sensors

MDPI

Article

Autonomous 6-DOF Manipulator Operation for Moving Target by a Capture and Placement Control System

Xiang Chen, Peilin Liu *, Rendong Ying and Fei Wen

School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University,
Shanghai 200240, China; chenxiang_hit@sjtu.edu.cn (X.C.); rdying@sjtu.edu.cn (R.Y.); wenfei@sjtu.edu.cn (F.W.)
* Correspondence: liupeilin@sjtu.edu.cn

Abstract: The robot control technology combined with a machine vision system provides a feasible method for the autonomous operation of moving target. However, designing an effective visual servo control system is a great challenge. For the autonomous operation of the objects moving on the pipeline, this article is dedicated to developing a capture and placement control system for the six degrees of freedom (6-DOF) manipulator equipped with an eye-in-hand camera. Firstly, a path planning strategy of online capture and offline placement is proposed for real-time capture and efficient placement. Subsequently, to achieve the fast, stable, and robust capture for a moving target, a position-based visual servo (PBVS) controller is developed by combining estimated velocity feedforward and refined PID control. Feedforward control is designed using the estimated velocity by a proposed motion estimation method for high response speed. PID control is refined by dead zone constraint to reduce the manipulator's jitter caused by the frequent adjustment of manipulator control system. Besides, the proportional, integral, and differential coefficients of PID controller are adaptively tuned by fuzzy control to reject the noise, disturbance, and dynamic variation in the capture process. Finally, validation experiments are performed on the constructed ROS–Gazebo simulation platform, demonstrating the effectiveness of the developed control system.

Keywords: autonomous operation; 6-DOF manipulator; capture and placement; velocity feedforward; refined PID

check for updates

Citation: Chen, X.; Liu, P.; Ying, R.; Wen, F. Autonomous 6-DOF Manipulator Operation for Moving Target by a Capture and Placement Control System. *Sensors* **2022**, *22*, 4836. https://doi.org/10.3390/s22134836

Academic Editor: Carlo Alberto Avizzano

Received: 24 May 2022
Accepted: 23 June 2022
Published: 26 June 2022

1. Introduction

With the development of artificial intelligence, the manipulator control technology combined with visual feedback has developed rapidly and attracted much attention in the fields of assembling parts, sorting materials, and capturing objects, etc. Especially, the autonomous operation for the objects moving on the pipeline in industrial production or warehousing logistics is a great challenge. The inaccurate or unstable operation will lead to high costs and even damage to manipulator [1–3]. On one hand, the precise knowledge of the moving target needs to be obtained by machine vision technology. On the other hand, an effective control system should be developed based on the feedback of visual system to adjust the manipulator's pose and motion. It is impossible to cover all the above deeply in a single article, so the issues related to control are mainly studied here.

Generally, visual servo controller can be classified as position-based, image-based, and hybrid system according to the errors used in a controller [4,5]. Position-based visual servo (PBVS) controller adjusts the error between the desired and actual pose and motion in 3D workspace directly. Image-based visual servo (IBVS) controls a manipulator by the deviation between the measured and target's position on 2D image plane. However, additional measurement is required because IBVS lacks the depth information. Although hybrid visual servo is effective, it is more complex than PBVS and IBVS. This will reduce the real-time performance of a manipulator system. Compared with IBVS and hybrid system, PBVS has a simpler control framework that allows direct control for end-effector's pose and motion, while the disadvantages are that moving target's information are susceptible to camera calibration and image noise. At present, these difficulties have been successfully solved by researchers, such as the image errors caused by the uncalibrated camera [6,7] and the image noise generated by camera vibration during manipulator operation [8].

Owing to the advantages of follow-up view and the natural relationship of the end-effector's pose relative to target, PBVS equipped with an eye-in-hand camera has attracted much attention from the researchers. A visual servo control scheme based on Kalman filter was proposed to automatically capture the moving target, which presented good robustness under noise and unexpected disturbance [9]. The dual Kalman filter scheme was developed to improve the tracking robustness and provide smooth motion estimation for the PBVS robotic control system [10]. A pose and motion estimation algorithm was proposed based on photogrammetry and extended Kalman filter, and the PBVS controller was devised based on PID algorithm to capture the non-cooperative target [11]. The inverse kinematics method incorporated trajectory planning was proposed based on the virtual repulsive torque theory. It was applied to design a PBVS system, and the manipulator's trajectory approaching target was spontaneously determined by the output control law [12]. However, due to the nonlinearity, time-varying parameters and model uncertainty of manipulator, and the high requirements for efficiency, accuracy, and robustness in practical engineering, the current control methods can hardly meet the control needs.

Some studies have been conducted to improve the performance of manipulator control system. Sliding mode controller was developed using the nonlinear model of manipulator for high robustness [13,14]. However, this controller was implemented by modeling the manipulator system and was hard to achieve precise control due to the nonlinear characteristics of manipulator. To address the drawback, the neural network-based controller is being studied, such as the back-propagation neural network used for reducing manipulator's tracking error [15], recurrent neural network controller [16,17], reinforcement learning neural network controller for the manipulator with unknown parameters and dead zones [18], hybrid neural network and sliding mode control for manipulator in dynamical environment [19]. Although neural network-based control theory obtains good results, how to find the optimal solution and achieve the fast convergence rate need to be solved in control engineering. In contrast, the controllers combining PID and other optimization or control methods are widely used in nearly 90% of industrial control systems due to their advantages of simple framework, good stability, and high reliability [20–22]. The whale optimizer algorithm is applied to adjust the parameters of PID controller, obtaining less settling time and ITAE error during the trajectory tracking control of 2-DOF robot manipulator [23]. A fuzzy PID controller was used to design the kinematics control strategy of cable-driven snake-like manipulator, realizing the precise control [24]. The following error will increase when inputting rapidly changed signals for the PID-based controller. To overcome this drawback, the feedforward plus PID controller was developed and achieved better tracking results and less tuning time in the comparative experiments [25]. The control system combined feed-forward, and fuzzy PID controller was proposed to control a planar parallel manipulator, which showed good adaptability to the disturbances and dynamic variations [26].

Based on the above, it can be concluded that two problems need to be addressed in order to design an effective autonomous operation control system for the moving target on pipeline. One is how to raise the overall operation efficiency, and the other is how to improve the response speed, stability, and robustness for capturing the moving target. In this paper, a capture and placement control system is developed, which is as follows: a path planning strategy of online capture and offline displacement is proposed to improve the overall operation efficiency; a new PBVS