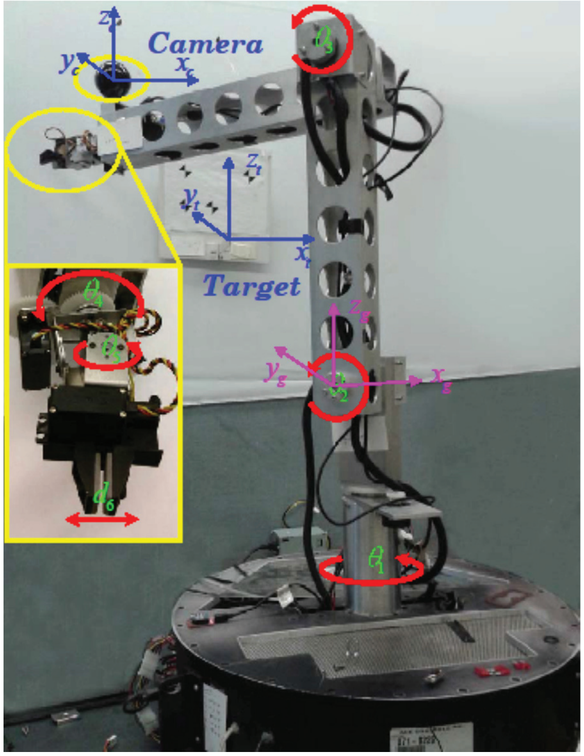


Fig. 1. Custom-built robotic manipulator with eye-in-hand configuration.

Define $\boldsymbol{\theta}_T = \{\theta_1, \theta_2, \theta_3\}^T$ as the first three revolute joint angles that control the translational motion of the end-effector, and $\boldsymbol{\theta}_R = \{\theta_4, \theta_5\}^T$ as the last two revolute joint angles that control the orientation of the end-effector. Therefore, the kinematics of robotic manipulator, which refers to the forward relationship between $\boldsymbol{\theta}_T$ and position of the end-effector regarding to the global frame $\mathbf{X}_e^G$, can be described as

$$\mathbf{X}_e^G = \mathbf{K}_f\left(\boldsymbol{\theta}_T\right) \tag{15}$$

The velocity of the end-effector is derived by taking the time derivative of Eq. (15), such that

$$\dot{\mathbf{X}}_e^G = \mathbf{J}\dot{\boldsymbol{\theta}}_T \tag{16}$$

where $\mathbf{J}$ is the Jacobian matrix of the robotic manipulator.

The robotic manipulator is controlled by actuators at rotational joints to achieve the desired joint angles, based on the desired end-effector position. Thus, the transformation from the desired position to the desired joint angles, namely inverse kinematics, should be derived as

$$\boldsymbol{\theta}_T = \tilde{\mathbf{K}}_f^{-1}\left(\mathbf{X}_e^G\right), \dot{\boldsymbol{\theta}}_T = \mathbf{J}^{-1}\dot{\mathbf{X}}_e^G \tag{17}$$

Notes that the inverse of Eq. (15) may have multiple solutions or even kinematic singularity due to the system redundancy and the periodicity of trigonometric functions. In order to address this challenge, physical limits or other extra constraints may have to be considered. In Eq. (17) the term $\tilde{\mathbf{K}}_f^{-1}\left(\mathbf{X}_e^G\right)$ stands for the inverse of Eq. (15) with the consideration of physical constraints to avoid the multiple solutions problem and the kinematic singularity.

The pose and motion of the target obtained from Section 3 is relative to the camera frame and should be transformed into the global frame to facilitate the capture of the target by the end-effector. Because of the eye-in-hand configuration of the camera, the transformation matrix from the camera frame to the global frame can be easily obtained based on the Denavit-Hartenberg convention. Thus, the coordinate transformation of the end-effector from the camera frame to the global frame can be expressed as,

$$\mathbf{X}_e^G = \mathbf{T}_{CG}\mathbf{X}_e^C \tag{18}$$

Where $\mathbf{T}_{CG}$ denotes the transformation matrix from the camera frame to the global frame. The corresponding transformation for velocity can be derived by taking the time derivative of Eq. (18), such that,

$$\dot{\mathbf{X}}_e^G = \dot{\mathbf{T}}_{CG}\mathbf{X}_e^C + \mathbf{T}_{CG}\dot{\mathbf{X}}_e^C \tag{19}$$

According to Eqs. (18) and (19), the desired translational position and velocity of the end-effector are obtained based on the pose and motion estimates of the target, such that

$$\mathbf{X}_{ed}^G = \mathbf{T}_{CG}\mathbf{X}_t^C, \quad \dot{\mathbf{X}}_{ed}^G = \dot{\mathbf{T}}_{CG}\mathbf{X}_t^C + \mathbf{T}_{CG}\dot{\mathbf{X}}_t^C \tag{20}$$

Notes that $\mathbf{X}_t^C = \{x_{To}, y_{To}, z_{To}\}^T$ and $\dot{\mathbf{X}}_t^C = \{\dot{x}_{To}, \dot{y}_{To}, \dot{z}_{To}\}^T$. Substituting Eq. (20) into Eq. (17) leads to the desired joint angles and angular velocities of the first three joints, such as

$$\left\{ \boldsymbol{\theta}_T^d = \tilde{\mathbf{K}}_f^{-1}\left(\mathbf{T}_{CG}\mathbf{X}_t^C\right)\dot{\boldsymbol{\theta}}_T^d = \mathbf{J}^{-1}\left(\dot{\mathbf{T}}_{CG}\mathbf{X}_t^C + \mathbf{T}_{CG}\dot{\mathbf{X}}_t^C\right) \right. \tag{21}$$

The desired joint angles and angular velocities of the wrist roll and yaw are obtained as

$$\boldsymbol{\theta}_R^d = \{\theta_y, \theta_z\}^T, \quad \dot{\boldsymbol{\theta}}_R^d = \left\{\dot{\theta}_y, \dot{\theta}_z\right\}^T \tag{22}$$

The control input of the first three joints is the angular position errors with respect to the desired angles. In order to capture a dynamic target, the robotic manipulator should track the target trajectory and approach the interception point of the target based on the estimated target motion. The interception time is estimated approximately based on the position tracking error of the end-effector at current velocity, such that,

$$\Delta T = \frac{\|\mathbf{X}_{ed}^G - \mathbf{X}_e^G\|}{\|\dot{\mathbf{X}}_e^G\|} = \frac{\|\mathbf{T}_{CG}\mathbf{X}_t^C - \mathbf{K}_f(\boldsymbol{\theta}_T)\|}{\|\mathbf{J}\dot{\boldsymbol{\theta}}_T\|} \tag{23}$$

As the robot approaches the target, the interception time will approach to zero. Then, the desired joint angles of the first three joints at the interception point can be estimated as

$$\tilde{\boldsymbol{\theta}}_T^d = \boldsymbol{\theta}_T^d + (\Delta T + t)\dot{\boldsymbol{\theta}}_T^d \tag{24}$$

where t denotes the sample time, $\boldsymbol{\theta}_T^d$ and $\dot{\boldsymbol{\theta}}_T^d$ are obtained by Eq. (21) as aforementioned.

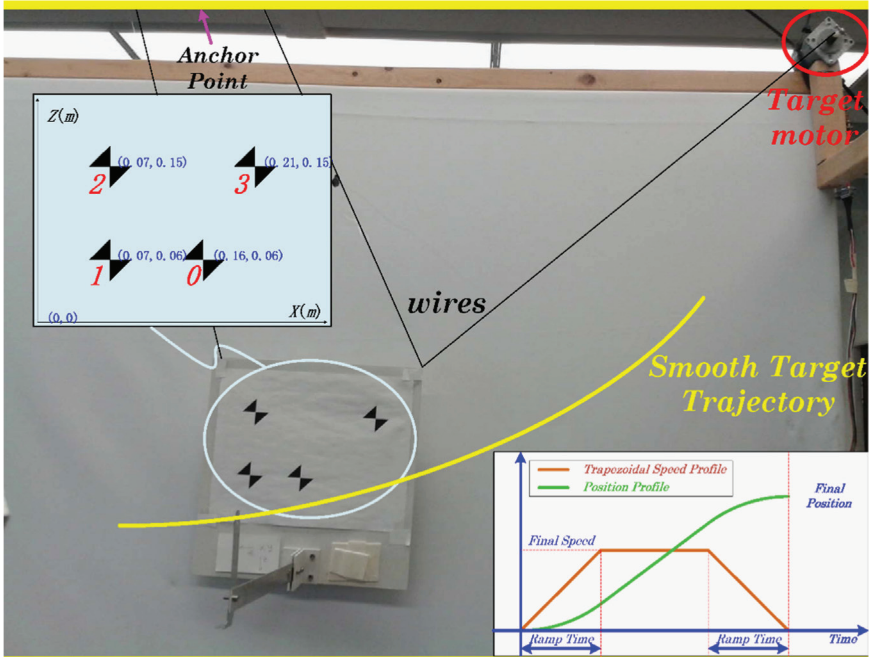


Fig. 2. Experimental setup of non-cooperative target system.

The control input of first three joints, which is the relative angular positions, is finally obtained as

$$\boldsymbol{\theta}_T^c = \tilde{\boldsymbol{\theta}}_T^d - \boldsymbol{\theta}_T \tag{25}$$

The control input of wrist roll and yaw are absolute positions, which can be simply defined as

$$\boldsymbol{\theta}_R^c = \boldsymbol{\theta}_R^d + t\dot{\boldsymbol{\theta}}_R^d \tag{26}$$

Once the grasp distance tolerance between the gripper and the grasp point of the target is achieved, the robotic manipulator is activated to make a capture [5].

5. Experimental validation

The proposed pose and motion estimation algorithm for the non-cooperative target and the autonomous capture strategy of the robotic manipulator are validated experimentally using a custom-built robotic manipulator as shown in Fig. 1. In order to obtain a friendly human-machine interface, the testing software for the pose and motion estimation was developed based on the Microsoft Fundamental Class (MFC). Once the camera is activated, a separate thread is created simultaneously for the camera to acquire and display image on a computer by the Open Source Computer Vision (OpenCV) [31]. After the target is locked, the imaging processing subsystem is activated and output pixel coordinates of each feature points in sequence. If the filtering algorithm is also enabled at this stage, the pixel coordinates will be transformed to metric ones and regarded as the measurements of filtering algorithm.

As shown in Fig. 2, the non-cooperative target was driven by a single stepper motor at the top right corner with programmable speed profiles independently for different testing scenarios. It possesses four low-noise feature points and the pattern of the feature points is carefully selected so that the error margins of the pitch and yaw estimates of the target are in the same order if estimated by the photogrammetry using a single camera. By applying the *cvGoodFeaturesToTrack* and *cvFindCornerSubPix* functions of OpenCV library, the target tracking program groups the corners of each feature point and track the centers of each group by *cvCalcOpticalFlowPyrLK* function.

The test results of image processing of a stationary target are shown in Fig. 3. It shows that the image noises of each feature point are within an area of 0.5 × 0.5 pixel^2 in the image plane. Based on the information, the covariance matrix of the measurement noise of AEKF can be estimated as, $\mathbf{R} = e^2\mathbf{I}_{8\times8}$, where $e = 3.125 \times 10^{-6}m$, stands for the metric dimension of half pixel and $\mathbf{I}_{8\times8}$ is the identity matrix. For the covariance matrix of processing noise, it cannot be determined in advance. By trial and error, the initial covariance matrix of processing noise is assumed as $\mathbf{Q} = 10^{-5}\mathbf{I}_{6\times6}$. It will be updated in the tracking process based on the proposed pose and motion estimation algorithm shown in Table 1.

To validate the proposed pose and motion estimation algorithm, experiments were performed under two scenarios: static and dynamic captures. In all cases, the poses estimated by the photogrammetry are used as the baseline and compared with the poses estimated by EKF and the proposed AEKF algorithm. The sample time step in experiments was around 0.032 s mainly dictated by the processing time of vision system to acquire and process one frame of image.

For the static scenario, the target was stationary in the global frame. The results are shown in Fig. 4. As aforementioned, the photogrammetry did not consider the measurement noise of image (±0.5 pixel), which

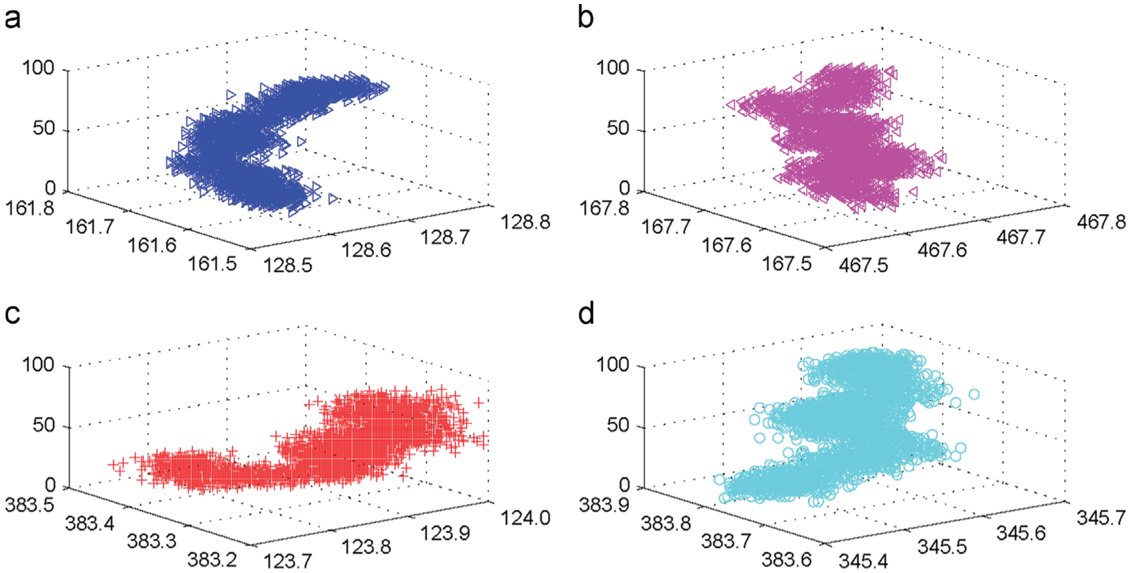


Fig. 3. Image processing error margins of four feature points of the stationary target over time. The horizontal planes represent the image plane in pixels. The vertical axes represent the time in seconds. (a–d) correspond to the feature points (2, 3, 1, 0) in Fig. 2.

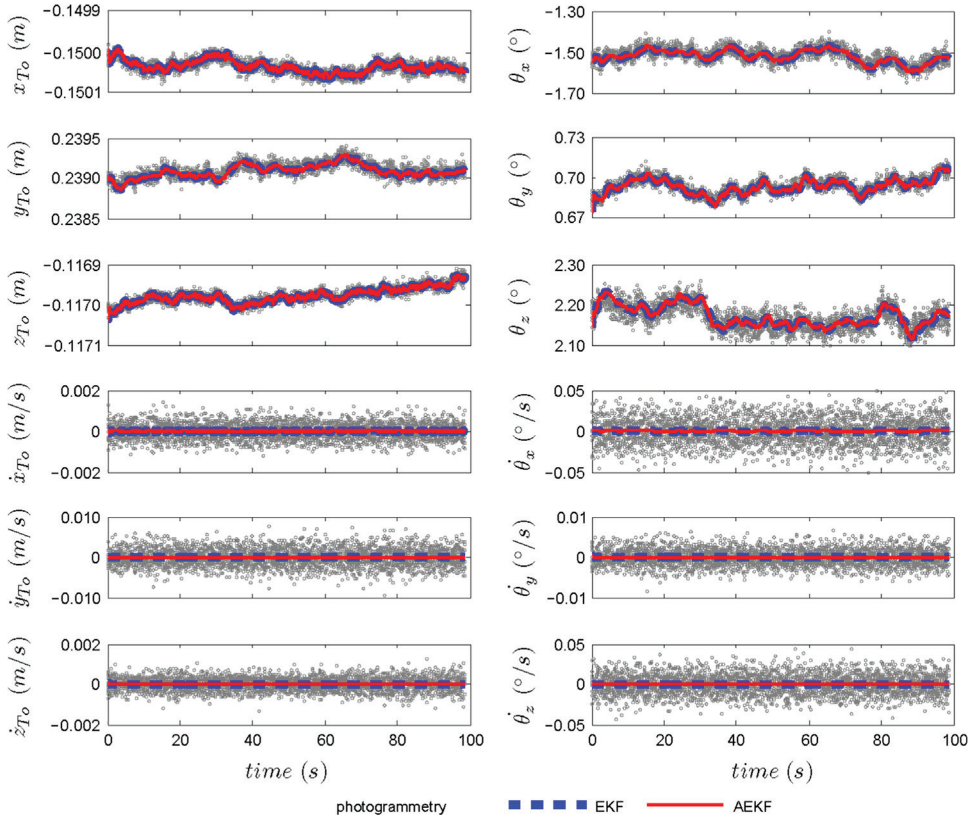


Fig. 4. Test results of estimation of a stationary target.

propagated downstream to the pose estimation. This is evident in Fig. 4, where the poses estimated by the photogrammetry are quite noisy even in this static case. In contrast, the EKF and proposed AEKF algorithms that considered the measurement noises successfully suppressed the spread of the measurement noises. The pose estimates by the EKF and proposed AEKF algorithm are very close in this static case and are much smoother than the photogrammetry as expected.

Next, the dynamic pose and motion estimation was conducted by manually translating and rotating the target along three axes of the target frame. The test results of dynamic pose and motion estimation by the EKF and the proposed AEKF algorithm are shown in Fig. 5 together with the photogrammetry. The EKF employs the Jacobian matrix of measurement model, which introduces linearization error. The stronger the nonlinearity, the greater the error may be. According to the measurement model, the nonlinearity of $y_{To}, \theta_x, \theta_z$ are relatively stronger than x_{To}, z_{To}, θ_y. Referring to the dynamic pose estimation results in Fig. 5, the estimates ofx_{To}, z_{To}, θ_y by EKF agree with the results of photogrammetry and the proposed algorithm much better than the estimates of $y_{To}, \theta_x, \theta_z$ when sudden changes of rotation occurred. As expected, the estimates of the proposed AEKF algorithm match the photogrammetry much better than the EKF.

Once the pose and motion estimation algorithm is validated, three capture cases: static, smooth and rough dynamic captures were conducted. In the experiments, the robotic manipulator always started from a home position (the corresponding joint angles are torso 0°, shoulder 90° and elbow 0°, as shown in Fig. 1).

In the static capture, the target was in an unknown static state within the workspace and the robotic manipulator attempted to make a capture autonomously. The desired and actual joint angles are changing simultaneously while the end-effector approaching the target, as shown in Fig. 6. This is because when the robotic manipulator attempts to make a capture, the position of the eye-in-hand camera is also affected by the robot motion.

In the smooth dynamic capture, the target was programmed to move within the workspace or capture zone of the robotic manipulator along a smooth but unknown trajectory while the robotic manipulator attempted a capture. In the current experiment, the capture zone in the target motion plane (*xz*-plane in the target frame as shown Fig. 2) was roughly 1.0 m (*x*-axis) by 0.8 m (*z*-axis) with the diagonal around 1.2 m due to the physical limits of our manipulator. The target was initially at the lower left corner and traveled diagonally through the capture zone to the upper right corner, pulled by a single stepper motor at the upper right corner. To ensure a sufficient capture window for the system to lock, track and capture the target, the linear speed of the target in the experiment was estimated based on the maximum rotating speed of the stepper motor at the joint of manipulator. Accordingly, the maximum linear speed of the grasping point on the target was designed to 1.2 cm/s such that the maximum capture

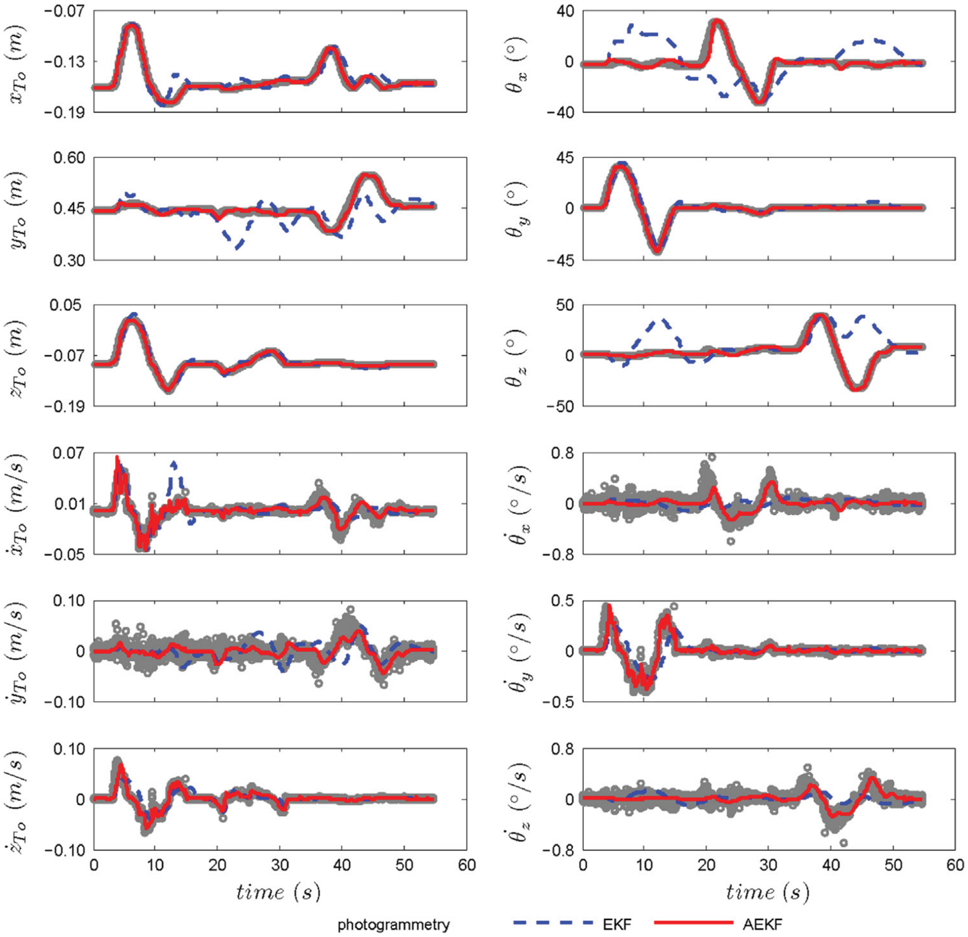


Fig. 5. Test results of estimation of a dynamic target.

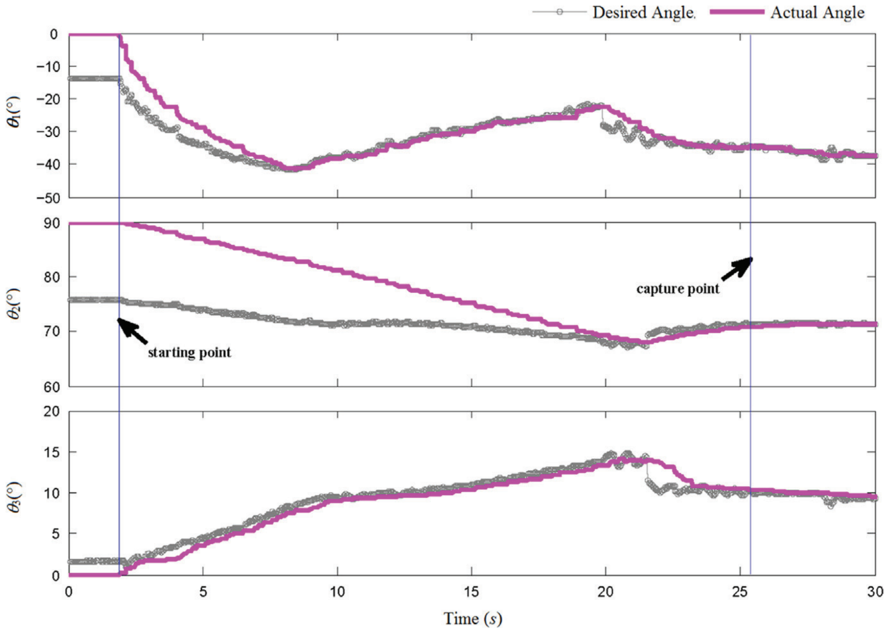


Fig. 6. Test results of a static capture.

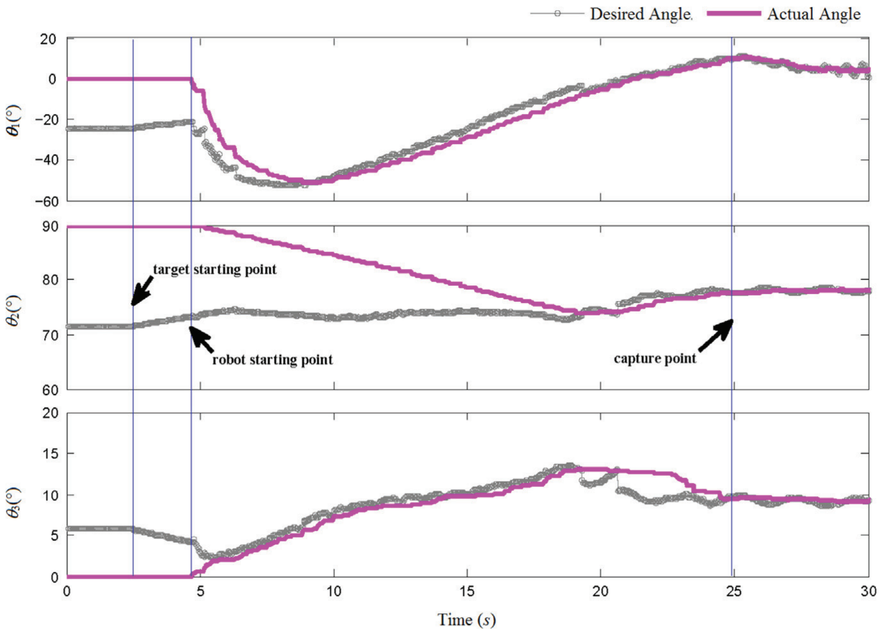


Fig. 7. Test results of a dynamic capture with a smooth trajectory.

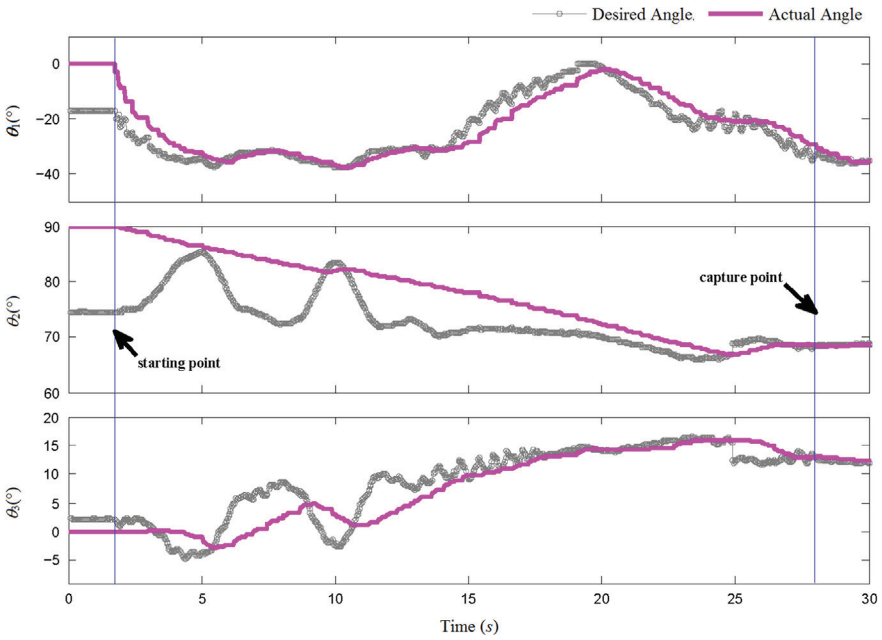


Fig. 8. Test results of rough target trajectory dynamic capture.

window for the robotic manipulator is around 100 s. The maximum linear velocity of the end–effector is subject to the configuration of the robotic manipulator as per Eq. (16). From Fig. 7, it shows that the actual capture time, from the system starting to move to the gripper capturing the target, in the experiments was around 25 s. Therefore, the target speed is reasonably fast for the current experimental setup. As shown in Fig. 7, the target motion caused the estimated desired joint angles varied ahead of the actual joint angles since the end-effector was approaching the interception instead of the current position. Fig. 7 shows the manipulator successfully captured the dynamic target.

Finally, a rough dynamic capture of the target was tested, where some perturbations were added to the target trajectory. As shown in Fig. 8, the desired joint angles varied dramatically. This is the evidence of accurate estimates of the target obtained by the proposed AEKF approach. After the target was locked, the manipulator tracked, approached and captured the target autonomously based the proposed AEKF algorithm. Due to the rough motion of the target observed, the desired joint angle as well as the actual joint angle are both affected based on the newly proposed control strategy. Therefore, the capture time was relatively longer than the smooth case.

In summary, the successful capture results in different capture scenarios, as shown in Figs. 6–8, demonstrate the effectiveness of the proposed AEKF algorithm and capture strategy.

6. Conclusions

This paper proposed a vision-based pose and motion estimation hybrid algorithm and a visual servo scheme for robotic manipulators to perform autonomous capture of a non-cooperative target. The hybrid algorithm is derived based on the adaptive extended Kalman filter and photogrammetry for the real-time state estimation of a non-cooperative target. Experiments have been designed and conducted by a custom-built six degrees of freedom robotic manipulator with an eye-in-hand configuration. The experiments demonstrate the proposed hybrid algorithm is robust in estimating the pose and motion of the non-cooperative target. The successful capture of a non-cooperative target by a visual servo robotic manipulator in a dynamic environment demonstrate the feasibility and effectiveness of the proposed estimation algorithm and the visual servo control strategy.

Acknowledgment

This work is supported by the discovery Grant (341917-2013) of the Natural Sciences and Engineering Research Council of Canada (NSERC).

Reference

[1] J. Yuh, Design and control of autonomous underwater robots: a survey, Auton. Robot. 8 (2000) 7–24.

[2] S.S. Mehta, T.F. Burks, Vision-based control of robotic manipulator for citrus harvesting, Comput. Electron. Agric. 102 (2014) 146–158.

[3] S.J. Lee, S.C. Lee, H.S. Ahn, Design and control of tele-matched surgery robot, Mechatronics 24 (2014) 395–406.

[4] L. Pedersen, D. Kortenkamp, D. Wettergreen, I. Nourbakhsh, A survey of space robotics, in: Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space, 2003, pp. 19–23.

[5] B.P. Larouche, Z.H. Zhu, Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control, Auton. Robot. 37 (2014) 157–167.

[6] M. Sabatini, R. Monti, P. Gasbarri, G.B. Palmerini, Adaptive and robust algorithms and tests for visual-based navigation of a space robotic manipulator, Acta Astronaut. 83 (2013) 65–84.

[7] A. Pisculli, L. Felicetti, P. Gasbarri, G. Palmerini, M. Sabatini, A reaction-null/Jacobian transpose control strategy with gravity gradient compensation for on-orbit space manipulators, Aerosp. Sci. Technol. 38 (2014) 30–40.

[8] P. Gasbarri, A. Pisculli, Dynamic/control interactions between flexible orbiting space-robot during grasping, docking and post-docking manoeuvres, Acta Astronaut. 110 (2015) 225–238.

[9] M. Kazemi, K. Gupta, M. Mehrandezh, Path planning for image-based control of wheeled mobile manipulators, in: Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 5306–5312.

[10] M.H. Ghasemi, N. Kashiri, M. Dardel, Time-optimal trajectory planning of robot manipulators in point-to-point motion using an indirect method, Proc. Inst. Mech. Eng., Part C: J. Mech. Eng. Sci. 226 (2011) 473–484.

[11] Q. Zhang, S.-R. Li, X.-S. Gao, Practical smooth minimum time trajectory planning for path following robotic manipulators, in: Proceedings of the IEEE American Control Conference (ACC), 2013, pp. 2778–2783.

[12] C. Liu, X.H. Huang, M. Wang, Target tracking for visual servoing systems based on an adaptive kalman filter, Int. J. Adv. Robot. Syst. 9 (2012) 1–12.

[13] T.W. Lim, Point cloud modeling using the homogeneous transformation for non-cooperative pose estimation, Acta Astronaut. 111 (2015) 61–76.

[14] D. Kragic, H.I. Christensen, Survey on Visual Servoing for Manipulation, Computational Vision and Active Perception Laboratory, Royal Institute of Technology, Stockholm, Sweden, 2002.

[15] T. Tzschichholz, L. Ma, K. Schilling, Model-based spacecraft pose estimation and motion prediction using a photonic mixer device camera, Acta Astronaut. 68 (2011) 1156–1167.

[16] G. Flandin, F. Chaumette, E. Marchand, Eye-in-hand/eye-to-hand cooperation for visual servoing, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'00, IEEE, 2000, pp. 2741–2746.

[17] G.J. Garcia, J.A. Corrales, J. Pomares, F. Torres, Survey of visual and force/tactile control of robots for physical interaction in Spain, Sensors 9 (2009) 9689–9733.

[18] F. Chaumette, S. Hutchinson, Visual servo control I. Basic approaches, IEEE Robot. Autom. Mag. 13 (2006) 82–90.

[19] F. Chaumette, S. Hutchinson, Visual servo control-part II: advanced approaches, IEEE Robot. Autom. Mag. 14 (2007) 109–118.

[20] C. Martínez, I.F. Mondragón, M.A. Olivares-Méndez, P. Campoy, On-board and ground visual pose estimation techniques for UAV control, J. Intell. Robot. Syst. 61 (2011) 301–320.

[21] K. Konolige, M. Agrawal, J. Sola, Large-scale visual odometry for rough terrain, in: Robotics Research, Springer, 2011, pp. 201–212.

[22] F. Janabi-Sharifi, M. Marey, A. Kalman-Filter-Based, Method for pose estimation in visual servoing, IEEE Trans. Robot. 26 (2010) 939–947.

[23] G. Caron, A. Dame, E. Marchand, Direct model based visual tracking and pose estimation using mutual information, Image Vis. Comput. 32 (2014) 54–63.

[24] R.E. Kalman, A new approach to linear filtering and prediction problems, ASME J. Basic Eng. 82 (1960) 35–45.

[25] S.Y. Chen, Kalman filter for robot vision: a survey, IEEE Trans. Ind. Electron. 59 (2012) 4409–4420.

[26] G. Dong, Z. Zhu, Position-based visual servo control of autonomous robotic manipulators, Acta Astronaut. 115 (2015) 291–302.

[27] K.A. Myers, B.D. Tapley, Adaptive sequential estimation with unknown noise statistics, IEEE Trans. Autom. Control 21 (1976) 520–523.

[28] V. Lippiello, B. Siciliano, L. Villani, Adaptive extended Kalman filtering for visual motion estimation of 3D objects, Control Eng. Pract. 15 (2007) 123–134.

[29] I. Blanchet, C. Frankignoul, M.A. Cane, A comparison of adaptive Kalman filters for a tropical Pacific Ocean model, Mon. Weather Rev. 125 (1997) 40–58.

[30] Q. Song, J.-D. HAN, An adaptive UKF algorithm for the state and parameter estimations of a mobile robot, Acta Autom. Sin. 34 (2008) 72–79.

[31] http://opencv.org/, Visited on July 23, 2015.

Reference Paper C

Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research 57 (2016) 1508–1514

Advances in Space Research
(a COSPAR publication)

www.elsevier.com/locate/asr

Incremental inverse kinematics based vision servo for autonomous robotic capture of non-cooperative space debris

Gangqi Dong, Z.H. Zhu *

Department of Earth and Space Science and Engineering, York University, 4700 Keele Street, Toronto, Ontario M3J 1P3, Canada

Received 31 October 2015; received in revised form 31 December 2015; accepted 12 January 2016
Available online 21 January 2016

Abstract

This paper proposed a new incremental inverse kinematics based vision servo approach for robotic manipulators to capture a non-cooperative target autonomously. The target's pose and motion are estimated by a vision system using integrated photogrammetry and EKF algorithm. Based on the estimated pose and motion of the target, the instantaneous desired position of the end-effector is predicted by inverse kinematics and the robotic manipulator is moved incrementally from its current configuration subject to the joint speed limits. This approach effectively eliminates the multiple solutions in the inverse kinematics and increases the robustness of the control algorithm. The proposed approach is validated by a hardware-in-the-loop simulation, where the pose and motion of the non-cooperative target is estimated by a real vision system. The simulation results demonstrate the effectiveness and robustness of the proposed estimation approach for the target and the incremental control strategy for the robotic manipulator.
© 2016 COSPAR. Published by Elsevier Ltd. All rights reserved.

Keywords: Visual servo; Autonomous capture; Non-cooperative target; Space debris; Active debris removal; Space robotic manipulator

1. Introduction

The increasing population of space debris in low and geo-stationary Earth orbits severely threats the safety of orbiting satellites and the long-term sustainability of space activities (Jankovic et al., 2015). To address the threat on a global base, the Inter-Agency Space Debris Coordination Committee (IADC) has suggested that certain remediation measures must be taken to stabilize the increasing trend of space debris population, for instance, by active debris removal (ADR) of a few large space debris per year from some crowded altitudes and inclinations of orbits (Liou, 2011). Numerous debris removal technologies have been proposed and investigated, such as the robotic debris removal (Jankovic et al., 2015), hybrid propulsion module (DeLuca et al., 2013), harpoon technology (Dudziak et al., 2015), and concepts considering end-of-mission self-deorbit by electrodynamic tethers (Zhong and Zhu, 2013), etc. Due to the similarity between the robotic on-orbit servicing (OOS) and ADR missions, the concept of autonomous ADR missions using space robotic manipulators is appealing in terms of technology readiness level. Although numerous human-in-the-loop OOS missions involving robotic captures of spacecraft were successfully performed (Yoshida, 2009), a fully autonomous robotic capture in space, especially considering non-cooperative objects is still an open subject facing enormous technical challenges (Flores-Abad et al., 2014). Recently, a preliminary concept design of guidance, navigation and control architecture to enable a safe and fuel-efficient capture of a non-cooperative target had been proposed (Jankovic et al., 2015), where the attention was focused on the close range autonomous rendezvous and proximity maneuver. In this paper, we focus

* Corresponding author. Tel.: +1 416 7362100x77729.
E-mail address: gzhu@yorku.ca (Z.H. Zhu).

http://dx.doi.org/10.1016/j.asr.2016.01.011

on the autonomous capture of a non-cooperative target by a robotic manipulator after the orbit maneuver being completed.

One of the most challenging tasks in the autonomous capture of a non-cooperative target is the identification of target's kinematic state. Considering the non-cooperative nature, the non-intrusive vision based filtering methods have been extensively adopted in the pose estimation of target (Aghili, 2012; Chen, 2012; Gasbarri et al., 2014; Janabi-Sharifi and Marey, 2010; Sabatini et al., 2013). Once the position and velocity of a target are obtained, an effective controller is required to control a robotic manipulator to capture the target autonomously. Ideally, the interception point between trajectories of the target and the end-effector in capture scenarios should be used as the desired position of the end-effector in control (Liu et al., 2015). However, due to the non-cooperative nature, the trajectory of target is unknown in advance and the determination of the interception point becomes a challenging task. The task is further complicated by the fact that the velocity of the end-effector is related to the configuration (joint angles) of robotic manipulator, which is time-variant and nonlinear. Thus, the interception point is also subject to the variation of interception time. In order to address this challenge, a kinematics based incremental control strategy for the robotic manipulator is proposed and examined in this work. Since the capture process of a space debris is relatively slow, it is more intuitive to regard the joint position (joint angles) as control input to gain higher control reliability instead of velocity or acceleration. The paper is organized as follows. Followed by this brief introduction, Section 2 is dedicated to the vision based kinematic state estimation of a non-cooperative target by an integrated photogrammetry and extended Kalman filter approach. A kinematics based incremental controller for the robotic manipulator is then presented in Section 3. Section 4 is dedicated to the validation by hardware-in-the-loop simulation and discussion. Finally, Section 5 concludes the paper.

2. Vision based kinematic identification of non-cooperative target

Consider a robotic manipulator system shown in Fig. 1. Assume a global frame is attached to the fixed part of the robotic manipulator, a camera frame is fixed to the center of image plane, and a target frame to the center of rotation of the target, respectively. The transformation between the global and the camera frame can be easily obtained according to the system configuration. The pose of a target can be described with respect to (w.r.t.) the camera frame, such as, $\{x_{To}, y_{To}, z_{To}, \theta_x, \theta_y, \theta_z\}^T$, where $\{x_{To}, y_{To}, z_{To}\}^T$ is the origin of the target frame in the camera frame and $\{\theta_x, \theta_y, \theta_z\}^T$ are the Euler angles of the target frame w.r.t. the camera frame.

Accordingly, an augmented homogeneous transformation between the target and the camera frame can be written as

$$\begin{Bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{Bmatrix} = \begin{bmatrix} & & & x_{To} \\ & \mathbf{R}_{TC} & & y_{To} \\ & & & z_{To} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{Bmatrix} \tag{1}$$

where $\mathbf{R}_{TC}$ denotes the rotational matrix from the target to the camera frame formed by the composition of trigonometric functions of Euler angles, $\{x_T, y_T, z_T\}^T$ denote the coordinates of a feature point on the target in the target frame and $\{x_C, y_C, z_C\}^T$ denote the corresponding coordinates of the same feature point in the camera frame.

Consider a pinhole camera with a focal length f. By assuming the y-axis of the camera frame pointing toward the target, the feature point is projected onto the image plane by

$$\begin{Bmatrix} x_m \\ z_m \end{Bmatrix} = -\frac{f}{y_C - f} \begin{Bmatrix} x_C \\ z_C \end{Bmatrix} \tag{2}$$

where $\{x_m, z_m\}^T$ denotes the measurable image coordinates.

Substituting Eq. (1) into (2) yields two independent equations for one feature point, which contains six unknowns: $\{x_{To}, y_{To}, z_{To}, \theta_x, \theta_y, \theta_z\}^T$. Theoretically, one needs at least three feature points to solve for the six unknowns. However, four feature points are widely adopted in literature to avoid the ambiguity and increase the robustness of algorithm (Dong and Zhu, 2015). Consequently, there will be eight equations with six unknowns, which can be solved by an iterative least square approach with an initial guess. The pose estimation of a non-cooperative target by the photogrammetry is a Markov process based on the current measurement, which is prone to the measurement noise. Moreover, the computational time of photogrammetry may vary widely due to the initial guess used in the algorithm at each time instant. As a result, the system sampling time interval may be affected, which is undesirable for real-time control. Another issue of the photogrammetry is that it does not solve for motion directly, which is an important control parameter in capturing the target by a robotic manipulator autonomously in a dynamic scenario. To address these challenges, an integrated photogrammetry and extended Kalman filter (EKF) is presented as follows.

Define the system variable vector as

$$\mathbf{X} = \{x_{To}, \dot{x}_{To}, y_{To}, \dot{y}_{To}, z_{To}, \dot{z}_{To}, \theta_x, \dot{\theta}_x, \theta_y, \dot{\theta}_y, \theta_z, \dot{\theta}_z\}^T$$

Assume the target motion can be approximated as a linear motion within each sampling time interval t_s if it is sufficiently small. Thus, the system model of the target can be expressed as

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{B}\boldsymbol{\omega}_{k-1} \tag{3}$$

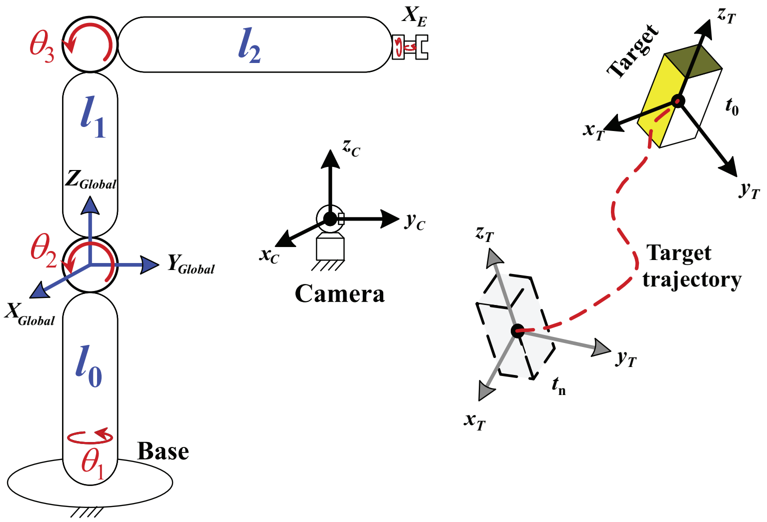


Fig. 1. Illustration of different coordinate systems.

where the subscripts k and $k-1$ denotes the current and previous states respectively, $\boldsymbol{\omega} = \{\ddot{x}_{To}, \ddot{y}_{To}, \ddot{z}_{To}, \ddot{\theta}_x, \ddot{\theta}_y, \ddot{\theta}_z\}^T$ is the acceleration vector that is considered as the process noise subject to the normal distribution with zero mean and covariance matrix $\mathbf{Q}$, and the coefficient matrices $\mathbf{A}$ and $\mathbf{B}$ are defined as

$$\mathbf{A} = diag[A \quad A \quad A \quad A \quad A \quad A], \quad A = \begin{bmatrix} 1 & t_s \\ 0 & 1 \end{bmatrix} \tag{4}$$

$$\mathbf{B} = diag[B \quad B \quad B \quad B \quad B \quad B], \quad B = \begin{Bmatrix} t_s^2/2 \\ t_s \end{Bmatrix} \tag{5}$$

Based on the pinhole camera from Eq. (2), the measurement model is defined as,

$$\mathbf{Z}_k = \mathbf{h}(\mathbf{X}_k) + \boldsymbol{\mu}_k \tag{6}$$

where $\mathbf{h}(\mathbf{X}) = -\frac{f}{y_C - f}\begin{Bmatrix} x_C \\ z_C \end{Bmatrix}$ is an equation vector for one feature point, $\boldsymbol{\mu}$ stands for the measurement noise that obeys the normal distribution with zero mean and covariance matrix $\mathbf{R}$, and $\mathbf{Z}$ is the real measurement vector of feature points.

Once the system and measurement models are derived, the EKF is applied to estimate the kinematic state of the target, such that,

$$\begin{cases} \mathbf{X}_{k|k-1} = \mathbf{A}\mathbf{X}_{k-1|k-1} \\ \mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T + \mathbf{B}\mathbf{Q}\mathbf{B}^T \\ \mathbf{K}_g = \mathbf{P}_{k|k-1}\mathbf{H}^T[\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}]^{-1} \\ \mathbf{X}_{k|k} = \mathbf{X}_{k|k-1} + \mathbf{K}_g[\mathbf{Z}_k - \mathbf{h}(\mathbf{X}_{k|k-1})] \\ \mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_g\mathbf{H}\mathbf{P}_{k|k-1} \end{cases} \tag{7}$$

where $\mathbf{H}$ is the Jacobian matrix of the measurement model, $\mathbf{P}$ is the covariance matrix of the system variable, and $\mathbf{K}_g$ is the Kalman gain.

As well known, the input of EKF includes initial conditions known in advance and measurements observed over time. Since initial conditions of a non-cooperative target are unknown, inappropriate initial guess may lead to poor performance of the EKF. Therefore, we initialized the EKF by the photogrammetry in order to improve the performance as well as accelerate the convergence rate of the EKF.

3. Kinematics based incremental control strategy

The forward kinematics of a robotic manipulator defines the position and velocity of the end-effector in terms of the corresponding joint variables, such that

$$\mathbf{X}_E = \mathbf{f}(\boldsymbol{\Theta}) \tag{8}$$

$$\dot{\mathbf{X}}_E = \mathbf{J}\dot{\boldsymbol{\Theta}} \tag{9}$$

where $\mathbf{X}_E \in \mathbf{R}^m$ is the position of the end-effector in the Cartesian space, $\boldsymbol{\Theta} \in \mathbf{R}^n$ is the generalized joint variable vector in the joint space, the overhead dot denotes the time derivative and $\mathbf{J}$ is the Jacobian matrix of the robotic manipulator defined by $\mathbf{J} = \partial\mathbf{f}/\partial\boldsymbol{\Theta}$.

In order to perform the capture task, one has to determine the joint angles by the inverse kinematics of robotic manipulator based on the information of the end-effector. The inverse of Eq. (8) may have multiple solutions due to system redundancy and periodicity of trigonometric functions. In order to obtain unique solution, physical limits or other extra constraints may have to be imposed. Generally, the complete inverse of Eq. (9) is composed by two parts, the particular solution and the null space solution. The particular solution can be obtained based on the pseudo inverse of the Jacobian matrix, denoted by $\mathbf{J}^\dagger$. Although the pseudo-inverse of the Jacobian matrix can be calculated by using the singular value decomposition, the computational load is relatively heavy. In many robotic applications, the Jacobian matrix is assumed to have full row rank, and the pseudo-inverse is replaced by the right inverse, such that, $\mathbf{J}^{\dagger} = \mathbf{J}^{T}(\mathbf{J}\mathbf{J}^{T})^{-1}$. Thus the complete solution for the inverse of Eq. (9) is written as

$$\dot{\boldsymbol{\Theta}} = J^{\dagger}\dot{\mathbf{X}}_E + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\xi \tag{10}$$

where $\mathbf{I}$ denotes the identity matrix and ξ is an arbitrary vector that projected onto the null space of the Jacobian. Therefore, by careful selection of this vector, additional kinematic objectives can be achieved if the kinematic redundancy exists.

Once the position and velocity of the target w.r.t. the camera frame is obtained in Section 2, we may easily transform them to the global frame by the homogeneous transformation, such that

$$\begin{pmatrix}\mathbf{X}_T\\1\end{pmatrix} = \mathbf{T}\begin{pmatrix}{}^{C}\mathbf{X}_T\\1\end{pmatrix} \tag{11}$$

$$\begin{pmatrix}\dot{\mathbf{X}}_T\\1\end{pmatrix} = \mathbf{T}\begin{pmatrix}{}^{C}\dot{\mathbf{X}}_T\\1\end{pmatrix} + \dot{\mathbf{T}}\begin{pmatrix}{}^{C}\mathbf{X}_T\\1\end{pmatrix} \tag{12}$$

where $\mathbf{X}_T, \dot{\mathbf{X}}_T$ and ${}^{C}\mathbf{X}_T, {}^{C}\dot{\mathbf{X}}_T$ stand for the target position and velocity w.r.t. the global and camera frame respectively, $\mathbf{T}$ denotes the Denavit–Hartenberg transformation matrix from the camera frame to the global frame (Denavit and Hartenberg, 1955), which can be easily obtained according to the geometric configuration of the system.

In the eye-to-hand configuration, the camera is stationary in the global frame. Therefore, the transformation matrix between the two camera and global frames will be time-invariant and the time derivative of the transformation matrix is zero matrix, such that, $\dot{\mathbf{T}} = \mathbf{0}$ in Eq. (12).

Once the position and velocity of the target in the global frame is obtained, the position of the target in the next moment can be estimated by

$$\mathbf{X}_{T'} = \mathbf{X}_T + \dot{\mathbf{X}}_T t_s \tag{13}$$

In order to make a capture, the most straightforward way for the end-effector to approach the target is directly towards it if the velocity of the target is relatively small compare to the robotic manipulator. Accordingly, the instantaneous velocity of the end-effector in the Cartesian space should be along with the vector defined by $\mathbf{X}_{T'} - \mathbf{X}_E$.

Denote the unit vector of $\mathbf{X}_{T'} - \mathbf{X}_E$ as

$$\mathbf{n}_{ET'} = \frac{\mathbf{X}_{T'} - \mathbf{X}_E}{\|\mathbf{X}_{T'} - \mathbf{X}_E\|} \tag{14}$$

Define the scale factor of the instantaneous velocity as $\lambda > 0$, so that the instantaneous velocity of the end-effector can be written as

$$\dot{\mathbf{X}}_E = \lambda\mathbf{n}_{ET'} \tag{15}$$

For the sake of simplicity, assume $\boldsymbol{\xi} = \mathbf{0}$ in Eq. (10). Then, substituting Eq. (15) into Eq. (10) leads to

$$\dot{\boldsymbol{\Theta}} = \lambda\mathbf{J}^{\dagger}\mathbf{n}_{ET'} \tag{16}$$

Let $\dot{\theta}_i, p_{ij}$ and $n_{ET'j}$ be the elements of $\dot{\boldsymbol{\Theta}}, \mathbf{J}^{\dagger}$ and $\mathbf{n}_{ET'}$, respectively. Then, Eq. (16) can be replaced by n scalar equations, such that

$$\dot{\theta}_i = \lambda\sum_{j=1}^{m} p_{ij} n_{ET'j}, \quad i = 1, 2, \cdots, n. \tag{17}$$

Due to the physical limit of the joint actuators and the transmission mechanisms, the motion of the robotic manipulator is subject to the joint velocity limit, denoted by $\dot{\boldsymbol{\Theta}}_{\max} = (\dot{\theta}_{\max 1}, \dot{\theta}_{\max 2}, \cdots, \dot{\theta}_{\max n})^{T}$. Assume the joint velocity has the same limit in both positive and negative directions, then the right side of equations in Eq. (17) is bounded by the closed interval $[-\dot{\theta}_{\max i}, \dot{\theta}_{\max i}]$, respectively, such that

$$-\dot{\theta}_{\max i} \leqslant \lambda\sum_{j=1}^{m} p_{ij} n_{ET'j} \leqslant \dot{\theta}_{\max i}, \quad i = 1, 2, \cdots, n. \tag{18}$$

Solving inequalities in Eq. (18) for λ yields n sets of scale factor λ_i. Then, the maximum scale factor of the instantaneous velocity can be found by taking the intersection of each solution, such as, $\lambda_{\max} = \max[\lambda_1 \cap \lambda_2 \cap \cdots \cap \lambda_n]$. By substituting $\lambda_{\max}$ into Eq. (16), we obtain the equivalent instantaneous joint velocity of the robotic manipulator, denoted by $\dot{\boldsymbol{\Theta}}_v$. Accordingly, the incremental joint angle control input for the robotic manipulator in the next moment, which will drive the end-effector directly towards the target, can be defined as,

$$\Delta\boldsymbol{\Theta} = \dot{\boldsymbol{\Theta}}_v t_s \tag{19}$$

The control input of the robotic manipulator could be $\boldsymbol{\Theta} + \Delta\boldsymbol{\Theta}$ if the control is achieved by absolute joint angles. After the end-effector moving towards the target by an increment $\Delta\mathbf{X}_E = \mathbf{J}\Delta\boldsymbol{\Theta}$, the procedure defined in Eqs. (13)–(19) is repeated for the next sampling time. A capture will be done once the position error between the end-effector and the target is within a pre-defined tolerance, denoted by $\boldsymbol{\varepsilon} \in \mathbf{R}^{m}$, such that,

$$|\mathbf{X}_E - \mathbf{X}_T| < \boldsymbol{\varepsilon} \tag{20}$$

4. Validation by hardware-in-the-loop simulation

The proposed kinematics based incremental control strategy is validated by hardware-in-the-loop simulation. The robotic manipulator is assumed to consist of three rigid links and three revolute joints. The visual control is done by an eye-to-hand configuration. The link length and the joint velocity limits of the manipulator are given in Table 1, based on our previous work (Dong and Zhu, 2015).

The joint angles are measured from their home positions $\theta_1 = 0°, \theta_2 = 90°, \theta_3 = 0°$, as illustrated in Fig. 1. The global frame is located to the center of the second joint and fixed in the inertia space, and the camera frame is attached

Table 1
Physical properties of the robotic manipulator.

Parameter	Value
l_0	0.1778 m
l_1	0.454025 m
l_2	0.4445 m
$\dot{\theta}_{\max 1}$	10.304 rad/s
$\dot{\theta}_{\max 2}$	0.412 rad/s
$\dot{\theta}_{\max 3}$	0.412 rad/s

to the center of camera's image plane, which coincides with the home position of the end-effector for simplicity. Accordingly, the Jacobian matrix of the robotic manipulator can be easily obtained. In order to simplify the calculation and reduce the computational cost in our case, the kinematic singularity of the robotic manipulator is physically avoided by limiting its workspace. Then, the pseudo-inverse of the Jacobian, which will be full rank, can be calculated simply by $\mathbf{J}^{\dagger} = \mathbf{J}^{-1}$.

The non-cooperative target is independently moving along a trajectory within the 3D workspace of the robotic manipulator. Its motion and trajectory are unknown in advance and must be estimated by the vision system. Fig. 2 shows the estimated position $\mathbf{X}_T = (x_T, y_T, z_T)^T$ and velocity $\dot{\mathbf{X}}_T = (\dot{x}_T, \dot{y}_T, \dot{z}_T)^T$ of the target by the eye-to-hand camera using the integrated approach presented in Section 2 w.r.t. the global frame. These profiles are input to the controller to calculate the incremental control input for the robotic manipulator.

The control input to the robotic manipulator in our case is the incremental joint angles. As shown clearly in Fig. 3, the calculated instantaneous joint velocities $\dot{\theta}_2$ and $\dot{\theta}_3$ reach the physical limits listed in Table 1 in certain periods, respectively. In general, the calculated instantaneous joint velocities $\dot{\mathbf{\Theta}}_v$ are quite smooth because the estimated position and velocity of the non-cooperative target in the global frame are relatively smooth. However, the control inputs $\Delta\mathbf{\Theta}$ in the Fig. 4 show isolated outliers from time to time although the overall trends are smooth. This is because the sampling time used in the controller is dictated by the image processing time, which varies from time to time. Accordingly, the incremental control input $\Delta\mathbf{\Theta}$ obtained by Eq. (19) is affected by the time-varying sampling time. If simulated target motion and velocity are used, there will be no outliers in the incremental control input. This indicates it is important to keep the sampling time constant in real implementation of visual servo control algorithm of robotic manipulator.

The simulation results of an autonomous capture by the robotic manipulator are shown in Fig. 4. The solid lines and circles denote the trajectories of the target and the end-effector respectively. The robotic manipulator is at the home position initially. By defining $\boldsymbol{\varepsilon} = (10^{-3}, 10^{-3}, 10^{-3})^T$ meters in Eq. (20), a capture is made

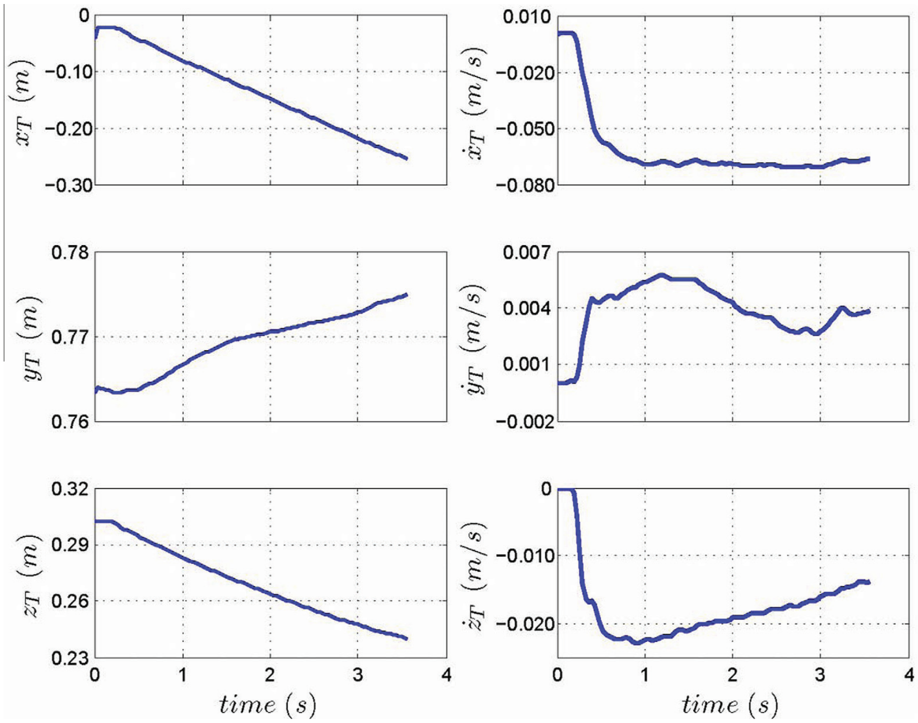


Fig. 2. Estimated position and velocity of the non-cooperative target in global frame.

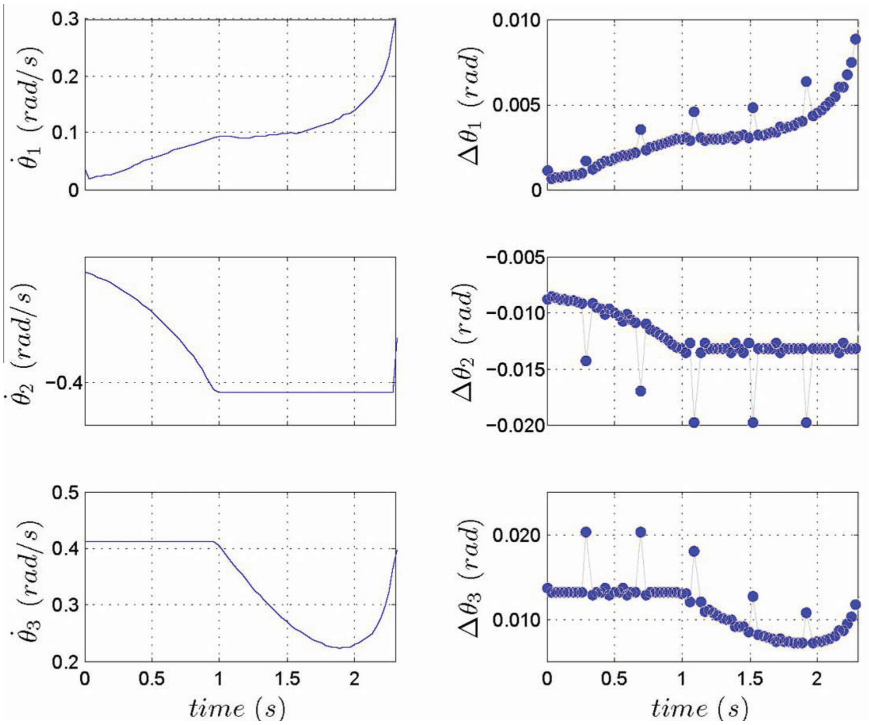


Fig. 3. Instantaneous joint velocity and incremental control input of the robotic manipulator.

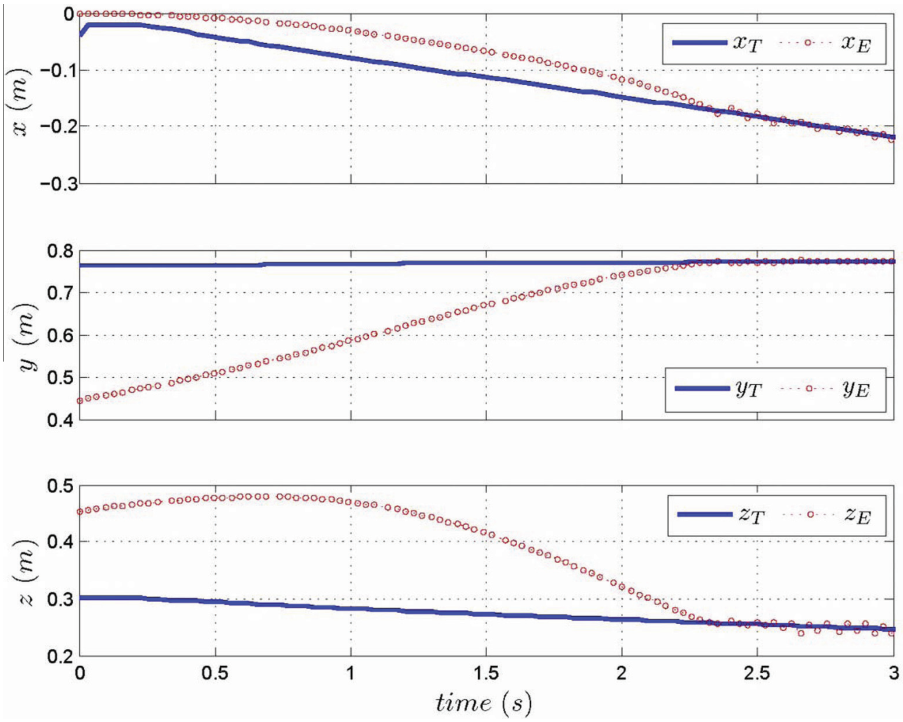


Fig. 4. Position of the target and the end-effector.

at about 2.3 s once the capture criteria is achieved. The position profiles of the end-effector in Fig. 4 are smooth and approximately monotonic, although there are some outliers in the incremental control input. This is generally preferred in the operation of robotic manipulators. The simulation results shown in Figs. 3 and 4 demonstrate the proposed kinematics based incremental control strategy is effective and efficient.

5. Conclusions

This paper proposed a new vision-based incremental kinematic control strategy for a robotic manipulator to perform autonomous capture of a non-cooperative target. The target's pose and motion are estimated visually by an integrated photogrammetry and EKF approach. Based on the input of target's pose and motion, the forward kinematics of robot manipulator is inversed in an incremental form where the robotic manipulator moves by increments from its current configuration. By adopting this approach, the multiple-solution problem of the inverse kinematics has been avoided. Since the velocity of the end-effector in Cartesian space in the current approach is always pointing to the target in the next moment, the possibility of losing track of the target by the vision system is greatly reduced. The proposed approach is validated by a hardware-in-the-loop simulation, where the pose and motion of the non-cooperative target is estimated by a real vision system. The simulation results of capture demonstrated proposed kinematics based incremental control strategy is effective and efficient.

Acknowledgements

This work is supported by the Natural Sciences and Engineering Research Council of Canada.

References

Aghili, F., 2012. A prediction and motion-planning scheme for visually guided robotic capturing of free-floating tumbling objects with uncertain dynamics. IEEE Trans. Rob. 28, 634–649.

Chen, S.Y., 2012. Kalman filter for robot vision: a survey. IEEE Trans. Ind. Electron. 59, 4409–4420.

DeLuca, L.T., Bernelli, F., Maggi, F., et al., 2013. Active space debris removal by a hybrid propulsion module. Acta Astronaut. 91, 20–33.

Denavit, J., Hartenberg, R.S., 1955. A kinematic notation for lower-pair mechanisms based on matrices. Trans. ASME J. Appl. Mech. 23, 215–221.

Dong, G., Zhu, Z.H., 2015. Position-based visual servo control of autonomous robotic manipulators. Acta Astronaut. 115, 291–302.

Dudziak, R., Tuttle, S., Barraclough, S., 2015. Harpoon technology development for the active removal of space debris. Adv. Space Res. 56, 509–527.

Flores-Abad, A., Ma, O., Pham, K., Ulrich, S., 2014. A review of space robotics technologies for on-orbit servicing. Prog. Aerospace Sci. 68, 1–26.

Gasbarri, P., Sabatini, M., Palmerini, G.B., 2014. Ground tests for vision based determination and control of formation flying spacecraft trajectories. Acta Astronaut. 102, 378–391.

Janabi-Sharifi, F., Marey, M., 2010. A Kalman-filter-based method for pose estimation in visual servoing. IEEE Trans. Rob. 26, 939–947.

Jankovic, M., Paul, J., Kirchner, F., 2015. GNC architecture for autonomous robotic capture of a non-cooperative target: preliminary concept design. Adv. Space Res.

Liou, J.C., 2011. An active debris removal parametric study for LEO environment remediation. Adv. Space Res. 47, 1865–1876.

Liu, X.F., Li, H.Q., Chen, Y.J., Cai, G.P., Wang, X., 2015. Dynamics and control of capture of a floating rigid body by a spacecraft robotic arm. Multibody Syst. Dyn. 33, 315–332.

Sabatini, M., Monti, R., Gasbarri, P., Palmerini, G.B., 2013. Adaptive and robust algorithms and tests for visual-based navigation of a space robotic manipulator. Acta Astronaut. 83, 65–84.

Yoshida, K., 2009. Achievements in space robotics expanding the horizons of service and exploration. IEEE Rob. Autom. Mag. 16, 20–28.

Zhong, R., Zhu, Z.H., 2013. Dynamics of nanosatellite deorbit by bare electrodynamic tether in low earth orbit. J. Spacecraft Rockets 50, 691–700.

Reference Paper D

ADVANCED ROBOTICS, 2016
VOL. 30, NO. 22, 1458–1465
http://dx.doi.org/10.1080/01691864.2016.1229633

FULL PAPER

Incremental visual servo control of robotic manipulator for autonomous capture of non-cooperative target

Gangqi Dong and Zheng H. Zhu

Department of Earth and Space Science and Engineering, York University, Toronto, Canada

ABSTRACT
This paper develops a new autonomous incremental visual servo control law for the robotic manipulator to capture a non-cooperative target, where the control input is the incremental joint angle to avoid the multiple solutions in the existing inverse kinematics. The position and motion of the non-cooperative target are estimated by an eye-to-hand vision system in real time by integrated photogrammetry and extended Kalman filter. The estimated position and motion of the target are fed into the newly developed position-based visual servo control law to drive the manipulator incrementally towards the dynamically predicted interception point between trajectories of the end effector and the target. To validate the proposed approach, a hardware-in-the-loop simulation has been conducted where the position and motion of the target is estimated by a real eye-to-hand camera and fed into the simulation of the robotic manipulator. The simulation results show the proposed incremental visual servo control law is stable and able to avoid the multiple solutions in the total inverse kinematics.

ARTICLE HISTORY
Received 19 February 2016
Revised 23 May 2016
Accepted 9 August 2016

KEYWORDS
Visual servo; autonomous robotic manipulator; incremental control; inverse kinematics; non-cooperative target

1. Introduction

Robotic manipulators have been widely used to assist humans to perform tasks in atrocious or even inaccessible environments.[1–3] Among the numerous applications, capture of non-cooperative targets is considered as one of the most challenging functions of robotic manipulators. Many manipulators adopted pre-programmed scripts to ensure high reliability, while other applications involve human-in-the-loop control to pass the majority of decision-making and guidance tasks into human hands. Due to the uncertainty of non-cooperative targets, autonomy is required in the robotic capture where human interference is limited and/or ineffective.[4–8]

In order to achieve the autonomy in a robotic capture, the pose and motion of target should be identified precisely in real time. Because the kinematic state of a non-cooperative target is unknown to the manipulator, vision systems are extensively employed to estimate the pose and motion of target due to their non-contact and non-intrusive nature.[6–13] Considerable efforts have been devoted to estimate the pose and motion of target from vision images in the field of visual servo of autonomous robotic manipulators. Generally, the vision systems used in visual servo consists of two categories in the literature: eye-in-hand and eye-to-hand.[14,15] The camera is mounted on the end effector in the eye-in-hand configuration while detached from the manipulator in the eye-to-hand configuration. Compared to the eye-in-hand configuration, the eye-to-hand camera monitors the whole workspace to ensure the target can always be detected and simplifies the coordinate transformation from the camera frame to the global frame. Once the pose and motion of target are estimated, the challenge arises in the development of a real-time control strategy for the robotic manipulator to perform autonomous capture. The manipulator has to follow an approaching trajectory defined by the time-varying desired position and/or velocity of the end effector to track, approach and capture the target. If the desired state of the end effector in the Cartesian space is determined, the corresponding state in the joint space can be derived by inverse kinematics. Unfortunately, the inverse kinematics may result in multiple solutions due to the periodicity of trigonometric functions or even becomes singular for a given position of the end effector.[16–18] To address the challenge, most existing works focused on the velocity-level and acceleration-level control.[19–21]

In the visual servo control, the desired position of the end effector is carefully designed based on the feedback

CONTACT Zheng H. Zhu gzhu@yorku.ca

of vision system at each sampling instant, and the robotic manipulator is controlled to move incrementally by joint actuators relative to the current angular position. Thus, one could control the motion of robotic manipulators by an increment from its current state to avoid the problem of multiple solutions in the inverse kinematics. Furthermore, if the current state of a dynamic target is used as the desired state of the end effector in the visual servo control, the end effector will require a large velocity at the last few steps, which may exceed the angular velocity limits of the actuators. To address the constraints, a constrained optimal control approach was introduced in [22]. In this paper, an intuitive approach is adopted by assuming the projected interception point between trajectories of the target and the end effector as the desired position of the end effector in the visual servo control. This is because the trajectory of the non-cooperative target is unknown to the manipulator in advance and time-varying as the target is moving during the tracking and capture period.

To address the challenges efficiently, we proposed a framework of incremental visual servo control strategy at the level of angular joint positions to perform autonomous robotic capture of a non-cooperative target for potential applications in space. Different from our previous work [23] where the target position in the next time instant is used as the desired position of the end effector, the current approach assumes the dynamically predicted interception points of trajectories of the end effector and the target as the desired position. The unknown trajectory of a non-cooperative target is estimated in real time by an integrated photogrammetry and extended Kalman filter approach developed in our previous work.[7] Furthermore, considering the fact that the motion of space robotic manipulators is relatively slow and many of them use stepper motors as actuators, a kinematic approach is adopted in the current study where the joint angle increments are used as the control input, instead of joint torques, to control the stepper motors directly. When the end effector is far from the predicted interception point, the end effector will move directly towards the predicted interception point at the maximum velocity of joint actuators. Once they are close enough, the end effector will approach the moving target with similar velocity because the visual system tracks the motion of the target in real time. The newly proposed concept is validated by a hardware-in-the-loop simulation where a real vision system is used to estimate the position and motion of the target. The estimates are then fed into the simulation where the motion of a robotic manipulator is simulated to demonstrate the effectiveness of our control law. It should be noted that the robotic manipulators in space are working on free floating bases, where the coupling effect among the manipulator, the free floating base spacecraft and the non-cooperative target complicates the design of visual servo control law of the manipulator. In order to prove the viability of the proposed incremental control, the robotic manipulator considered in this note is assumed on a fixed base to decouple the effect caused by the free floating base. The newly proposed incremental control will be integrated with the free floating manipulators in the future.

2. Position-level kinematic control strategy

Assume the position and motion of a rigid-body target is estimated by an eye-to-hand vision system in the camera frame. Then, the position of the target in the global frame is obtained by

$$\mathbf{X}_T = \mathbf{R}_C^{G\,C}\mathbf{X}_T + {}^{G}\mathbf{X}_{Co} \tag{1}$$

where $\mathbf{X}_T$ denotes the position vector of the target in the global frame, ${}^{C}\mathbf{X}_T$ denotes the position vector estimated in the camera frame, ${}^{G}\mathbf{X}_{Co}$ denotes the position vector of the camera frame origin in the global frame and $\mathbf{R}_C^G$ denotes the rotation matrix from the camera frame to the global frame.

Equation (1) can be rewritten by the homogeneous transform, such as

$$\begin{Bmatrix} \mathbf{X}_T \\ 1 \end{Bmatrix} = \mathbf{T}\begin{Bmatrix} {}^{C}\mathbf{X}_T \\ 1 \end{Bmatrix} \text{ and, } \mathbf{T} = \begin{bmatrix} \mathbf{R}_C^G & {}^{G}\mathbf{X}_{Co} \\ \mathbf{O} & 1 \end{bmatrix} \tag{2}$$

where $\mathbf{O}$ denotes a row vector of zeroes and $\mathbf{T}$ is the transformation matrix from the camera frame to the global frame.

Taking the first order time derivative on both sides of Equation (2) yields the velocity of the target in the global frame

$$\begin{Bmatrix} \dot{\mathbf{X}}_T \\ 0 \end{Bmatrix} = \mathbf{T}\begin{Bmatrix} {}^{C}\dot{\mathbf{X}}_T \\ 0 \end{Bmatrix} + \dot{\mathbf{T}}\begin{Bmatrix} {}^{C}\mathbf{X}_T \\ 1 \end{Bmatrix} \tag{3}$$

where the overhead dot denotes the time derivative.

For the eye-to-hand configuration, the camera frame is stationary and fixed in the global frame. Thus, the transformation matrix from the camera frame to the global frame is time-invariant and is time-derivative $\dot{\mathbf{T}}$ should be zero matrix in Equation (3).

Once the target motion is estimated with respect to the camera frame, the target motion with respect to the global frame is obtained by applying Equations (2) and (3). Therefore, unless otherwise stated, the target motion mentioned in the following is referred to the global frame.

The robotic manipulators generally consist of a set of rigid link and joint combinations as well as an end effector. The links and joints mainly contribute to the position of the end effector, while the end effector provides dexterous orientation and gripping functions to interact with the environment. The forward kinematics maps the motion of the manipulator in the joint space to the motion of the end effector in the Cartesian space, such that,

$$\mathbf{X}_E = \mathbf{f}(\boldsymbol{\Theta}) \tag{4}$$

$$\dot{\mathbf{X}}_E = \mathbf{J}\dot{\boldsymbol{\Theta}} \tag{5}$$

where $\mathbf{X}_E \in \mathbf{R}^m$ is the position of the end effector in the Cartesian space, $\boldsymbol{\Theta} \in \mathbf{R}^n$ is the generalized joint variable vector, and $\mathbf{J}$ is the Jacobian matrix defined by $\mathbf{J} = \partial \mathbf{f} / \partial \boldsymbol{\Theta}$.

For the given velocity of the end effector in the Cartesian space, the corresponding velocity of the robotic manipulator in the joint space can be derived by the inverse kinematics, such that

$$\dot{\boldsymbol{\Theta}} = \mathbf{J}^{\dagger}\dot{\mathbf{X}}_E + \left(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}\right)\boldsymbol{\xi} \tag{6}$$

where $\mathbf{J}^{\dagger}$ denotes the pseudo inverse of the Jacobian matrix, $\mathbf{I}$ is the identity matrix, $\boldsymbol{\xi}$ is an arbitrary vector that is projected onto the null space of the Jacobian, such that additional kinematic objectives can be achieved by the selection of the vector if the kinematic redundancy exists.

The inverse kinematics in Equation (6) describes the motion of the robotic manipulator in the joint space by the motion of the end effector in the Cartesian space. The pseudo inverse of the Jacobian matrix could be calculated using the geometric approach, such that

$$\mathbf{J}^{\dagger} = \mathbf{J}^T\left(\mathbf{J}\mathbf{J}^T\right)^{-1} \tag{7}$$

As well known, multiple solutions may exist in the position-level inverse kinematics even if the Jacobian is full row rank. To address the challenge, a new incremental kinematic approach is developed as follows.

Assume the target is always within the workspace of the robotic manipulator and the field of view of camera. The capture control strategy is defined that the capture should be made at the interception point $\mathbf{X}_C$ after time t measured from the current instant, which is estimated approximately in the global frame as

$$\mathbf{X}_C = \mathbf{X}_T + \dot{\mathbf{X}}_T t \tag{8}$$

where $\mathbf{X}_T$ and $\dot{\mathbf{X}}_T$ are the current instant position and velocity of the target.

In the same time period t, the end effector is required to reach the same interception point $\mathbf{X}_C$ to capture the target along the shortest path, such that,

$$\mathbf{X}_C = \mathbf{X}_E + \dot{\mathbf{X}}_E t \tag{9}$$

where $\mathbf{X}_E$ and $\dot{\mathbf{X}}_E$ are the current position and the velocity of the end effector.

It should be noted that both the interception point $\mathbf{X}_C$ and time t in Equations (8) and (9) are unknown and must be estimated because the target is moving.

Solving Equations (8) and (9) for the required velocity of the end effector $\dot{\mathbf{X}}_E$ yields

$$\dot{\mathbf{X}}_E = \frac{\mathbf{X}_T - \mathbf{X}_E}{t} + \dot{\mathbf{X}}_T \tag{10}$$

It should be noted from Equation (10) that the velocity of the end effector will approach the velocity of the target by the proposed control as it is getting close to the target.

Substituting Equation (10) into Equation (6) leads to

$$\dot{\boldsymbol{\Theta}} = \mathbf{J}^{\dagger}\left\{\frac{\mathbf{X}_T - \mathbf{X}_E}{t} + \dot{\mathbf{X}}_T\right\} \tag{11}$$

where $\boldsymbol{\xi} = \mathbf{0}$ in Equation (6) is assumed for simplicity.

Let $\dot{\theta}_i$, p_{ij}, x_{Tj}, $\dot{x}_{Tj}$ and x_{Ej} denote the elements of $\dot{\boldsymbol{\Theta}}$, $\mathbf{J}^{\dagger}$, $\mathbf{X}_T$, $\dot{\mathbf{X}}_T$ and $\mathbf{X}_E$, respectively. Then, Equation (11) is decomposed into n equations, such that

$$\dot{\theta}_i = \sum_{j=1}^{m} p_{ij}\left(\frac{x_{Tj} - x_{Ej}}{t} + \dot{x}_{Tj}\right), \quad i = 1, 2, \ldots, n. \tag{12}$$

The joint velocity limit, denoted by $\dot{\boldsymbol{\Theta}}_{\mathrm{max}} = \left(\dot{\theta}_{\mathrm{max}1}, \dot{\theta}_{\mathrm{max}2}, \cdots, \dot{\theta}_{\mathrm{max}n}\right)^T$, can be easily determined by the physical limit of the joint actuators and the transmission mechanisms. By assuming that the joint velocity has the same limit in both revolute directions (forward and reverse), we have the right-hand sides of equations in Equation (12) bounded by $\left[-\dot{\theta}_{\mathrm{max}i}, \dot{\theta}_{\mathrm{max}i}\right]$, respectively, such that,

$$-\dot{\theta}_{\max i} \le \sum_{j=1}^{m} p_{ij}\left(\frac{x_{Tj} - x_{Ej}}{t} + \dot{x}_{Tj}\right) \le \dot{\theta}_{\max i}, \quad i = 1, 2, \cdots, n. \tag{13}$$

Solving inequalities in Equation (13) with the consideration of $t > 0$ and intersecting of each solution of t yields the minimum interception time t_{min}. By substituting the minimum interception time into Equation (11), we obtain the joint velocity of the robotic manipulator, which will drive the end effector directly towards the interception point. Accordingly, the incremental joint angle control input within one sample time step t_s can be defined approximately as,

$$\Delta\boldsymbol{\Theta} = \dot{\boldsymbol{\Theta}} t_s \tag{14}$$

In the position-based visual servo control, the feedback position and velocity tracking errors are defined as [12,13],

$$\mathbf{E} = \mathbf{X}_T - \mathbf{X}_E \tag{15}$$

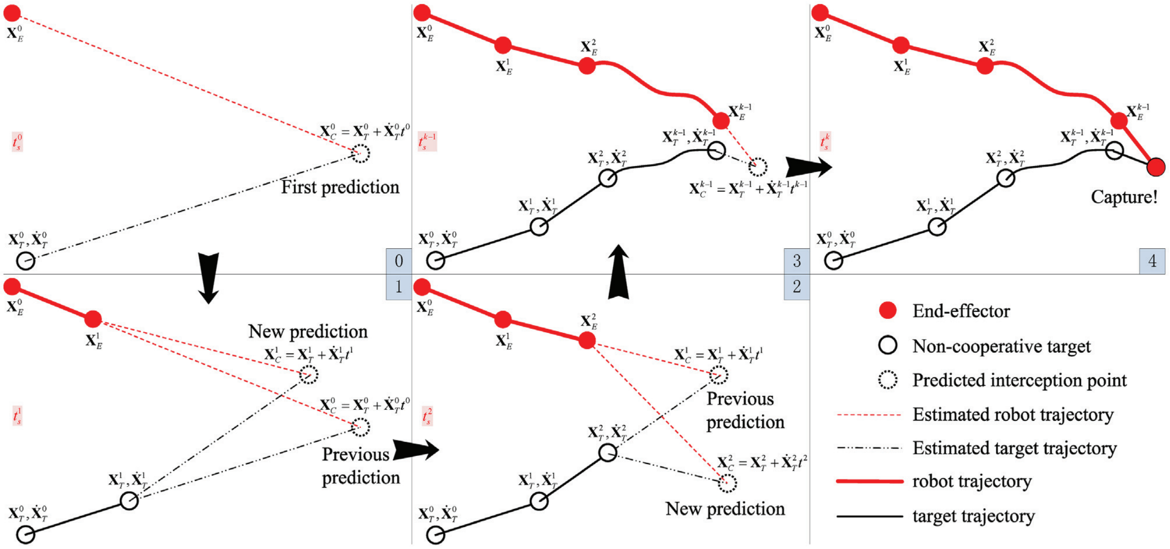

$\mathbf{X}_E^0$
t_s^0
$\mathbf{X}_C^0 = \mathbf{X}_T^0 + \dot{\mathbf{X}}_T^0 t^0$
First prediction
$\mathbf{X}_T^0, \dot{\mathbf{X}}_T^0$
0
$\mathbf{X}_E^0$
$\mathbf{X}_E^1$
$\mathbf{X}_E^2$
$\mathbf{X}_E^{k-1}$
t_s^{k-1}
$\mathbf{X}_T^{k-1}, \dot{\mathbf{X}}_T^{k-1}$
$\mathbf{X}_T^2, \dot{\mathbf{X}}_T^2$
$\mathbf{X}_C^{k-1} = \mathbf{X}_T^{k-1} + \dot{\mathbf{X}}_T^{k-1} t^{k-1}$
$\mathbf{X}_T^1, \dot{\mathbf{X}}_T^1$
$\mathbf{X}_T^0, \dot{\mathbf{X}}_T^0$
3
$\mathbf{X}_E^0$
$\mathbf{X}_E^1$
$\mathbf{X}_E^2$
$\mathbf{X}_E^{k-1}$
t_s^k
$\mathbf{X}_T^{k-1}, \dot{\mathbf{X}}_T^{k-1}$
$\mathbf{X}_T^2, \dot{\mathbf{X}}_T^2$
Capture!
$\mathbf{X}_T^1, \dot{\mathbf{X}}_T^1$
$\mathbf{X}_T^0, \dot{\mathbf{X}}_T^0$
4
1
$\mathbf{X}_E^0$
$\mathbf{X}_E^1$
New prediction
$\mathbf{X}_C^1 = \mathbf{X}_T^1 + \dot{\mathbf{X}}_T^1 t^1$
$\mathbf{X}_C^0 = \mathbf{X}_T^0 + \dot{\mathbf{X}}_T^0 t^0$
t_s^1
Previous prediction
$\mathbf{X}_T^1, \dot{\mathbf{X}}_T^1$
$\mathbf{X}_T^0, \dot{\mathbf{X}}_T^0$
2
$\mathbf{X}_E^0$
$\mathbf{X}_E^1$
$\mathbf{X}_E^2$
$\mathbf{X}_C^1 = \mathbf{X}_T^1 + \dot{\mathbf{X}}_T^1 t^1$
Previous prediction
t_s^2
$\mathbf{X}_T^2, \dot{\mathbf{X}}_T^2$
$\mathbf{X}_C^2 = \mathbf{X}_T^2 + \dot{\mathbf{X}}_T^2 t^2$
$\mathbf{X}_T^1, \dot{\mathbf{X}}_T^1$
$\mathbf{X}_T^0, \dot{\mathbf{X}}_T^0$
New prediction
● End-effector
○ Non-cooperative target
◌ Predicted interception point
Estimated robot trajectory
Estimated target trajectory
robot trajectory
target trajectory

Figure 1. An illustration of the autonomous capture process.

$$\dot{\mathbf{E}} = \dot{\mathbf{X}}_T - \dot{\mathbf{X}}_E \tag{16}$$

Substituting Equations (15) and (16) into Equations (8) and (9) yields

$$\dot{\mathbf{E}} = -t^{-1}\mathbf{E} \tag{17}$$

Since the time t is always positive, the tracking error is guaranteed to converge to zero and the proposed control law is stable.

After the end effector moves one step towards the interception point, the procedure defined by Equations (8)–(14) is repeated for the next step until the capture requirement is achieved. As the end effector approaches the target, the error between the dynamically predicted and real interception points will be diminished. At the same time, the velocity of the end effector approaches to the velocity of the target. It is worth pointing out that the predicted interception point may change if the target is not stationary. As a result, the resulting trajectory of the end effector towards the target by the above approach is not necessary straight although the incremental joint control input is estimated based on a straight-line approach. Figure 1 illustrates a typical autonomous capture process by the proposed control strategy. After the first prediction of the interception point, the end effector moves by an increment $\Delta\mathbf{X}_E = \mathbf{J}\Delta\boldsymbol{\Theta}$ towards the interception point by one step. The kinematic state of the target is then updated by the vision system before the second prediction is performed. A capture will be made once the estimated minimum interception time $t_{\min}$ is smaller than the sample time t_s of control system. The iterative algorithm of the incremental control process is summarized in Table 1.

Table 1. The iterative algorithm of the incremental position-level control strategy.

```
while (true)
{
Obtain joint variables of the robotic manipulator $\boldsymbol{\Theta}$;
 Evaluate transformation matrix $\mathbf{T}$, Jacobian matrix $\mathbf{J}$ and position
   of the end effector $\mathbf{X}_E$ at $\boldsymbol{\Theta}$;
Calculate $\dot{\mathbf{T}}$ and $\mathbf{J}^{\dagger}$;
Obtain estimates of target's position and velocity $\mathbf{X}_T$, $\dot{\mathbf{X}}_T$ [7];
Apply the velocity limit $\dot{\boldsymbol{\Theta}}_{\max}$ to obtain $t_{\min}$ by Equation (13);
Evaluate Equation (11) at $t_{\min}$ to obtain $\dot{\boldsymbol{\Theta}}$;
if ($t_{\min} < t_s$)
Capture and break;
Else
 Calculate the position-level relative control input $\Delta\boldsymbol{\Theta} = \dot{\boldsymbol{\Theta}} t_s$ and
   apply to the robot;
}
```

3. Hardware-in-the-loop simulation results and discussion

The proposed incremental visual servo control law is examined by a hardware-in-the-loop simulation, where a three-link robotic manipulator was simulated and an eye-to-hand camera fed the simulation with the target position and velocity. As shown in Figure 2, the global frame is fixed in the inertial space with the Z-axis aligned with the axis of the first link (l_0). The XY-plane of the global frame is offset by l_0 from the fixed base. Initially, the second (l_1) and third (l_2) links are within the YZ-plane. The home position of the robotic manipulator in the global frame is set to $\theta_1 = 0^\circ$, $\theta_2 = 90^\circ$ and $\theta_3 = 0^\circ$. The link length and joint velocity limits of the manipulator are listed in Table 2. Finally, the update rate in the hardware-in-the-loop simulation is assumed the same as the camera measurement process time, which is running at a rate of approximately 30 Hz.

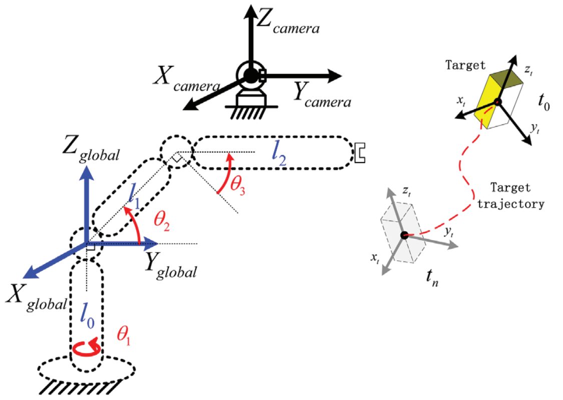


Figure 2. Schematic of the hardware-in-the-loop validating system.

Table 2. Physical properties of the robotic manipulator.

Parameter	Value
l_0, m	0.1778
l_1, m	0.4540
l_2, m	0.4445
$\dot{\theta}_{\max 1}$, rad/s	10.304
$\dot{\theta}_{\max 2}$, rad/s	0.412
$\dot{\theta}_{\max 3}$, rad/s	0.412

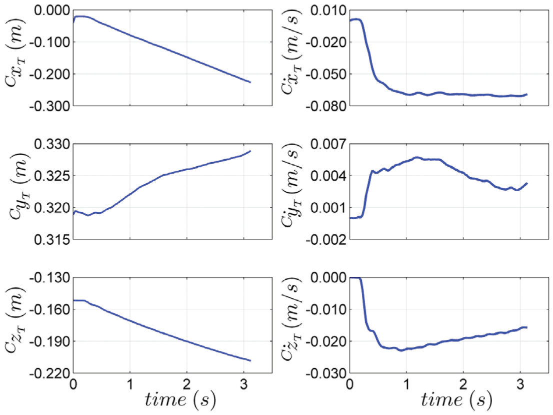


Figure 3. Measured position (a) and velocity (b) of the non-cooperative target in the camera frame.

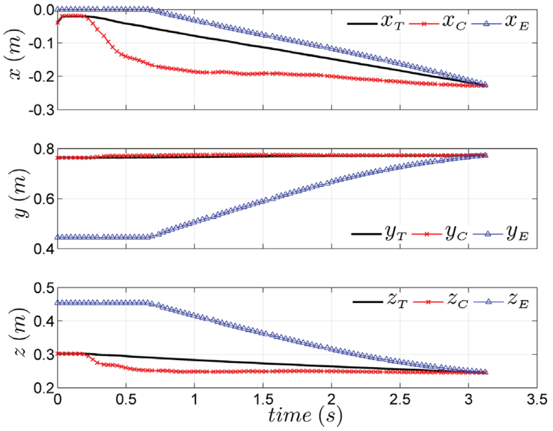


Figure 4. Position history of the target, the end effector and the predicted interception point.

By defining $c_i = \cos(\theta_i)$ and $s_i = \sin(\theta_i)$, the Jacobian matrix of the robotic manipulator becomes

$$\mathbf{J} = \begin{bmatrix} -l_1c_1c_2 - l_2c_1s_2c_3 - l_2c_1c_2s_3 & l_1s_1s_2 - l_2s_1c_2c_3 + l_2s_1s_2s_3 & -l_2c_2c_3 + l_2s_2s_3 \\ -l_1s_1c_2 - l_2s_1s_2c_3 - l_2s_1c_2s_3 & -l_1c_1s_2 + l_2c_1c_2c_3 - l_2c_1s_2s_3 & l_2c_2c_3 - l_2s_2s_3 \\ 0 & l_1c_2 + l_2s_2c_3 + l_2c_2s_3 & l_2s_2c_3 + l_2c_2s_3 \end{bmatrix} \tag{18}$$

The position of the end effector is then derived by the forward kinematics

$$\mathbf{X}_E = \begin{Bmatrix} x_E \\ y_E \\ z_E \end{Bmatrix} = \begin{Bmatrix} -l_1s_1c_2 - l_2s_1s_2c_3 - l_2s_1c_2s_3 \\ l_1c_1c_2 + l_2c_1s_2c_3 + l_2c_1c_2s_3 \\ l_1s_2 - l_2c_2c_3 + l_2s_2s_3 \end{Bmatrix} \tag{19}$$

The non-cooperative target in the experiment is moved independently in the 3D workspace of the robotic manipulator by a stepping motor at the estimated speed of 6 cm/s. Initially, the target is placed around 35 cm away from the end effector. Figure 3 shows the position and velocity of the target with respect to the camera frame measured by the vision system, denoted by ${}^{C}\mathbf{X}_T = \left({}^{C}x_T, {}^{C}y_T, {}^{C}z_T\right)^T$ and ${}^{C}\dot{\mathbf{X}}_T = \left({}^{C}\dot{x}_T, {}^{C}\dot{y}_T, {}^{C}\dot{z}_T\right)^T$, respectively. They are transferred to the global frame by Equations (2) and (3) and fed into the simulation.

In the simulation, the robotic manipulator is at the home position initially. The target is set to move roughly 0.65 s ahead of the end effector. Figure 4 shows the time histories of the measured positions of the target $\mathbf{X}_T = (x_T, y_T, z_T)^T$, the simulated position of the end effector $\mathbf{X}_E = (x_E, y_E, z_E)^T$ and the predicted interception point $\mathbf{X}_C = (x_C, y_C, z_C)^T$ in the global frame. The corresponding tracking error $\mathbf{E}$ is shown in Figure 5. The position of the predicted interception point, as shown in Figure 4, varied rapidly at the early stage. This is evident that the proposed control law works effectively and efficiently to predict the interception point. As the end effector approaches the target over time, the prediction will be more and more accurate. It is noted that the control error in Figure 5 approaches zero asymptotically, which

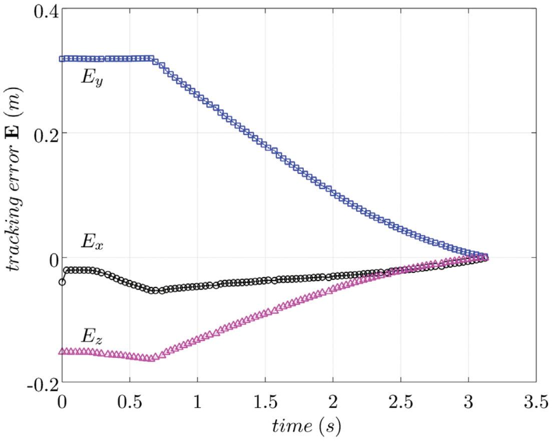


Figure 5. Time history of the tracking error.

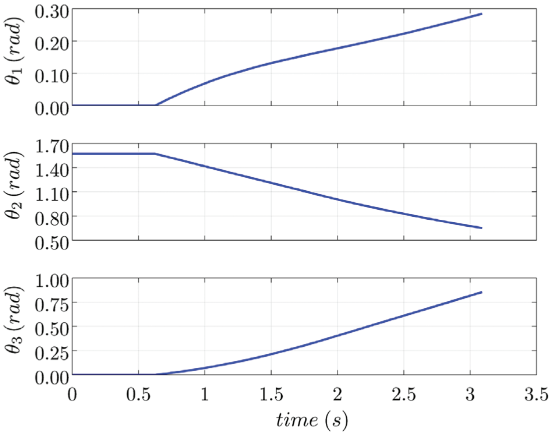


Figure 6. Joint angles of the robotic manipulator.

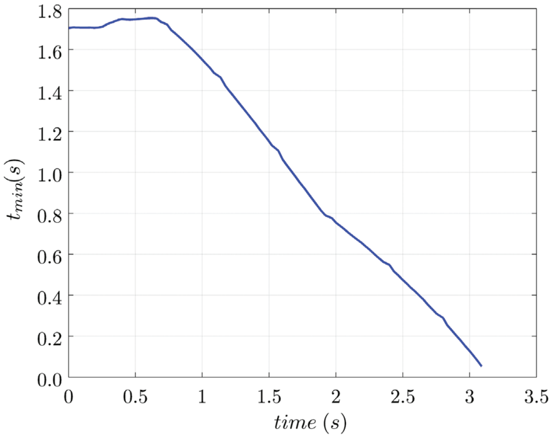


Figure 7. Time history of the estimated minimum interception time.

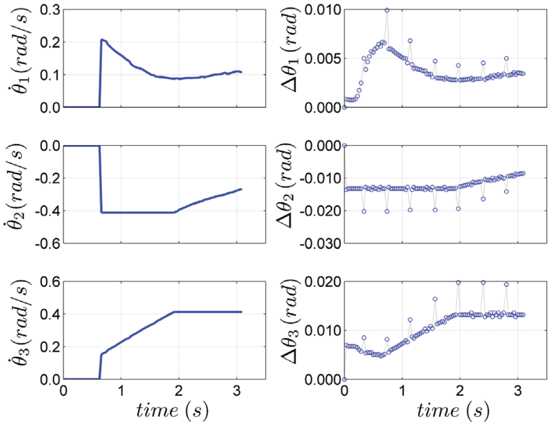


Figure 8. (a) Calculated joint velocity and (b) Incremental control input of the robotic manipulator.

indicates the visual servo control law is stable. A capture was made at about 3.1 s once the end effector is close enough to the target. The corresponding joint angles of the robotic manipulator during the autonomous capture process are shown in Figure 6. The motion profiles of the end effector in Figure 4, the tracking error in Figure 5 and the variations of the joint variables in Figure 6 are quite smooth, which are generally preferred in the operation of robot.

Next, the estimated minimum interception time in the tracking process is kept updated and is shown in Figure 7. The estimated time increase at the beginning is due to the target moves first while the end effector remained at the home position for roughly 0.65 s. Once the control of the robotic manipulator is activated, the estimated minimum interception time is reduced asymptotically at a roughly constant rate as the end effector approached the target. The decrease in minimum interception time in Figure 7 indicates the end effector is approaching the target effectively. Since the control law is designed based on the piecewise uniform linear motion of the target within each sample interval, the decrease in Figure 7 for a non-linear target motion, as shown in Figure 3, also indicates the proposed approach is tolerant to modelling errors.

The control input of the robotic manipulator in the current case is the relative or incremental joint angles that are calculated from the joint velocities. The time histories of the joint velocities and corresponding incremental control input for joint angles are depicted in Figure 8. It shows that the joint velocities $\dot{\theta}_2$ and $\dot{\theta}_3$ reach the velocity limits during the approaching process, see Figure 8(a), which indicates the newly proposed controller effectively enforces the constraints of joint velocity limit. It is worthy to note that there are outliers in the incremental control input periodically as shown in Figure 8(b). This is due to

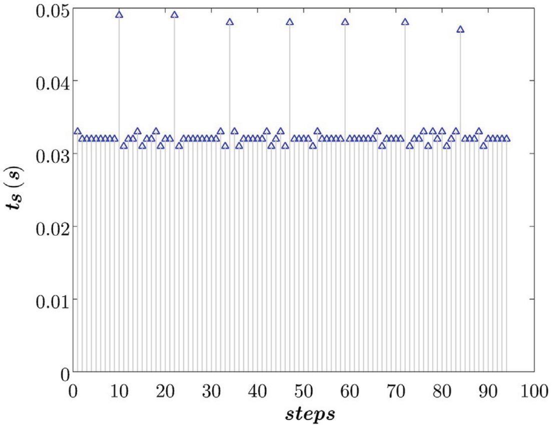


Figure 9. Irregular sampling times of a digital camera with 30 Hz sampling rate.

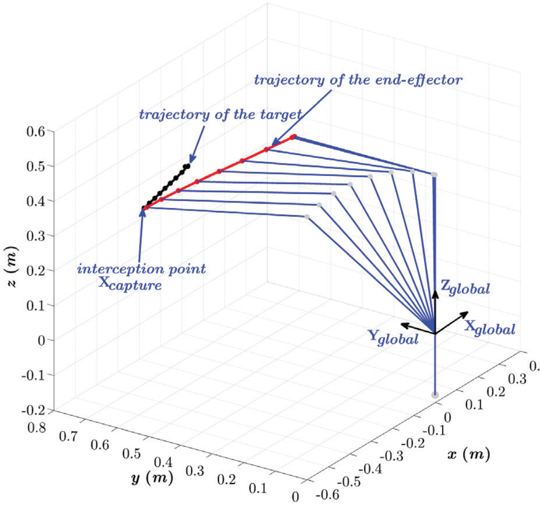


Figure 10. Schematic of the autonomous capture trajectory in 3D space.

the irregular sampling rate of camera as shown in Figure 9. In the real-time implementation, these outliers can be easily filtered out.

Figure 10 illustrates the 3D autonomous capture process at every 10 sample points conducted by the proposed control strategy defined in Table 1. The simulation results demonstrated the effectiveness and efficiency of the proposed incremental visual servo control strategy at position-level.

4. Conclusion

This work proposed an incremental visual servo control law for the robotic manipulator at the position-level in joint space to capture a non-cooperative target autonomously. The main contribution is to avoid the multiple solutions in the inverse kinematics by solving the increments of joint position from the current position based on visual feedback. Furthermore, the interception point, predicted dynamically by the feedback of current position and the target position, is used as the desired position of the end effector. The newly proposed control law is validated by a hardware-in-the-loop simulation, where the kinematic state of the non-cooperative target is measured by an eye-to-hand vision system and the motion of the robotic manipulator is simulated. The simulation results show the end effector moves effectively towards the predicted interception point by the incremental angular control input at joints.

Funding

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) [grant number 341917-2013].

Notes on contributors

Gangqi Dong received his BSc and MSc degrees in guidance, navigation and control from Northwestern Polytechnical University, Xi'an, Shaanxi, China, in 2005 and 2008, respectively. He was with Xi'an Institute of Microelectronic Technology as a computer hardware engineer from 2008 to 2010. Then he joined Huawei Technologies Co., Ltd. as a hardware engineer of communication from 2010 to 2011. He is currently a PhD candidate in space engineering at York University, Toronto, ON, Canada. His research interests include development of mechatronic systems and visual servo robotic systems.

Zheng H. Zhu received his BEng., MEng. and PhD degrees in mechanics from Shanghai Jiao Tong University located in Shanghai, China. He also received his MASc. in robot control from University of Waterloo and PhD in mechanical engineering from University of Toronto all located in Ontario, Canada. From 1993 to 1995, he worked as a research associate in the Department of Mechanical and Industrial Engineering, University of Toronto. From 1995 to 2006, he was a senior engineer with the Curtiss-Wright – Indal Technologies located in Mississauga, Ontario, Canada. Since 2006, he has been a professor with the Department of Earth and Space Science and Engineering, York University in Toronto, Canada. He is the author of more than 180 articles. His research interests include on-orbit service robot, dynamics and control of tethered space system. He is the editor-in-chief of the International Journal of Space Science and Engineering. Zhu is the fellow of Engineering Institute of Canada, associate fellow of AIAA, fellow of CSME, senior member of IEEE, member of ASME and licensed professional engineer in Ontario.

ORCiD

Gangqi Dong http://orcid.org/0000-0002-5297-2755
Zheng H. Zhu http://orcid.org/0000-0002-0149-0473

References

[1] Yoshida K. Achievements in space robotics. IEEE Rob. Autom. Mag. 2009;16:20–28.

[2] Bac CW, van Henten EJ, Hemming J, et al. Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. J. Field Rob. 2014;31:888–911.

[3] Smith JA, Jivraj J, Wong R, et al. 30 years of neurosurgical robots: review and trends for manipulators and associated navigational systems. Ann. Biomed. Eng. 2015;1–11.

[4] Flores-Abad A, Ma O, Pham K, et al. A review of space robotics technologies for on-orbit servicing. Prog. Aerosp. Sci. 2014;68:1–26.

[5] Yoshida K, Nakanishi H, Ueno H, et al. Dynamics, control and impedance matching for robotic capture of a non-cooperative satellite. Adv. Rob. 2004;18:175–198.

[6] Larouche BP, Zhu ZH. Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control. Auton. Robots. 2014;37:157–167.

[7] Dong G, Zhu ZH. Position-based visual servo control of autonomous robotic manipulators. Acta Astronaut. 2015;115:291–302.

[8] Dong G, Zhu ZH. Autonomous robotic capture of non-cooperative target by adaptive extended Kalman filter based visual servo. Acta Astronaut. 2016;122:209–218.

[9] Gui J, Gu D, Wang S, et al. A review of visual inertial odometry from filtering and optimisation perspectives. Adv. Rob. 2015;29:1289–1301.

[10] Ma L, Hovakimyan N. Vision-based cyclic pursuit for cooperative target tracking. J. Guidance Control Dyn. 2013;36:617–622.

[11] Mehta SS, Burks TF. Vision-based control of robotic manipulator for citrus harvesting. Comput. Electron. Agric. 2014;102:146–158.

[12] Janabi-Sharifi F, Marey M. A Kalman-filter-based method for pose estimation in visual servoing. IEEE Trans. Rob. 2010;26:939–947.

[13] Wang C, Lin CY, Tomizuka M. Statistical learning algorithms to compensate slow visual feedback for industrial robots. J. Dyn. Syst.-T Asme. 2015;137:031011.

[14] Chaumette F, Hutchinson S. Visual servo control. I. Basic approaches. IEEE Rob. Autom. Mag. 2006;13:82–90.

[15] Chaumette F, Hutchinson S. Visual servo control. II. Advanced approaches [Tutorial]. IEEE Rob. Autom. Mag. 2007;14:109–118.

[16] Siciliano B. Kinematic control of redundant robot manipulators: a tutorial. J. Intell. Rob. Syst. 1990;3:201–212.

[17] Jareanpon C, Maneewongvatana S. Shape control of a hyper-redundant arm for planar object manipulation. Adv. Rob. 2011;25:1159–1181.

[18] Rolf M, Neumann K, Queißer JF, et al. A multi-level control architecture for the bionic handling assistant. Adv. Rob. 2015;29:847–859.

[19] Cocuzza S, Pretto I, Debei S. Least-squares-based reaction control of space manipulators. J. Guidance Control Dyn. 2012;35:976–986.

[20] Wang Y, Luo J, Li W, et al. Inverse-free D1G1 solution to acceleration-level inverse kinematics of redundant robot manipulators. Progress in IEEE Informatics and Computing (PIC). 2014;280–284.

[21] Chiaverini S. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. IEEE Trans. Rob. Autom. 1997;13:398–410.

[22] Wang C, Lin CY, Tomizuka M. Design of kinematic controller for real-time vision guided robot manipulators. In: Robotics and Automation (ICRA), 2014 IEEE International Conference; 2014 May 31. p. 4141–4146. IEEE.

[23] Dong G, Zhu ZH. Incremental inverse kinematics based vision servo for autonomous robotic capture of non-cooperative space debris. Adv. Space Res. 2016;57:1508–1514.

Reference Paper E

Visual Servo Kinematic Control for Autonomous Robotic Capture of Non-cooperative Target[3]

Gangqi Dong and Z. H. Zhu

Abstract

This paper presents a visual servo incremental kinematic control scheme for the robotic manipulator with the eye-in-hand configuration to perform autonomous capture of a non-cooperative target. An integrated algorithm of the photogrammetry and adaptive extended Kalman filter is proposed to improve the efficiency and accuracy of the target motion estimation in real time. Based on the vision feedback, the end-effector moves directly towards the estimated position of the target at the next time instant incrementally, which effectively avoid the kinematic singularity and multiple solutions of inverse kinematics in the joint space. Validating experiments are performed on a custom built robotic manipulator with an eye-in-hand configuration. The experimental results demonstrate the effectiveness and robustness of the proposed control scheme.

Keywords Kinematic control・Visual servo・Autonomous capture・Robotic

[3] Submitted to **Autonomous Robots** on April 02, 2016.

manipulator・Non-cooperative target・Eye-in-hand

1 Introduction

Robotic manipulators have been extensively employed in space missions to perform tasks that are dangerous, complex, and even impossible to be conducted by human beings, seen in Yoshida (2009), Bac et al. (2014) and Smith et al. (2015). For instance, missions like on-orbit-servicing and active debris removal require capturing non-cooperative targets by robotics with high accuracy and autonomy (Hsiao et al., 2011; Ignakov et al., 2012; Larouche and Zhu, 2014). The recent study by Flores-Abad et al. (2014) shows that autonomous robotic capture, especially when involving the non-cooperative targets, is still very challenging, although numerous enabling techniques have been proposed and several on-orbit-servicing missions were successfully performed with the participation of human beings. Furthermore, a preliminary concept design of guidance, navigation and control architecture had been proposed by Jankovic et al. (2015) for a safe and fuel-efficient robotic capture of a non-cooperative target in active debris removal missions, in which the capture process was divided into three phases: the far range rendezvous, the close range rendezvous and the final capture. In this paper, we focus on the visual servo control in the final autonomous capture of a non-cooperative target by a robotic manipulator.

Two challenges arise sequentially in the autonomous capture of a non-cooperative target by a robotic manipulator. First, the robotic manipulator has to promptly know where the target is, which means the position and motion of the target has to be estimated in real time. Because of the non-contact nature, vision systems are extensively adopted in robotic control for monitoring, detecting and tracking purpose, seen in Ghadiok et al. (2012). Considerable efforts have been devoted to the development of vision-based position and motion estimation methods. Generally, existing methods can be divided into four categories, such as geometric, learning-based optimization, offline and filtering-based methods. Geometric methods are prone to image noises since they are based on the current measurement only. Learning-based optimization methods require sufficient knowledge of the target in advance, which is not available if the target is non-cooperative. The offline methods perform estimation afterwards and is not suitable for real-time applications. Therefore, filtering-based estimation methods, especially the Kalman filter based methods, are generally favored in the literature as they do not require priori knowledge of the targets, seen in Janabi-Sharifi and Marey (2010); Aghili (2012); Chen (2012); Sabatini et al. (2013) and Dong and Zhu (2015, 2016). Depending on the camera configuration, the vision systems are classified into two types: the eye-in-hand and the eye-to-hand. As the name implies, the camera is mounted on the robotic manipulator in the eye-in-hand configuration while detached from the manipulator and fixed in workspace in the eye-to-hand configuration. Consequently, the eye-to-hand camera monitors the whole workspace to provide global yet less accurate estimates of target's position and motion in a global frame. In contrast, the eye-in-hand camera is usually mounted close to the end-effector and provides a close and more accurate scene of the target in a local/camera frame. The accuracy of the estimates of position and motion increases as the end-effector approaches the target. Accordingly, the eye-in-hand configuration is adopted in the current study. To avoid the loss of target tracking caused by the moving target and the motion/oscillation of the eye-in-hand camera, an integrated estimation algorithm of photogrammetry and adaptive extended Kalman filter (AEKF) is proposed to estimate the position and motion of the non-cooperative target in real time.

Once the position and motion of the target are available to the robotic manipulator, the second challenge is how to determine the trajectory of the robotic manipulator in order to make a capture autonomously. In space applications, the robotic manipulators are generally driven by stepping motors at the joints slowly with angular position as control input. It is reasonable and acceptable to design the control strategy of space robotic manipulators in the current study based on the kinematic approach. Ideally, the desired position of the end-effector should be the interception point between trajectories of the target and the end-effector. However, the determination of the interception point is challenging in case of non-cooperative targets where the target trajectory is unknown in advance. In order to address this challenge, a new target-aiming kinematic control strategy is proposed in this work with vision input and the dynamic update of the trajectory of the robotic manipulator at each sampling instant. The incremental nature of the approach effectively avoids the multiple solutions in the inverse kinematics of robotic manipulators. The approach is approved robust in tracking the motion of the non-cooperative target and stable in approaching the target. It is worth noting that the current approach is derived based on a fixed-base manipulator to prove the concept of the current approach. Additional efforts will be required to apply the current work to free floating manipulators in space by replacing the Jacobian matrix in the current kinematic approach with a generalized Jacobian matrix (Umetani and Yoshida, 1989), or mapping the kinematics of a free floating manipulator to a fixed-base manipulator (Vafa and Dubowsky, 1987; Liang, et al, 1998). The newly proposed approach is applied to a robotic manipulator with an eye-in-hand configuration and validated experimentally using a custom built robotic manipulator. The experimental results successfully demonstrated the effectiveness and robustness of the proposed motion algorithm of non-cooperative targets as well as the target-aiming kinematic control strategy for the autonomous robotic capture.

2 Vision-Based Target Motion Estimation

Consider a non-cooperative target and a vision-based robot system in space. Without loss of generality, assume the global frame, denoted by $\mathcal{G}$, is fixed in the inertia space; the camera frame, denoted by $\mathcal{C}$, is attached to the center of the image plane of camera and the target frame, denoted by $\mathcal{T}$, is fixed to the target body, respectively. Further assume the position of a feature point on the target is known with respect to (w.r.t.) $\mathcal{T}$ and is denoted by $\{x_T, y_T, z_T\}^T$ in the target frame. Then, the position of this feature point w.r.t. $\mathcal{C}$, denoted by $\{x_C, y_C, z_C\}^T$ in the camera frame, can be derived by an augmented homogeneous transformation from $\mathcal{T}$ to $\mathcal{C}$, such that

$$\begin{Bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{Bmatrix} = \begin{bmatrix} & & & x_{To} \\ & \mathbf{R}_{TC} & & y_{To} \\ & & & z_{To} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{Bmatrix} \tag{1}$$

where $\{x_{To}, y_{To}, z_{To}\}^T$ denotes the origin position of $\mathcal{T}$ w.r.t. $\mathcal{C}$ in the target frame and $\mathbf{R}_{TC}$ stands for the transformation or rotational matrix from $\mathcal{T}$ to $\mathcal{C}$. The transformation matrix is formed by the trigonometric functions of the Euler angles between axes of $\mathcal{T}$ and $\mathcal{C}$, denoted by $\{\theta_x, \theta_y, \theta_z\}^T$.

According to Eq. (1), $\{x_C, y_C, z_C\}^T$ can be expressed as the functions of $\{x_{To}, y_{To}, z_{To}\}^T$ and $\{\theta_x, \theta_y, \theta_z\}^T$. Note that $\{x_{To}, y_{To}, z_{To}\}^T$ and $\{\theta_x, \theta_y, \theta_z\}^T$ are the unknown position and orientation of $\mathcal{T}$ w.r.t. $\mathcal{C}$ and will be determined by the photogrammetry as follows. Let a feature point on the target is projected onto the image plane by a pinhole camera, such that

$$\begin{Bmatrix} x_m \\ z_m \end{Bmatrix} = -\frac{f}{y_C - f} \begin{Bmatrix} x_C \\ z_C \end{Bmatrix} \tag{2}$$

where f is the focal length of the camera and $\{x_m, z_m\}^T$ is the projected coordinates of the feature point on the image plane and can be obtained from the image processing. Here, the y-axis of $\mathcal{C}$ is assumed to be perpendicular with the image plane and pointing to the target.

Rearranging Eq. (2) yields two independent equations with six unknowns for one feature point,

$$\begin{cases} x_m y_C - x_m f + x_C f = 0 \\ z_m y_C - z_m f + z_C f = 0 \end{cases} \tag{3}$$

Theoretically, a minimum of three distinguished feature points are required to solve for six unknowns but ambiguous solutions may occur due to the periodic solutions of trigonometric functions. To avoid the ambiguity, at least four feature points are usually adopted to increase the robustness of solution (Larouche and Zhu, 2014). The resulting equations are highly nonlinear and are generally solved iteratively by the least square approach with an initial guess. Noting this process is solely based on the current measurement, the memoryless characteristics of the photogrammetry leads to the solution sensitive to the measurement noises. Furthermore, the computational cost may subject to the errors between initial guess and the real solution because the initial guess is made arbitrarily, leading to the large variation of the system sampling time. The latter is not desirable in real-time applications. In addition, similar to the limitation of other geometric approach, the photogrammetry does not solve for the target motion directly. In order to address these challenges, an integrated algorithm of photogrammetry and AEKF is employed to estimate the kinematic states of the target timely.

The state variable vector of a target is defined based on its kinematic variables, such as,

$$\mathbf{X}=\left\{x_{To},\dot{x}_{To},y_{To},\dot{y}_{To},z_{To},\dot{z}_{To},\theta_x,\dot{\theta}_x,\theta_y,\dot{\theta}_y,\theta_z,\dot{\theta}_z\right\}^T \tag{4}$$

The overhead dot denotes the time derivatives of the corresponding variables. By assuming the target moving at a constant speed within a sufficient small sampling time interval t_s and the acceleration vector, denoted by $\boldsymbol{\omega}=\left\{\ddot{x}_{To},\ddot{y}_{To},\ddot{z}_{To},\ddot{\theta}_x,\ddot{\theta}_y,\ddot{\theta}_z\right\}^T$, as the process noise for simplicity, the system model is defined as

$$\mathbf{X}_k=\mathbf{A}\mathbf{X}_{k-1}+\mathbf{B}\boldsymbol{\omega}_{k-1} \tag{5}$$

The subscripts k and $k-1$ in Eq. (5) indicate the current and previous sampling time steps. The matrices $\mathbf{A}$ and $\mathbf{B}$ can be written as

$$\mathbf{A}=diag\begin{bmatrix}\mathbf{a} & \mathbf{a} & \mathbf{a} & \mathbf{a} & \mathbf{a} & \mathbf{a}\end{bmatrix},\quad \mathbf{a}=\begin{bmatrix}1 & t_s\\ 0 & 1\end{bmatrix} \tag{6}$$

$$\mathbf{B}=diag\begin{bmatrix}\mathbf{b} & \mathbf{b} & \mathbf{b} & \mathbf{b} & \mathbf{b} & \mathbf{b}\end{bmatrix},\quad \mathbf{b}=\begin{Bmatrix}t_s^2/2\\ t_s\end{Bmatrix} \tag{7}$$

In order to apply the Kalman filtering method, the process noise $\boldsymbol{\omega}$ is assumed to obey the Gaussian distribution with normal mean vector $\mathbf{q}$ and covariance matrix $\mathbf{Q}$, such that

$$\boldsymbol{\omega}\sim\mathcal{N}(\mathbf{q},\mathbf{Q}) \tag{8}$$

Because the pose of the target are detected by the camera, based on the pinhole camera model described by Eq. (2), the measurement model is defined as

$$\mathbf{Z}_k=\mathbf{h}(\mathbf{X}_k)+\boldsymbol{\mu}_k \tag{9}$$

where $\mathbf{Z}$ is the position vector of the feature points in image plane, $\boldsymbol{\mu}$ stands for the measurement noise of the camera, and $\mathbf{h}(\mathbf{X})$ is defined by

$$\mathbf{h}(\mathbf{X})=-\frac{f}{y_C-f}\begin{Bmatrix}x_C\\ z_C\end{Bmatrix} \tag{10}$$

Note that the measurement noise of the camera depends on operation conditions, such as illumination, environmental temperature, jittering of pixels, etc. It is reasonable to assume that the measurement noise obeys the zero mean Gaussian distribution with covariance matrix $\mathbf{R}$, such that,

$$\boldsymbol{\mu} \sim \mathcal{N}\left(\mathbf{0},\mathbf{R}\right) \tag{11}$$

Generally, the process and measurement noises are time-varying and it is desirable to adaptively update them for high accuracy. However, it is difficult to distinguish them from each other in reality. As a result, adaptively updating the distributions of both process and measurement noises may not be robust according to Song and Han (2008). Considering the fact that the measurement noises are characteristics of the measurement system and are independent on the target, it is reasonable to assume the process and measurement noises are independent. Thus, the covariance matrix of the measurement noises $\mathbf{R}$ can be determined experimentally in advance, while only $\mathbf{q}$ and $\mathbf{Q}$ are adaptively updated for the process noises in the current work.

However, challenge arises for the determination of $\mathbf{q}$ and $\mathbf{Q}$ due to the non-cooperative nature of the target as well as the eye-in-hand configuration of the vision system where the camera is affected by the motion of the end-effector. For given initial conditions, the state variable vector $\mathbf{X}$ and its corresponding covariance matrix $\mathbf{P}$ in the next step can be predicted by following the typical process of Kalman filtering algorithm, such that,

$$\mathbf{X}_{k|k-1} = \mathbf{A}\mathbf{X}_{k-1|k-1} + \mathbf{B}\mathbf{q}_{k-1} \tag{12}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^{T} + \mathbf{B}\mathbf{Q}_{k-1}\mathbf{B}^{T} \tag{13}$$

Then, the Kalman gain is derived by

$$\mathbf{K}_{g} = \mathbf{P}_{k|k-1}\mathbf{H}_{k}^{T}\left(\mathbf{H}_{k}\mathbf{P}_{k|k-1}\mathbf{H}_{k}^{T} + \mathbf{R}\right)^{-1} \tag{14}$$

where $\mathbf{H}$ denotes the Jacobian matrix of the measurement model, which is defined by

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}(\mathbf{X})}{\partial \mathbf{X}}\right|_{\mathbf{X}=\mathbf{X}_{k|k-1}} \tag{15}$$

Once the measurement vector $\mathbf{Z}$ is obtained from the image processing, the state variable vector and its corresponding covariance matrix in the next step are updated by

$$\mathbf{X}_{k|k} = \mathbf{X}_{k|k-1} + \mathbf{K}_g\left(\mathbf{Z}_k - \mathbf{h}\left(\mathbf{X}_{k|k-1}\right)\right) \tag{16}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_g \mathbf{H}_k \mathbf{P}_{k|k-1} \tag{17}$$

In order to adaptively update the normal mean and covariance matrix of the process noise, an intuitive approximation of $\mathbf{q}$ at the jth step is defined in accordance with Myers and Tapley (1976), such as,

$$\hat{\mathbf{q}}_j = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\left(\mathbf{X}_{j|j} - \mathbf{A}\mathbf{X}_{j-1|j-1}\right) \tag{18}$$

Accordingly, the unbiased estimates of $\mathbf{q}$ and $\mathbf{Q}$ at the time k can be evaluated under the assumption that the process noises are independent and uniformly distributed over N time steps, such that,

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \frac{1}{N}\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N}\right) \tag{19}$$

$$\mathbf{Q}_k = \mathbf{Q}_{k-1} + \frac{1}{N-1}\left(\begin{array}{c}\left(\hat{\mathbf{q}}_k - \mathbf{q}_k\right)\left(\hat{\mathbf{q}}_k - \mathbf{q}_k\right)^T - \left(\hat{\mathbf{q}}_{k-N} - \mathbf{q}_k\right)\left(\hat{\mathbf{q}}_{k-N} - \mathbf{q}_k\right)^T + \\ \frac{1}{N}\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N}\right)\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N}\right)^T + \frac{N-1}{N}\left(\mathbf{\Lambda}_{k-N} - \mathbf{\Lambda}_k\right)\end{array}\right) \tag{20}$$

$$\mathbf{\Lambda}_j = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\left(\mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T - \mathbf{P}_{k|k}\right)\mathbf{B}\left(\mathbf{B}^T\mathbf{B}\right)^{-1} \tag{21}$$

The unbiased estimates of $\mathbf{q}$ and $\mathbf{Q}$ at the time k as described by Eqs. (19), (20) and (21) are the adaptive distribution updates of the process noises. Since the initial state variables of the non-cooperative target is unknown, a poor initial guess would lead to a long convergence time of the AEKF, which is not desired in real-time application. Thus, we adopted the estimation results of photogrammetry to initialize the AEKF to accelerate the convergence of the

AEKF. Eqs. (12) - (21) are iteratively performed and the estimate of the state variable vector is obtained by Eq. (16) during each iteration.

It should be noted that the estimated position and motion of the target are w.r.t. $\mathcal{C}$. For a concise description, let ${}^{C}\mathbf{X}_T, {}^{C}\dot{\mathbf{X}}_T$ and $\mathbf{X}_T, \dot{\mathbf{X}}_T$ denote the estimated position and motion of the target w.r.t. $\mathcal{C}$ and $\mathcal{G}$, respectively. According to the geometric configuration of the camera, the transformation from $\mathcal{C}$ to $\mathcal{G}$ can be easily accomplished by the homogeneous transformation matrix, denoted by $\mathbf{T}$, such that

$$\begin{Bmatrix} \mathbf{X}_T \\ 1 \end{Bmatrix} = \mathbf{T} \begin{Bmatrix} {}^{C}\mathbf{X}_T \\ 1 \end{Bmatrix} \tag{22}$$

$$\begin{Bmatrix} \dot{\mathbf{X}}_T \\ 0 \end{Bmatrix} = \mathbf{T} \begin{Bmatrix} {}^{C}\dot{\mathbf{X}}_T \\ 0 \end{Bmatrix} + \dot{\mathbf{T}} \begin{Bmatrix} {}^{C}\mathbf{X}_T \\ 1 \end{Bmatrix} \tag{23}$$

Based on the above discussion, the estimates of target's position and motion w.r.t. $\mathcal{G}$ are obtained by the integrated photogrammetry and AEKF algorithm.

3 Target-Aiming Kinematic Control

The kinematics of a robotic manipulator defines the kinematic relationship between the end-effector in Cartesian space and the robotic manipulator in joint space. For given joint angle vector and angular velocity vector of the actuators, denoted by $\boldsymbol{\Theta}$ and $\dot{\boldsymbol{\Theta}}$ respectively, the position and velocity of the end-effector w.r.t. $\mathcal{G}$, denoted by $\mathbf{X}_E$ and $\dot{\mathbf{X}}_E$ respectively, can be easily obtained by the so called forward kinematics, such as,

$$\mathbf{X}_E = \mathbf{f}\left(\boldsymbol{\Theta}\right) \tag{24}$$

$$\dot{\mathbf{X}}_E = \mathbf{J}\dot{\boldsymbol{\Theta}} \tag{25}$$

where $\mathbf{\Theta} \in \mathbf{R}^n$, $\mathbf{X}_E \in \mathbf{R}^m$ and $\mathbf{J}$ represents the Jacobian matrix of the robotic manipulator and is derived by the partial derivative of the right hand side of Eq. (24) w.r.t. $\mathbf{\Theta}$, such as, $\mathbf{J} = \partial \mathbf{f} / \partial \mathbf{\Theta}$.

In contrast, for the given position and velocity of the end-effector w.r.t. $\mathcal{G}$, the inverse kinematics is required to solve for the corresponding joint angle and angular velocity of the manipulator. It is well known that the inverse of Eq. (24) may induce multiple solutions or kinematic singularity due to the periodicity of trigonometric functions and the redundant geometric configuration of the robotic manipulator. Generally, extra constraints, such as the physical limits, are imposed to avoid the kinematic singularity and obtain unique solution of the joint angle vector, as presented in Vahrenkamp and Asfour (2015). By neglecting the null space solution, the inverse of Eq. (25) can be written as

$$\dot{\mathbf{\Theta}} = \mathbf{J}^{\dagger} \dot{\mathbf{X}}_E \tag{26}$$

where $\mathbf{J}^{\dagger}$ denotes the pseudo inverse of the Jacobian matrix $\mathbf{J}$.

It is usually assumed that the Jacobian matrix $\mathbf{J}$ is full row rank in the literature of robotics. Thus, the pseudo inverse may be simplified to the right inverse, such that,

$$\mathbf{J}^{\dagger} = \mathbf{J}^T \left(\mathbf{J} \mathbf{J}^T \right)^{-1} \tag{27}$$

Based on its current position and velocity, the future position of the target w.r.t. $\mathcal{G}$, denoted by $\mathbf{X}_{T'}$, can be predicted as

$$\mathbf{X}_{T'} = \mathbf{X}_T + \dot{\mathbf{X}}_T t_s \tag{28}$$

In order to guide the end-effector moving directly towards the predicted target position in the next moment, the instantaneous velocity of the end-effector in the Cartesian space should have the same direction with the vector defined by $\mathbf{X}_{T'} - \mathbf{X}_E$, such that,

$$\dot{\mathbf{X}}_E = \lambda \mathbf{n}_{ET'} \tag{29}$$

where λ is the positive scale factor of the instantaneous velocity of the end-effector and $\mathbf{n}_{ET'}$ denotes the unit vector of $\mathbf{X}_{T'} - \mathbf{X}_E$, such that

$$\mathbf{n}_{ET'} = \frac{\mathbf{X}_{T'} - \mathbf{X}_E}{\left\| \mathbf{X}_{T'} - \mathbf{X}_E \right\|} \tag{30}$$

Accordingly, the position of the end-effector would be updated by

$$\mathbf{X}_{E'} = \mathbf{X}_E + \dot{\mathbf{X}}_E t_s \tag{31}$$

The requirement of $\mathbf{X}_{T'} = \mathbf{X}_{E'}$ at the capture yields

$$\dot{\mathbf{X}}_T - \dot{\mathbf{X}}_E = -t_s^{-1} \left(\mathbf{X}_T - \mathbf{X}_E \right) \tag{32}$$

Thus, the proposed visual servo control law is stable because the time t_s is always positive.

Note that λ is unknown and should be determined. Substituting Eq. (29) into Eq. (26) leads to

$$\dot{\mathbf{\Theta}} = \lambda \mathbf{J}^{\dagger} \mathbf{n}_{ET'} \tag{33}$$

Define $\dot{\theta}_i$, p_{ij} and $n_{ET'j}$ as the elements of $\dot{\mathbf{\Theta}}$, $\mathbf{J}^{\dagger}$ and $\mathbf{n}_{ET'}$, respectively. Then, Eq. (33) can be decomposed into n scalar equations, such that,

$$\dot{\theta}_i = \lambda \sum_{j=1}^{m} p_{ij} n_{ET'j}, \quad i = 1, 2, \cdots, n. \tag{34}$$

Assume the angular velocities of joint actuators in both forward and reversal directions are limited by $\dot{\mathbf{\Theta}}_{\max} = \left\{ \dot{\theta}_{\max 1}, \dot{\theta}_{\max 2}, \cdots, \dot{\theta}_{\max n} \right\}^T$. Applying the box constraint $-\dot{\theta}_{\max i}, \le \theta_i \le \dot{\theta}_{\max i}$ to the right hand side of Eq. (34) yields

$$-\dot{\theta}_{\max i} \le \lambda \sum_{j=1}^{m} p_{ij} n_{ET'j} \le \dot{\theta}_{\max i}, \quad i = 1, 2, \cdots, n. \tag{35}$$

Solving the inequalities in Eq. (35) independently for λ yields n sets of possible solutions, each denoted by λ_i. The intersection of n sets of λ_i defines the desired scale factor of the instantaneous velocity of the end-effector,

$$\lambda_d = \max\left[\lambda_1 \cap \lambda_2 \cap \cdots \cap \lambda_n\right] \tag{36}$$

Substituting Eq. (36) into Eq. (33) yields the equivalent instantaneous angular velocity vector of the joint actuators, denoted by $\dot{\mathbf{\Theta}}_\upsilon$, such as

$$\dot{\mathbf{\Theta}}_\upsilon = \lambda_d \mathbf{J}^\dagger \mathbf{n}_{ET'} \tag{37}$$

Accordingly, the incremental angular position control input of the joint actuators in the next time step, which drives the end-effector towards the target directly, is obtained as

$$\delta\mathbf{\Theta} = \dot{\mathbf{\Theta}}_\upsilon t_s \tag{38}$$

By applying the control input vector in Eq. (38) to the joint actuators, the end-effector moves towards the target by an increment at each time interval in the Cartesian space. The procedure defined by Eq. (22) - (38) is iteratively performed until the position error between the end-effector and the target is within a pre-defined tolerance and the capture action will be taken at the end of the procedure.

4 Experimental Validation

The proposed integrated motion estimation algorithm and incremental kinematic control strategy are validated experimentally on an independent target system and a custom built robotic manipulator with an eye-in-hand configuration. As shown in Fig. 1, the independent target system is driven by a single stepper motor fixed at the top right corner and anchored on the ceiling at the top left corner. By operating the target system independently, a non-cooperative target is generated for the robotic manipulator. Four carefully designed low-noise feature shapes are adopted here. By detecting and grouping the corners of each feature shape, the centers of feature shapes are calculated and tracked for the kinematic state estimation of the non-cooperative target.

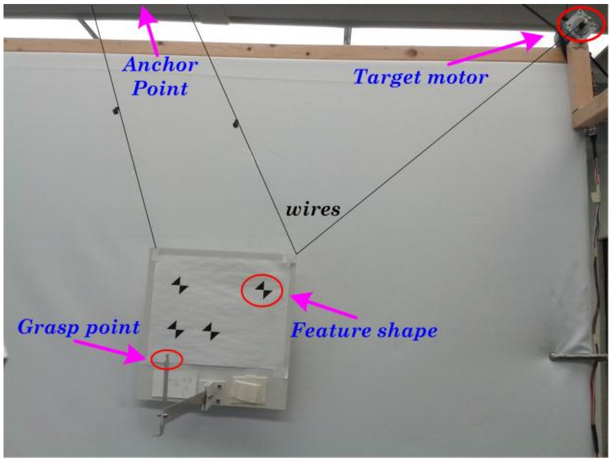


Fig. 1 Experimental setup of the non-cooperative target system.

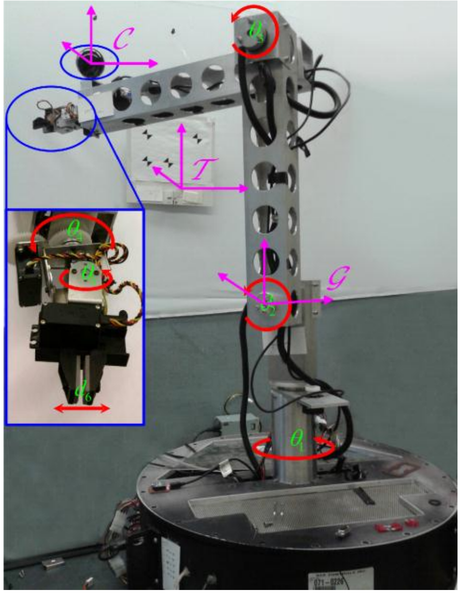


Fig. 2 Custom-built robotic manipulator with eye-in-hand configuration.

The experimental setup of the custom built robotic manipulator with an eye-in-hand configuration is shown in Fig. 2. The motion of the end-effector is controlled by three revolute joints, namely torso, shoulder and elbow, which are denoted by θ_1, θ_2 and θ_3, respectively. It should be noted that the degrees of freedom θ_4 and θ_5 are designated for grasping the target and are not used in the current control law. Shaft encoders were employed to measure joint angles in real time. Home position of the robotic manipulator is set to $\theta_1 = 0^\circ$, $\theta_2 = 90^\circ$, $\theta_3 = 0^\circ$. The link length between the shoulder and elbow joints is 0.4540 m and the link length between the elbow and the end-effector is 0.4445 m, as shown in Fig. 2. In order to further simplify Eq. (27), physical constraints are introduced to avoid the kinematic singularity of the robotic manipulator. Accordingly, the pseudo inverse of the Jacobian can be calculated by the normal inverse.

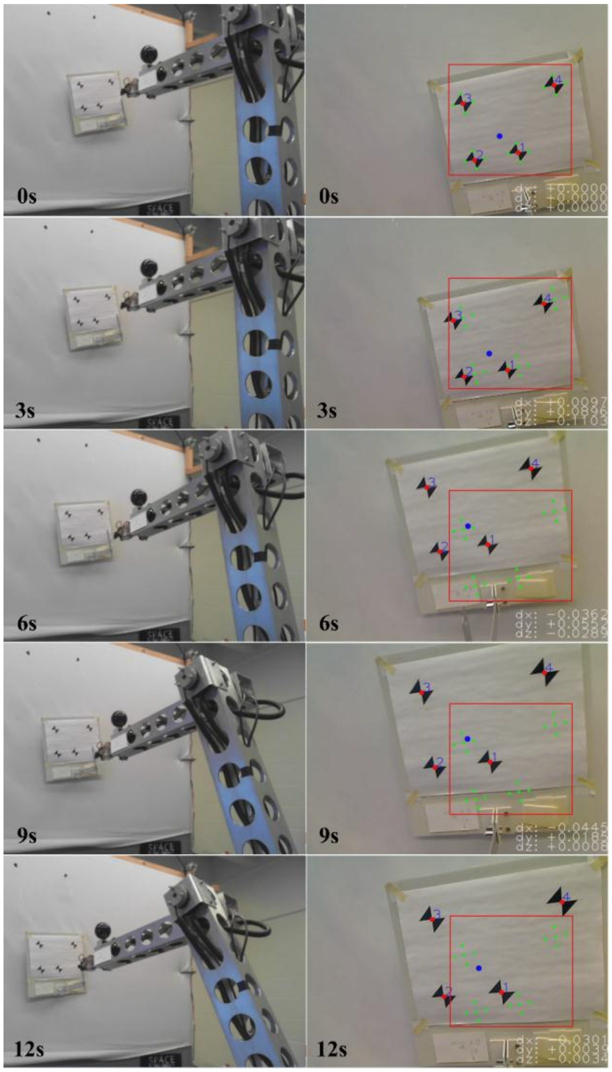


Fig. 3 Autonomous robotic capture process of the non-cooperative target.

The experimental results are shown in Figs. 3 - 7. Fig. 3 shows an autonomous robotic capture process of a non-cooperative target. Initially, the robotic manipulator was at the home position. The red rectangle indicates the region of interest for the image processing at the beginning. After the target is locked, the optical flow function of the OpenCV library takes over the tracking task of the four features on the target. The target was activated 50 sampling time steps (roughly two seconds) ahead of the robotic manipulator in order to become a dynamic non-cooperative target. The capture was deemed to successfully achieve when the distance between the end-effector and the target was reduced within the predefined tolerance.

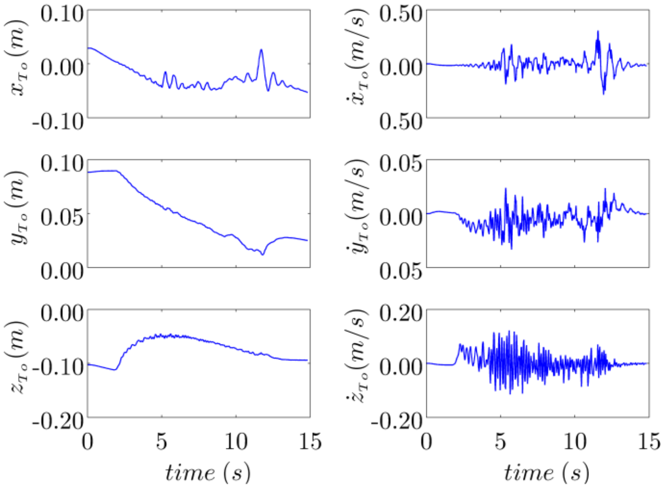


Fig. 4 Estimation results of the target motion w.r.t. $\mathcal{C}$.

To analyze the capture process in detail, Fig. 4 illustrates the estimates of target's motion by the eye-in-hand camera w.r.t. $\mathcal{C}$. During the experiment, the target motor was independently programmed to move the target within the workspace of the robotic manipulator. As mentioned above, the robotic manipulator was activated 50 sampling time steps after the activation of the target. This is reflected in Fig. 4 where the estimated target motion is quite smooth because the camera was stationary. Once the robotic manipulator was activated, the eye-in-hand camera was no longer stationary and the estimates of the target's motion were coupled with the motion of the robotic manipulator and become oscillation. This is much obviously shown in the velocity estimates.

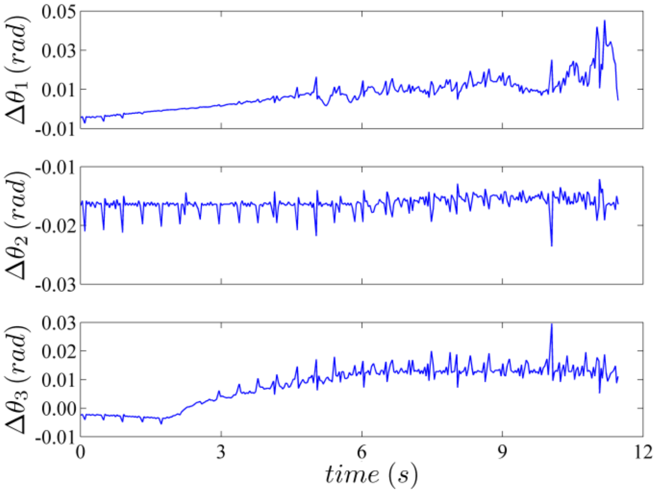


Fig. 5 Time history of incremental control input of joint actuators.

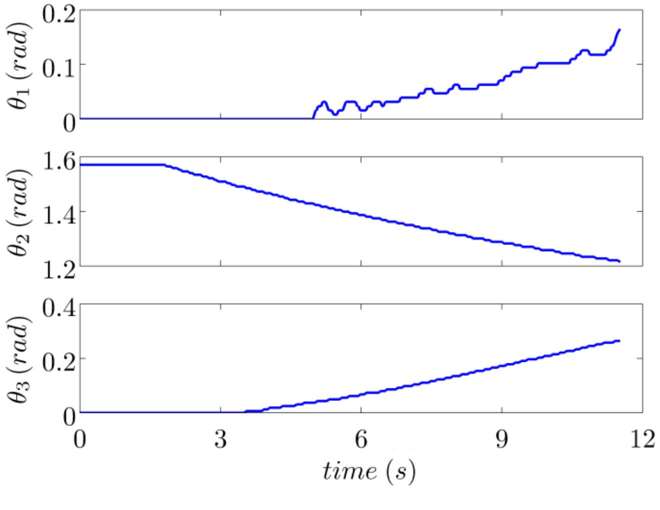


Fig. 6 Time history of joint angles measured by the shaft encoders.

Once the target motion is estimated, according to Eqs. (22) - (38), the incremental control inputs for the joint actuators are obtained as shown in Fig. 5. Joint angles of the robotic manipulator are measured by the shaft encoders directly and fed back to the controller in real time, as shown in Fig. 6. As can be seen, the robotic manipulator behaves almost uniformly in joint space, except the torso angle θ_1, which is affected by the flexible coupling between the torso actuator and the first link. This kind of mechanical design is to ease the alignment requirement of rotational axes. Nevertheless, it shows the control strategy is robust to handle the disturbance not considered in the control law.

Fig. 7 describes the time histories of the positions of the end-effector and the target. Since the position of the end-effector is w.r.t. $\mathcal{G}$, in order to compare the position of the end-effector with the target, the target position is transformed to $\mathcal{G}$ by Eq. (22), represented by ${}^{G}x_{To}$, ${}^{G}y_{To}$ and ${}^{G}z_{To}$ in the legends. It can be seen that, after the robotic manipulator was activated, the end-effector approached and captured the target autonomously at around 12 seconds successfully. The experimental results demonstrate the effectiveness and robustness of the estimation algorithm for the non-cooperative target as well as the kinematics based control strategy for autonomous robotic capture.

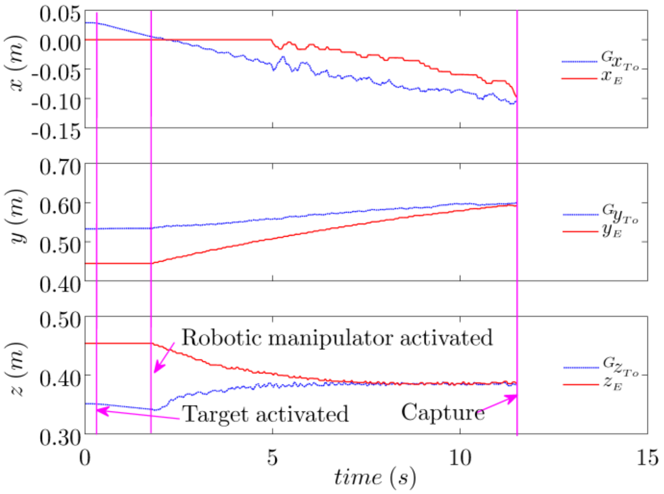


Fig. 7 Time histories of positions of the end-effector and the target, w.r.t. $\mathcal{G}$.

5 Conclusion

This paper presents a novel incremental kinematic control strategy for the visual servo robotic manipulator with eye-in-hand configuration to perform autonomous capture of a non-cooperative target. A vision based estimation algorithm that integrates the photogrammetry and AEKF is developed for the kinematic state estimation of the non-cooperative target. Initialized by the photogrammetry and enhanced by solely adaptive process noise distribution of the Kalman filtering method, the motion of a non-cooperative target by an eye-in-hand camera is successfully tracked and accurately estimated. In order to avoid the multiple solutions problem of the inverse kinematics in robotics, incremental joint control input is generated by the kinematics based control. Since the end-effector is always aiming to the target position in the next moment, the possibility of losing target tracking by the eye-in-hand camera is reduced substantially. Validating experiments have been performed by a custom robotic manipulator. The test results of a successful capture demonstrate the effectiveness and robustness of the estimation algorithm as well as the visual servo incremental kinematic control strategy.

Acknowledgement

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Aghili, F. (2012). A Prediction and Motion-Planning Scheme for Visually Guided Robotic Capturing of Free-Floating Tumbling Objects With Uncertain Dynamics. *Ieee Transactions on Robotics, 28*(3), 634-649.

Bac, C. W., van Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting

Robots for High-value Crops: State-of-the-art Review and Challenges Ahead. *Journal of Field Robotics, 31*(6), 888-911.

Chen, S. Y. (2012). Kalman Filter for Robot Vision: A Survey. *Ieee Transactions on Industrial Electronics, 59*(11), 4409-4420.

Dong, G., & Zhu, Z. H. (2015). Position-based visual servo control of autonomous robotic manipulators. *Acta Astronautica, 115*, 291-302.

Dong, G., & Zhu, Z. H. (2016). Autonomous Robotic Capture of Non-cooperative Target by Adaptive Extended Kalman Filter Based Visual Servo. *Acta Astronautica*.

Flores-Abad, A., Ma, O., Pham, K., & Ulrich, S. (2014). A review of space robotics technologies for on-orbit servicing. *Progress in Aerospace Sciences, 68*, 1-26.

Ghadiok, V., Goldin, J., & Ren, W. (2012). On the design and development of attitude stabilization, vision-based navigation, and aerial gripping for a low-cost quadrotor. *Autonomous Robots, 33*(1-2), 41-68.

Hsiao, K., Kaelbling, L. P., & Lozano-Perez, T. (2011). Robust grasping under object pose uncertainty. *Autonomous Robots, 31*(2-3), 253-268.

Ignakov, D., Okouneva, G., & Liu, G. J. (2012). Localization of a door handle of unknown geometry using a single camera for door-opening with a mobile manipulator. *Autonomous Robots, 33*(4), 415-426.

Janabi-Sharifi, F., & Marey, M. (2010). A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing. *Ieee Transactions on Robotics, 26*(5), 939-947.

Jankovic, M., Paul, J., & Kirchner, F. (2015). GNC architecture for autonomous robotic capture of a non-cooperative target: preliminary concept design. *Advances in Space Research*.

Larouche, B. P., & Zhu, Z. H. (2014). Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control.

Autonomous Robots, 37(2), 157-167.

Liang, B., Xu, Y. S., & Bergerman, M. (1998). Mapping a space manipulator to a dynamically equivalent manipulator. *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme,* 120(1), 1-7.

Myers, K. A., & Tapley, B. D. (1976). Adaptive Sequential Estimation with Unknown Noise Statistics. *Ieee Transactions on Automatic Control, 21*(4), 520-523.

Sabatini, M., Monti, R., Gasbarri, P., & Palmerini, G. B. (2013). Adaptive and robust algorithms and tests for visual-based navigation of a space robotic manipulator. *Acta Astronautica, 83*, 65-84.

Smith, J. A., Jivraj, J., Wong, R., & Yang, V. (2015). 30 Years of Neurosurgical Robots: Review and Trends for Manipulators and Associated Navigational Systems. *Ann Biomed Eng*, 1-11.

Song, Q., & Han, J.-D. (2008). An adaptive UKF algorithm for the state and parameter estimations of a mobile robot. *Acta Automatica Sinica, 34*(1), 72-79.

Umetani, Y., & Yoshida, K. (1989). Resolved Motion rate control of space manipulators with generalized Jacobian matrix. *IEEE Transactions on Robotics and Automation,* 5(3), 303-314.

Vafa, Z., & Dubowsky, S. (1987). On the dynamics of manipulators in space using the virtual manipulator approach. *Robotics and Automation. Proceedings. 1987 IEEE International Conference* 4, 579-585.

Vahrenkamp, N., & Asfour, T. (2015). Representing the robot's workspace through constrained manipulability analysis. *Autonomous Robots, 38*(1), 17-30.

Yoshida, K. (2009). Achievements in Space Robotics Expanding the Horizons of Service and Exploration. *Ieee Robotics & Automation Magazine, 16*(4), 20-28.

Reference Paper F

Kinematics-based Incremental Visual Servo for Robotic Capture of Non-cooperative Target[4]

Gangqi Dong and Z. H. Zhu

Abstract

This paper presents a kinematics-based incremental visual servo control scheme for robotic manipulators with eye-in-hand configuration to capture non-cooperative targets autonomously. The vision system is adopted to estimate the three dimensional position and motion of the target in real time by an integrated algorithm of photogrammetry and adaptive extended Kalman filter. The unknown intercept point of trajectories of the target and the end-effector is predicted and updated dynamically based on the target estimates, and is served as the desired position of the end-effector. An incremental control law is developed for the robotic manipulator to avoid the multiple solutions of the inverse kinematics. The proposed control scheme is validated experimentally by a custom built robotic manipulator with an eye-in-hand configuration. The experimental results demonstrate the proposed control scheme is effective and reliable.

[4] Submitted to **IEEE/ASME Transactions on Mechatronics** on June 03, revised form submitted on August 05, 2016.

Index Terms—Robotic manipulator, non-cooperative target, autonomous capture, visual servo, incremental kinematic control

1 Introduction

Robotic manipulators have been extensively adopted in numerous fields to satisfy the growing demands of dexterity, efficiency, and automation [1-4]. Among them, autonomous capture of a target attracts increasing attention in robotics [5-9], especially in space applications when the target is non-cooperative [10-12]. Several on-orbit-servicing missions were successfully performed with human participation, while the fully autonomous robotic capture of non-cooperative targets is still facing many challenges [11, 13], such as, the target estimation and the robotic control.

The target motion estimation is essential to control the robotic manipulator in order to achieve desired position/motion of the end-effector. Due to the non-contact and non-invasive nature, vision system is generally favored in robotic control for target estimation [14-17] and the corresponding control approach is known as visual servo [18, 19]. The vision system used for target detecting and tracking purpose can be either mounted on a robotic manipulator, known as eye-in-hand, or fixed to the workspace, known as eye-to-hand. The eye-in-hand configuration may provide a near and detailed scene of the target. The visual information is coupled with the motion of the robotic manipulator and the effect of the latter could be decoupled with the feedback of the state of the robotic manipulator. The accuracy and details of the target increase as the end-effector approaches the target. Alternatively, the eye-to-hand configuration monitors the whole workspace and provide a global but less accurate scene of the target. In some specific circumstances during the robotic operations, the view of target may be blocked by the robotic manipulator in the eye-to-hand configuration, leading to the tracking failure. Therefore, the eye-in-hand configuration is adopted in the current study to ensure the high control accuracy in the capture phase. In the visual servo control, it is imperative to keep the target in the camera's field of view, especially for eye-in-hand configuration when the end-effector is in the vicinity of target. Common approaches in the literature are adoption either wide angle cameras or motorized cameras with orientation control. However, the control of camera orientation will in generally couple with the control of end-effector and complicated the control law design. The current work is focused on the development of kinematics-based incremental visual servo for robotic capture. Therefore, a camera with large field of view is adopted in the current study to simplify the development and validation of the proposed visual servo control scheme.

Many vision-based methods for target estimation were developed, such as the geometric method, learning-based optimization method, offline estimation method and filtering-based method. Based on requirements for real time autonomous capture, an integrated algorithm of photogrammetry and adaptive extended Kalman filter (AEKF) [17] is employed for real time estimation of position and velocity of a non-cooperative target in the current work.

Once the pose and motion of the target are estimated, a control strategy is employed to capture the dynamic target. In our previous works, the estimated current target position [17] or the estimated target position in the next time instant [12] was assumed as the desired position in the proposed control law. It was found that the end-effector may not be able to capture the target in both cases if the end-effector lags behind the target initially. To address this issue, the current work assume the intercept point of trajectories of the target and the end-effector as the desired position in the control law. In addition, the velocity of the end-effector at interception shall be aligned with the target's velocity as much as possible to avoid hard contact at capture. However, the determination of the intercept point is challenging since the target trajectory is unknown in advance due to the non-cooperative nature. This challenge is further complicated by the fact that the time-variant nonlinear motion of the end-effector will induce variation of tracking time and affect the determination of the intercept point. Furthermore, for any given position of the end-effector in the Cartesian space, the inverse kinematics of the robotic manipulator must be performed to obtain the corresponding angular position in the joint space. Difficulty arises in the inverse kinematics where multiple solutions occurs due to the periodicity of trigonometric functions and the redundant geometric configuration of the robotic manipulator. In our previous work [17], the issue of multiple solutions of direct inverse kinematics was avoided by considering work space constraints, which is problem specific. In the current work, a kinematics-based incremental control strategy is proposed to avoid the multiple solutions in a generic way.

The proposed approach is validated experimentally using a dynamic non-cooperative target and a custom built robotic manipulator with eye-in-hand configuration. The experimental results demonstrated the effectiveness and reliability of the proposed control strategy for autonomous capture of a non-cooperative target in real time.

2 Vision-based Target Estimation

Consider a robotic manipulator with the eye-in-hand configuration working in an inertia space together with a non-cooperative target. A global frame is fixed in the inertia space and is denoted by $\mathcal{F}_G$. The target frame is attached to the target body and is denoted by $\mathcal{F}_T$. The camera frame, denoted by $\mathcal{F}_C$, is placed in the image plane of the camera with the origin at the center of the image plane. The coordinates of the origin of $\mathcal{F}_T$ in $\mathcal{F}_C$ are defined as ${}^C\mathbf{X}_{To} = \left\{ {}^C x_{To}, {}^C y_{To}, {}^C z_{To} \right\}^T$. Furthermore, it is assumed that the non-cooperative target contains λ known feature points in $\mathcal{F}_T$ denoted by ${}^T\mathbf{X}_i$, where the subscript $i = 1, 2, \cdots \lambda$. Based on the above definitions and assumptions, the coordinates of feature points of that target in $\mathcal{F}_C$ can be expressed as

$$ {}^C\mathbf{X}_i = {}^C_T\mathcal{R}\left({}^C\mathbf{\Omega}_T \right) {}^T\mathbf{X}_i + {}^C\mathbf{X}_{To} \tag{1} $$

where ${}^C_T\mathcal{R}(\cdot)$ stands for the rotational transform matrix from $\mathcal{F}_T$ to $\mathcal{F}_C$ and is formed by trigonometric functions of Eulerian angles ${}^C\mathbf{\Omega}_T = \left\{ {}^C\theta_{Tx}, {}^C\theta_{Ty}, {}^C\theta_{Tz} \right\}^T$ between these two reference frames. It should be noted that ${}^C\mathbf{X}_{To}$ and ${}^C\mathbf{\Omega}_T$ are the position and orientation of $\mathcal{F}_T$ with respect to $\mathcal{F}_C$ and are both unknown for a non-cooperative target.

For a pinhole camera, the feature points are projected onto the image plane of the camera by the collinearity condition

$$ \mathbf{Z}_i = \mathcal{P}\left({}^C\mathbf{X}_i, f \right) \tag{2} $$

where f is the focal length of the camera and $\mathbf{Z}_i \in \mathbf{R}^2$ is the corresponding two dimensional image coordinates of each feature points in meters, which can be obtained by performing image processing.

Consequently, the relationship between the measurements $\mathbf{Z}_i$ and the unknown pose (position and orientation) of the non-cooperative target is established by (1) and (2). For each of the feature points, two equations with six unknowns are obtained. Therefore, at least three feature points ($\lambda = 3$) should be available to solve six unknowns. In order to eliminate the ambiguity caused by the periodic solutions of trigonometric functions, four feature points

($\lambda = 4$) are usually adopted and the unknowns are solved by the iterative least square approach based photogrammetry [8]. Noting that the photogrammetry does not solve for the motion of the target and are prone to the measurement noise due to its memoryless nature. The measurement noise in the image processing may be amplified by the photogrammetry and propagate into the visual servo control algorithm. To overcome this limitation, an AEKF algorithm [17] is developed to estimate the motion of the non-cooperative target in real time.

Define the state variable of the target with respect to $\mathscr{F}_C$ as

$$\mathbf{X} = \left\{ {}^{C}x_{To}, {}^{C}\dot{x}_{To}, {}^{C}y_{To}, {}^{C}\dot{y}_{To}, {}^{C}z_{To}, {}^{C}\dot{z}_{To}, {}^{C}\theta_{Tx}, {}^{C}\dot{\theta}_{Tx}, {}^{C}\theta_{Ty}, {}^{C}\dot{\theta}_{Ty}, {}^{C}\theta_{Tz}, {}^{C}\dot{\theta}_{Tz} \right\}^T \tag{3}$$

The target motion is approximated by the first order linear motion if the sampling time is sufficient small. Thus, the process model can be written as

$$\mathbf{X}_{k+1} = \mathbf{A}\mathbf{X}_k + \mathbf{B}\boldsymbol{\omega}_k \tag{4}$$

where the subscripts k and $k+1$ indicate the current and next sampling time steps, $\boldsymbol{\omega}$ is the process noise formed by the acceleration vector as $\boldsymbol{\omega} = \left\{ {}^{C}\ddot{x}_{To}, {}^{C}\ddot{y}_{To}, {}^{C}\ddot{z}_{To}, {}^{C}\ddot{\theta}_{Tx}, {}^{C}\ddot{\theta}_{Ty}, {}^{C}\ddot{\theta}_{Tz} \right\}^T$, $\mathbf{A} \in \mathbf{R}^{12\times12}$ and $\mathbf{B} \in \mathbf{R}^{12\times6}$ are the coefficient matrices with the elements formed by functions of sampling time t_s, such as

$$\begin{cases} A_{i,i} = 1 & \text{for } i = 1, 2, \cdots, 12 \\ A_{2i-1,2i} = t_s & \text{for } i = 1, 2, \cdots, 6 \\ A_{i,j} = 0 & \text{for all the other elements} \end{cases} \tag{5}$$

$$\begin{cases} B_{2i-1,i} = t_s^2/2 & \text{for } i = 1, 2, \cdots, 6 \\ B_{2i,i} = t_s & \text{for } i = 1, 2, \cdots, 6 \\ B_{i,j} = 0 & \text{for all the other elements} \end{cases} \tag{6}$$

The process noise is assumed to obey the normal distribution with the mean $\mathbf{q}$ and covariance $\mathbf{Q}$, such that

$$\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{q}, \mathbf{Q}) \tag{7}$$

Based on (1) and (2), the measurement model is defined as

$$\mathbf{Z}_k = \mathcal{H}\left(\mathbf{X}_k\right) + \boldsymbol{\mu}_k \tag{8}$$

$$\mathcal{H}\left(\mathbf{X}_k\right) = \mathcal{P}\left({}^{C}_{T}\mathcal{R}\left({}^{C}\boldsymbol{\Omega}_T\right)^T \mathbf{X}_i + {}^{C}\mathbf{X}_{To}, f\right) \tag{9}$$

where $\boldsymbol{\mu}$ stands for the measurement noise of the camera and is assumed to obey the normal distribution with zero mean and constant covariance matrix, i.e. $\boldsymbol{\mu} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}\right)$.

The camera model in (8) and (9) is highly nonlinear and the direct application of common extended Kalman filters may lead to poor estimates due to the high nonlinearity of the measurement model and unsuitable approximation of the noise distribution. To deal with this issue, AEKF was proposed to improve the performance of common extended Kalman filters by adapting the distribution of the noises. In the current case, although the covariance matrix of the measurement noise $\mathbf{R}$ can be determined experimentally in advance, the mean and covariance matrix of the process noise, $\mathbf{q}$ and $\mathbf{Q}$, are time-variant and have to be updated. The estimation process with adaptive process noise is concisely described as follows.

Based on the process model in (4), the state variable $\mathbf{X}$ and its covariance matrix $\mathbf{P}$ in the next step can be predicted by

$$\mathbf{X}_{k+1|k} = \mathbf{A}\mathbf{X}_{k|k} + \mathbf{B}\mathbf{q}_k \tag{10}$$

$$\mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^T + \mathbf{B}\mathbf{Q}_k\mathbf{B}^T \tag{11}$$

where the subscript $\alpha|\beta$ denotes the variables estimated at step α with respect to step β, $\mathbf{q}_k$ and $\mathbf{Q}_k$ denote the mean and covariance matrix of the process noise at step k, respectively.

The Kalman gain is given by

$$\mathbf{K}_g = \mathbf{P}_{k+1|k}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k+1|k}\mathbf{H}_k^T + \mathbf{R}\right)^{-1} \tag{12}$$

where $\mathbf{H}$ denotes the Jacobian of the measurement model and is defined by

$$\mathbf{H}_k = \partial \mathcal{H}(\mathbf{X}) / \partial \mathbf{X}\big|_{\mathbf{X}=\mathbf{X}_{k+1|k}} \tag{13}$$

Following the classic procedure of Kalman filtering, the estimates in (10) and (11) are updated once the measurement is obtained for the current step, such that

$$\mathbf{X}_{k+1|k+1} = \mathbf{X}_{k+1|k} + \mathbf{K}_g \left(\mathbf{Z}_k - \mathcal{H}\left(\mathbf{X}_{k+1|k} \right) \right) \tag{14}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_g \mathbf{H}_k \mathbf{P}_{k+1|k} \tag{15}$$

In order to adaptive the distribution of process noise, an intuitive approximation of $\mathbf{q}$ at step j is defined by [20]

$$\hat{\mathbf{q}}_j = \left(\mathbf{B}^T \mathbf{B} \right)^{-1} \mathbf{B}^T \left(\mathbf{X}_{j+1|j+1} - \mathbf{A}\mathbf{X}_{j|j} \right) \tag{16}$$

Further assume the process noise is independent and uniformly distributed over N sampling time steps, then the unbiased estimate of $\mathbf{q}$ at step k can be defined by

$$\mathbf{q}_k = \frac{1}{N} \sum_{j=k-N+1}^{k} \hat{\mathbf{q}}_j = \mathbf{q}_{k-1} + \frac{1}{N} \left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N} \right) \tag{17}$$

Accordingly, the covariance matrix of the process noise is given by

$$\mathbf{Q}_k = \mathbf{Q}_{k-1} + \frac{1}{N-1} \begin{pmatrix} \left(\hat{\mathbf{q}}_k - \mathbf{q}_k \right)\left(\hat{\mathbf{q}}_k - \mathbf{q}_k \right)^T - \left(\hat{\mathbf{q}}_{k-N} - \mathbf{q}_k \right)\left(\hat{\mathbf{q}}_{k-N} - \mathbf{q}_k \right)^T \\ + \frac{1}{N} \left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N} \right)\left(\hat{\mathbf{q}}_k - \hat{\mathbf{q}}_{k-N} \right)^T + \frac{N-1}{N} \left(\mathbf{\Lambda}_{k-N} - \mathbf{\Lambda}_k \right) \end{pmatrix} \tag{18}$$

$$\mathbf{\Lambda}_k = \left(\mathbf{B}^T \mathbf{B} \right)^{-1} \mathbf{B}^T \left(\mathbf{A}\mathbf{P}_{k|k} \mathbf{A}^T - \mathbf{P}_{k+1|k+1} \right) \mathbf{B} \left(\mathbf{B}^T \mathbf{B} \right)^{-1} \tag{19}$$

The AEKF iteration defined by (10)-(19) is initialized by the result of photogrammetry in order to accelerate the iterating convergence.

It should be noted that the obtained target estimates are in $\mathcal{F}_C$. Because the vision system is configured as eye-in-hand, the estimated position and velocity should be transformed from $\mathcal{F}_C$ to $\mathcal{F}_G$. Denoted the transformation matrix by $\mathbf{T} \in \mathbf{R}^{4\times 4}$, which is obtained by the Denavit-Hartenberg transformation [109], such that

$$\left\{\mathbf{X}_T, 1\right\}^T = \mathbf{T}\left\{{}^C\mathbf{X}_{To}, 1\right\}^T \tag{20}$$

where $\mathbf{X}_T$ denotes the estimated position of the target expressed in $\mathcal{F}_G$. Notes that the elements of $\mathbf{T}$ are formed by the trigonometric functions of joint angles and the length of links of the robotic manipulator.

Taking the time derivative of (20) yields

$$\left\{\dot{\mathbf{X}}_T, 0\right\}^T = \mathbf{T}\left\{{}^C\dot{\mathbf{X}}_{To}, 0\right\}^T + \dot{\mathbf{T}}\left\{{}^C\mathbf{X}_{To}, 1\right\}^T \tag{21}$$

where $\dot{\mathbf{X}}_T$ denotes the estimated velocity of the target expressed in $\mathcal{F}_G$.

The elements of $\dot{\mathbf{T}}$ are functions of trigonometric functions of joint angles, joint velocities and the length of links of the robotic manipulator. We assume the joint angles can be measured by shaft encoders and the corresponding joint velocities can be obtained by a standard Kalman filter. Hereto the estimated position and velocity of the non-cooperative target are obtained in both $\mathcal{F}_C$ and $\mathcal{F}_G$.

3 Incremental Visual Servo Control

Generally, robotic manipulators are a sequence of joints and links with a dexterous end-effector attached to terminus of the kinematic chain. The joints and links mainly contribute to place the end-effector to a desired position and the dexterous end-effector can be utilized to align with the orientation of grasping point on a target in most robotic capture scenarios. Thus, the control for the alignment of gripper with the target's grasping feature can be decoupled from the position control of the end-effector, as presented in this work.

For a given angular position and velocity of the robotic manipulator, the position and velocity of the end-effector in $\mathcal{F}_G$ can be obtained by the forward kinematics as

$$\mathbf{X}_E = \mathbf{f}\left(\mathbf{\Theta}\right) \tag{22}$$

$$\dot{\mathbf{X}}_E = \mathbf{J}\dot{\mathbf{\Theta}} \tag{23}$$

where $\mathbf{\Theta} \in \mathbf{R}^n$ and $\dot{\mathbf{\Theta}} \in \mathbf{R}^n$ denote the angular position and velocity of the joints of robotic manipulator, $\mathbf{X}_E \in \mathbf{R}^m$ and $\dot{\mathbf{X}}_E \in \mathbf{R}^m$ denote the position and velocity of the end-effector, and $\mathbf{J} \in \mathbf{R}^{m \times n}$ is the Jacobian matrix of the robotic manipulator that is defined by $\mathbf{J} = \partial \mathbf{f} / \partial \mathbf{\Theta}$.

Based on the definition of pseudo inverse of a matrix, the inverse kinematics corresponding to (23) can be written as, if the null space solution is neglected,

$$\dot{\mathbf{\Theta}} = \mathbf{J}^{\dagger}\dot{\mathbf{X}}_E \tag{24}$$

where $\mathbf{J}^{\dagger}$ denotes the pseudo inverse of $\mathbf{J}$ and can be calculated by $\mathbf{J}^{\dagger} = \mathbf{J}^T\left(\mathbf{J}\mathbf{J}^T\right)^{-1}$ with the assumption that $\mathbf{J}$ is non-singular.

However, the inverse kinematics corresponding to (22) is complicated by multiple solutions due to the periodicity of trigonometric functions and the redundant geometric configuration of the robotic manipulator. In order to avoid the difficulties, an incremental visual servo control strategy for the robotic manipulator to intercept the non-cooperative target is developed as follows.

Assume the target is always within the workspace of the robotic manipulator and will be captured at the interception point $\mathbf{X}_C$ after tracking time t, which can be estimated by the first order approximation

$$\mathbf{X}_C = \mathbf{X}_T + \dot{\mathbf{X}}_T t \tag{25}$$

In order to intercept with the target, the end-effector is required to reach the same interception point within the same time period. Thus, the trajectory for the end-effector can be roughly designed as

$$\mathbf{X}_C = \mathbf{X}_E + \dot{\mathbf{X}}_E t \tag{26}$$

Because $\mathbf{X}_E$ can be obtained in accordance with (22) and $\mathbf{X}_T$ and $\dot{\mathbf{X}}_T$ can be obtained by the estimation algorithm in section II, the desired velocity of the end-effector is determined based on (25) and (26), such that

$$\dot{\mathbf{X}}_E = \frac{1}{t}\left\{\mathbf{X}_T + t\dot{\mathbf{X}}_T - \mathbf{X}_E\right\} \tag{27}$$

Substituting (27) into (23) yields

$$\dot{\mathbf{\Theta}} = \frac{1}{t}\mathbf{J}^{\dagger}\left\{\mathbf{X}_T + t\dot{\mathbf{X}}_T - \mathbf{X}_E\right\} \tag{28}$$

Applying the constraint of angular velocity to (28) yields

$$\dot{\theta}_{\min i} \le \frac{1}{t}\sum_{j=1}^{m} p_{ij}\left(x_{Tj} + t\dot{x}_{Tj} - x_{Ej}\right) \le \dot{\theta}_{\max i} \tag{29}$$

where the subscripts $i = 1, 2, \cdots, n$. $\dot{\theta}_i$, p_{ij}, x_{Tj}, $\dot{x}_{Tj}$, and x_{Ej} are the elements of $\dot{\mathbf{\Theta}}$, $\mathbf{J}^{\dagger}$, $\mathbf{X}_T$, $\dot{\mathbf{X}}_T$, and $\mathbf{X}_E$, respectively.

Since $t > 0$, the inequalities in (29) yields n solution intervals of tracking time, which are denoted by $t_1, t_2, \cdots, t_n$, respectively. Then, the predicted intercept time is determined as the lowest bound of the intersect of the solution intervals, such that

$$t = \min\left\{t_1 \cap t_2 \cap \cdots \cap t_n\right\} \tag{30}$$

Substituting the determined t into (28) yields the expected angular velocity of the robotic manipulator. Finally, the control input for the robotic manipulator in the next sampling step is obtained as the increment of angles in one sampling step, such as

$$\Delta\mathbf{\Theta} = \frac{t_s}{t}\mathbf{J}^{\dagger}\left\{\mathbf{X}_T + t\dot{\mathbf{X}}_T - \mathbf{X}_E\right\} \tag{31}$$

It is worth pointing out that the issue of multiple solutions associated with the total inverse kinematics is avoided by the incremental inverse kinematics in (31) if the time step t_s is small.

Furthermore, it can be seen from (27) that the velocity of the end-effector approaches to the velocity of the target as the relative distance between them diminishes, such that, $\left(\mathbf{X}_T - \mathbf{X}_E\right) \to 0$. This implies there will be no hard contact at capture because the velocity of the end-effector will align with the target.

After the end-effector moves one step towards the target, the prediction of the intercept point and the tracking time will be updated again with the new estimates of target state. As a result, the predicted intercept point and the tracking time are dynamically updated by the control loop until the end-effector intercept the target.

To prove the stability of the proposed visual servo, define the tracking error of the robotic capture as

$$\mathbf{E} = \mathbf{X}_T - \mathbf{X}_E \tag{32}$$

$$\dot{\mathbf{E}} = \dot{\mathbf{X}}_T - \dot{\mathbf{X}}_E \tag{33}$$

Equaling (25) and (26) yields

$$\dot{\mathbf{E}} = -\frac{1}{t}\mathbf{E} \tag{34}$$

Now define the Lyapunov function as

$$\mathbf{V} = \frac{1}{2}\mathbf{E}^T\mathbf{E} \tag{35}$$

The time derivative of (35) is

$$\dot{\mathbf{V}} = \mathbf{E}^T\dot{\mathbf{E}} \tag{36}$$

Substituting (34) into (36) leads to

$$\dot{\mathbf{V}} = -\frac{1}{t}\mathbf{E}^T\mathbf{E} \le 0 \tag{37}$$

Equation (37) indicates the proposed visual servo control law is asymptotically stable because the tracking time t is always positive. Thus, the position of the end-effector $\mathbf{X}_E$ is guaranteed to converge to the position of the target by the proposed visual servo control scheme even though the initial prediction of intercept point is rough.

Once the end-effector reaches the desired position, the alignment of the gripper with the grasping feature of the target is independently controlled based on the estimated orientation of the target in $\mathcal{F}_C$.

4 Experimental Results

The proposed incremental visual servo control scheme is validated experimentally by a custom built robotic manipulator with an eye-in-hand configuration. The target independently driven by a single stepper motor independent to the robot controller so that it can be treated as non-cooperative. As shown in Fig. 1, the target is suspended in the air by a wire. One end of wire was anchored on the ceiling at the top left corner, while the other end was attached to the stepper motor fixed at the top right corner. In order to improve the accuracy of image processing, four low noise and identical feature shapes were carefully designed [8]. The distances between centroids of any two feature shapes are greater than the size of the feature shape by two times. The corners of all feature shapes were detected by OpenCV functions and grouped into four groups according to the distance between any two corners. Then the centers of each feature shape were calculated. They were numbered based on the distance from the intersection point of diagonal lines of the quadrangle that is formed by the centers of feature shapes, from the smallest to the largest. Once numbered, the centers of feature shapes were fed to the target estimation algorithm. Before the testing, the target was activated first so that a dynamic non-cooperative target was generated. Furthermore, extra jittering was intentionally added on the target trajectory to demonstrate the strength of algorithms of target estimation and the visual servo control. The jittering was generated by driving the stepper motor with sudden starts and stops to induce vibrations/swings of the target because the target was hanging by flexible wires. After the activation of the target for 50 sampling steps (around 2s), the robotic manipulator was activated for the autonomous capture. The tolerance for the tracking error is set as (0.01m, 0.01m, 0.01m).

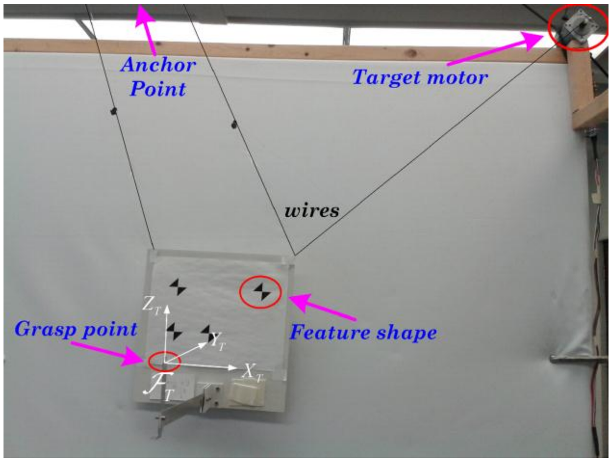


Fig. 1 Setup of the non-cooperative target.

Figure 2 shows the custom built robotic manipulator with an eye-in-hand camera mounted closely to the end-effector. The camera was a webcam. The pixel size, optical sensor (CCD) size and frame size (resolution) of the camera are 5.6x5.6 μm, 6.35 mm (1/4") and 640*480, respectively. The distortion is corrected automatically by the driver software from the camera manufacturer.

The main rotational joints of the robotic manipulator are driven by stepper motors, namely torso, shoulder, and elbow. They are denoted by θ_1, θ_2, and θ_3, respectively. Shaft encoders were mounted at the main joints to measure θ_1, θ_2, and θ_3 in real time. It should be noted that a flexible coupling is adopted at θ_1 in order to handle the misalignment of rotational axes between the actuator and the first link. The length of links and the joint speed limits of the robotic manipulator are listed in TABLE I, where the joint speed limits of each joint are assumed symmetric in both positive and negative directions, that is, $\dot{\theta}_{\min i} = -\dot{\theta}_{\max i}$.

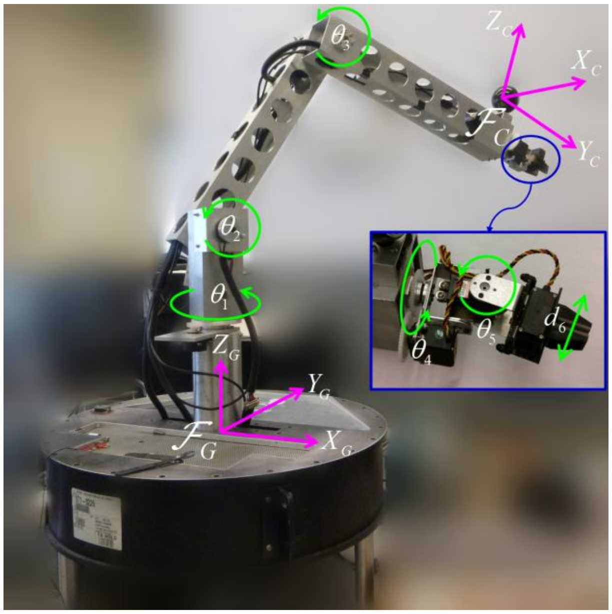


Fig. 2 Custom built robotic manipulator with the eye-in-hand configuration.

TABLE I Physical Properties of The Robotic Manipulator

Symbols	Descriptions	Values
$l_{1,2}$	Link length between torso and shoulder	0.1778m
$l_{2,3}$	Link length between shoulder and elbow	0.454025m
$l_{3,E}$	Link length between elbow and end-effector	0.4445m
$\dot{\theta}_{\max 1}$	Speed limit of torso	10.304 rad/s
$\dot{\theta}_{\max 2}$	Speed limit of shoulder	0.412 rad/s
$\dot{\theta}_{\max 3}$	Speed limit of elbow	0.412 rad/s

As shown in Fig. 2, the end-effector includes a gripper with two revolute joints and one prismatic joint. They are denoted by θ_4, θ_5 and d_6, respectively. All joints are driven by servo motors made by HiTEC. The alignments with the grasping feature of the target are fulfilled by the independently controlled end-effector, such as $\theta_4 = {}^C\theta_{Ty} + {}^C\dot{\theta}_{Ty} t_s$ and $\theta_5 = {}^C\theta_{Tz} + {}^C\dot{\theta}_{Tz} t_s$.

Two cases of experiments have been conducted to validate the proposed control scheme. The home position of the robotic manipulator was set to $\theta_1 = \theta_3 = 0°$ and $\theta_2 = 90°$ in case 1, with the initial position of the end-effector roughly 15 cm away from the target. In case 2, the home position of the robotic manipulator was set to $\theta_1 = 0°$, $\theta_2 = 115°$ and $\theta_3 = -25°$, with the initial position of the end-effector roughly 40 cm away from the target.

A. Case 1

First, the target position and velocity in $\mathcal{F}_C$ are estimated by the proposed estimation algorithm and compared with the results obtained by the photogrammetry. The comparisons are shown in Figure 3. It is noted that the estimates by the proposed algorithm are much smoother than that of the photogrammetry where the effect of jittering is amplified. In addition, the photogrammetry does not estimate the velocity while the proposed algorithm estimates both position and velocity at the same time. It can be seen in Fig. 3 that the target motion is coupled with the motion of the robotic manipulator due to the eye-in-hand configuration after the robot started 2s later. Because of this kinetic coupling effect, the estimated velocity is pretty noisy. The obvious oscillation in the final approaching stage is the evidence of the flexible coupling effect at the first joint of the robotic manipulator.

According to the estimated target motion, the dynamically predicted minimum tracking time t is shown in Fig. 4. The noise is mainly caused by the jittering motion of the target as well as the time-variant velocity of the end-effector. The jittering motion of the target causes the variation of the intercept point, and further affects the tracking time. Besides, the Jacobian is mainly formed by the products of trigonometric functions of joint angles and length of links. It largely depends on the joint angles. Thus, the velocity of the end-effector varies significantly to the interception point and in turn affects the tracking time. Nevertheless, the overall trend of the predicted minimum tracking time is reduced asymptotically as the end-effector approaches the target. This indicates the proposed control law is effective and stable.

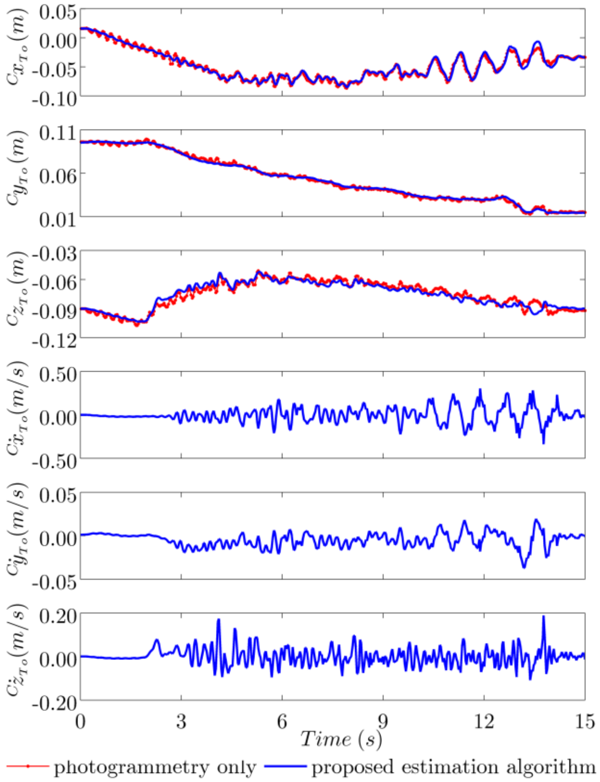


Fig. 3 Estimated target motion in case 1.

The angular position histories of the robotic manipulator recorded by the shaft encoders are shown in Fig. 5. In the first 2s, the robotic manipulator was stationary so that the angular positions of the robotic manipulator were remained at the home position. Then the robotic manipulator started to track the target. The oscillation in θ_1 is again caused by the flexible coupling at the first joint.

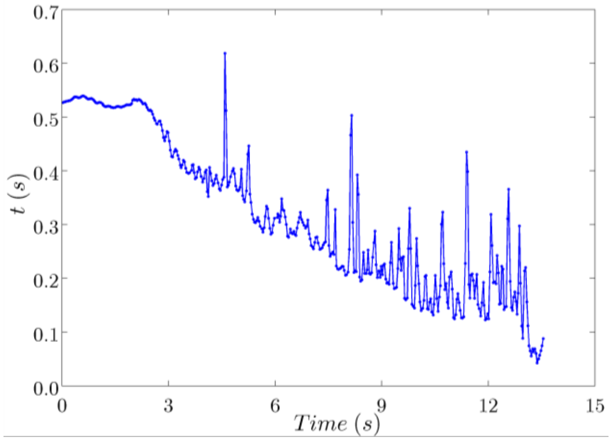


Fig. 4 Predicted minimum tracking time in case 1.

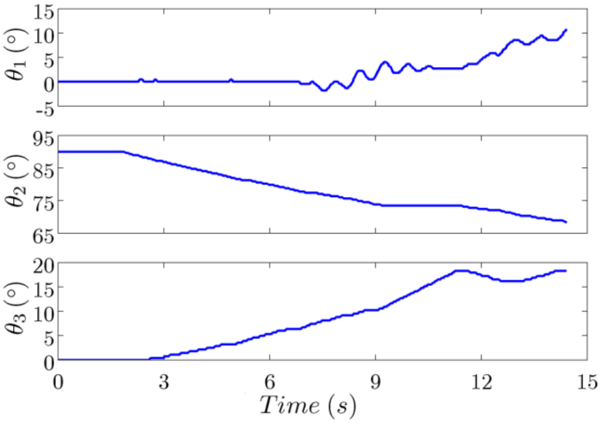


Fig. 5 Time history of angular position of the robotic manipulator in case 1.

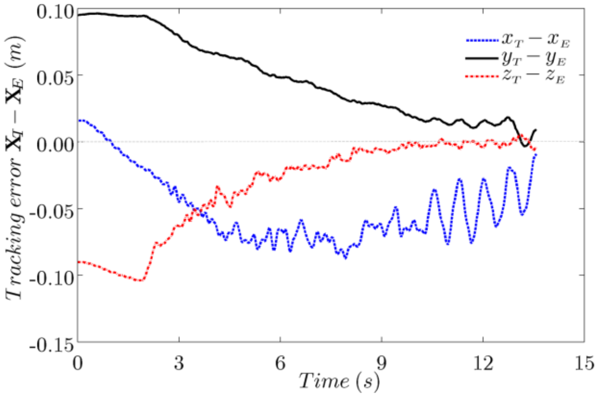


Fig. 6 Time history of tracking error in case 1.

The tracking error between the position of the end-effector and the target is shown in Fig. 6. It can be seen that the tracking errors approach zero and finally the end-effector intercepts the target at around 14s.

Intuitively, the robotic capture process is illustrated in Fig. 7. The pictures in the left column were taken by an external webcam for observation purpose only, and the images in the right column were recorded by the eye-in-hand camera. In the right column, the rectangles in red indicate the region of interest (ROI), which is selected manually before the capture process starts. Once the ROI was selected, 20 corners inside the ROI were detected by OpenCV functions and marked as green dots. Then the corners were grouped in four groups according to the distance between each of these two corners. The centers of each group, marked as red dots in the images, were calculated in order to apply the OpenCV optical flow function to track the target motion. The two advantages of tracking the centers are the higher image processing accuracy and the lower computational cost than tracking all the corners. It should be noted that once the capture process start, only the red dots will be tracked for the target estimation. This is the reason that the red rectangle of the ROI and the green dots are remained at the original position in the images of the eye-in-hand camera. The four feature centers form a quadrangle. The intersect point of the diagonal lines are calculated and marked as a blue dot in the images of Fig. 7. Then the four feature centers were numbered according to the distance between the intersect point and each of the feature centers, from the smallest to the largest. The order of the feature centers is fed to the target estimation algorithm together with image coordinates.

At the beginning, the robotic manipulator was at the home position as shown in the left column of the first row in Fig 7. Then, the pictures in the second row indicate that the target was activated while the robotic manipulator remained at its home position during the first 2s. After 2s, the robotic manipulator started to track and approach the target. The moment at around 8s was taken as an example of this stage and shown in the third row. Finally, the gripper was closed and secured the gasping feature when the tracking error of the end-effector is less than a preset tolerance and the gripper is aligned with grasping feature. This is shown in Fig. 7 at 14s in the left column of the last row.

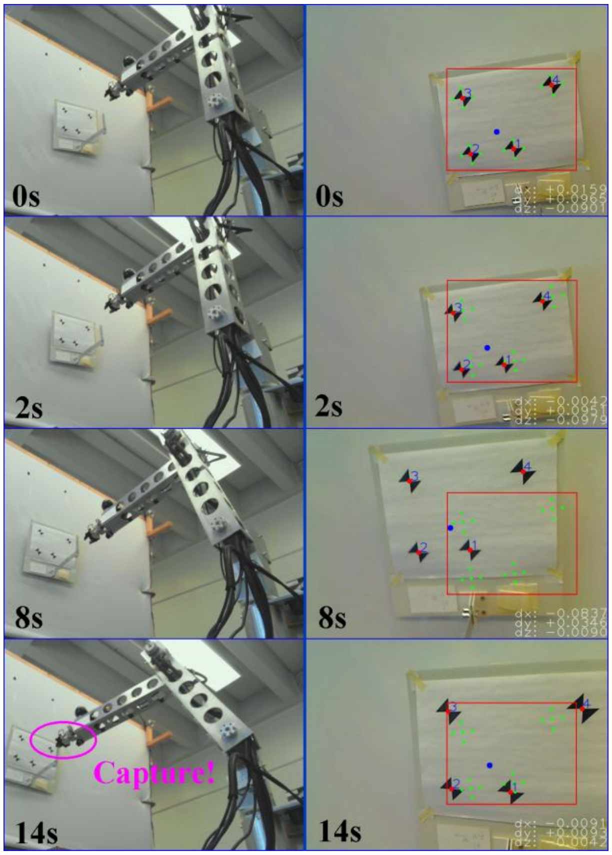


Fig. 7 Experimental capture process in case 1.

B. *Case 2*

Furthermore, another case of experiment has been performed with different home position initially. Time histories of the angular position of robotic manipulator and the tracking error are shown in Fig. 8 and Fig. 9, respectively. The performance of the proposed approach is similar to case 1, except it took a longer time to capture (about 34 s), because the initial tracking error is large.

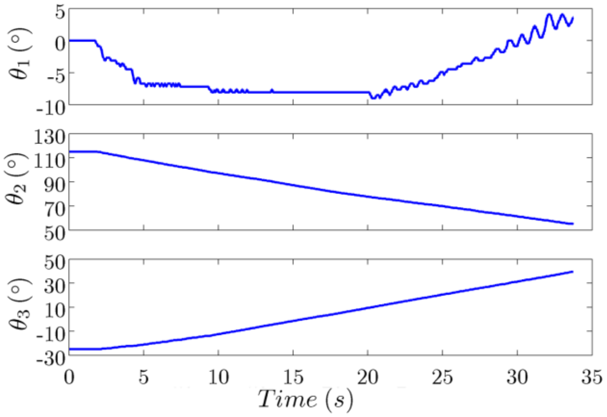


Fig. 8 Time history of angular position of the robotic manipulator in case 2.

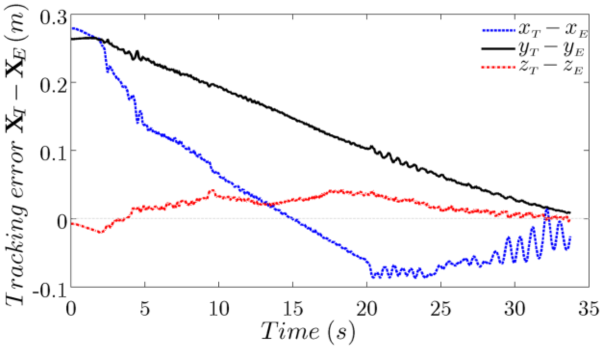


Fig. 9 Time history of tracking error in case 2.

The experimental results of both cases demonstrated the effectiveness of the proposed estimation algorithm and the kinematics based incremental control scheme.

5 Conclusion

This paper presents a kinematics-based incremental visual servo control strategy for the robotic capture of a non-cooperative target autonomously. An integrated algorithm of photogrammetry and adaptive extended Kalman filter is adopted to estimate the kinetic state of the non-cooperative target. A kinematics-based incremental control strategy is developed for the robotic manipulator to perform the autonomous capture. Based on the target estimates, the intercept point of the trajectories of the end-effector and the target, as well as the tracking time are dynamically predicted. The multiple solutions of inverse kinematics are avoided by considering the angular velocity limits of the actuators of robotic manipulator and the incremental angular position as the control input. Validating experiments were performed on a custom built robotic manipulator with the eye-in-hand configuration and a non-cooperative target. External jittering was added to the target motion in the experiments. The successful capture of the non-cooperative target demonstrated the effectiveness of the estimation algorithm and the incremental kinematic control strategy.

References

[1] K. Yoshida, "Achievements in Space Robotics Expanding the Horizons of Service and Exploration," *IEEE Robotics & Automation Magazine,* vol. 16, no. 4, pp. 20-28, Dec, 2009.

[2] D. J. Kim, Z. Wang, and A. Behal, "Motion Segmentation and Control Design for UCF-MANUS-An Intelligent Assistive Robotic Manipulator," *Ieee-Asme Transactions on Mechatronics,* vol. 17, no. 5, pp. 936-948, Oct, 2012.

[3] J. A. Smith, J. Jivraj, R. Wong, and V. Yang, "30 Years of Neurosurgical Robots: Review and Trends for Manipulators and Associated Navigational Systems," *Ann Biomed Eng*, pp. 1-11, Oct 14, 2015.

[4] R. M. Robinson, C. S. Kothera, R. M. Sanner, and N. M. Wereley, "Nonlinear Control of Robotic Manipulators Driven by Pneumatic

Artificial Muscles," *Ieee-Asme Transactions on Mechatronics,* vol. 21, no. 1, pp. 55-68, Feb, 2016.

[5] D. Hujic, E. A. Croft, G. Zak, R. G. Fenton, J. K. Mills, and B. Benhabib, "The robotic interception of moving objects in industrial settings: Strategy development and experiment," *Ieee-Asme Transactions on Mechatronics,* vol. 3, no. 3, pp. 225-239, Sep, 1998.

[6] R. A. McCourt, and C. W. de Silva, "Autonomous robotic capture of a satellite using constrained predictive control," *Ieee-Asme Transactions on Mechatronics,* vol. 11, no. 6, pp. 699-708, Dec, 2006.

[7] F. Aghili, "A Prediction and Motion-Planning Scheme for Visually Guided Robotic Capturing of Free-Floating Tumbling Objects With Uncertain Dynamics," *Ieee Transactions on Robotics,* vol. 28, no. 3, pp. 634-649, Jun, 2012.

[8] B. P. Larouche, and Z. H. Zhu, "Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control," *Autonomous Robots,* vol. 37, no. 2, pp. 157-167, Aug, 2014.

[9] M. H. Shan, J. Guo, and E. Gill, "Review and comparison of active space debris capturing and removal methods," *Progress in Aerospace Sciences,* vol. 80, pp. 18-32, Jan, 2016.

[10] R. Lampariello, "Motion planning for the on-orbit grasping of a non-cooperative target satellite with collision avoidance," in Int. Symp. Artif. Intell. Robot. Automat. Space, 2010, pp. 636-643.

[11] M. Jankovic, J. Paul, and F. Kirchner, "GNC architecture for autonomous robotic capture of a non-cooperative target: preliminary concept design," *Advances in Space Research*, 2015.

[12] G. Dong, and Z. H. Zhu, "Incremental Inverse Kinematics based Vision Servo for Autonomous Robotic Capture of Non-Cooperative Space Debris," *Advances in Space Research*, 2016.

[13] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, “A review of space robotics technologies for on-orbit servicing,” *Progress in Aerospace Sciences,* vol. 68, pp. 1-26, Jul, 2014.

[14] F. Janabi-Sharifi, and M. Marey, “A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing,” *Ieee Transactions on Robotics,* vol. 26, no. 5, pp. 939-947, Oct, 2010.

[15] S. Zimmermann, T. Tiemerding, and S. Fatikow, “Automated Robotic Manipulation of Individual Colloidal Particles Using Vision-Based Control,” *Ieee-Asme Transactions on Mechatronics,* vol. 20, no. 5, pp. 2031-2038, Oct, 2015.

[16] G. Dong, and Z. H. Zhu, “Position-based visual servo control of autonomous robotic manipulators,” *Acta Astronautica,* vol. 115, pp. 291-302, Oct-Nov, 2015.

[17] G. Dong, and Z. H. Zhu, “Autonomous Robotic Capture of Non-cooperative Target by Adaptive Extended Kalman Filter Based Visual Servo,” *Acta Astronautica*, 2016.

[18] F. Chaumette, and S. Hutchinson, “Visual servo control. I. Basic approaches,” *Robotics & Automation Magazine, IEEE,* vol. 13, no. 4, pp. 82-90, 2006.

[19] F. Chaumette, and S. Hutchinson, “Visual servo control - Part II: Advanced approaches,” *Ieee Robotics & Automation Magazine,* vol. 14, no. 1, pp. 109-118, Mar, 2007.

[20] K. A. Myers, and B. D. Tapley, “Adaptive Sequential Estimation with Unknown Noise Statistics,” *Ieee Transactions on Automatic Control,* vol. 21, no. 4, pp. 520-523, 1976.

[21] J. Denavit, “A kinematic notation for lower-pair mechanisms based on matrices,” *Trans. of the ASME. Journal of Applied Mechanics*, vol. 22, pp. 215-221, 1955.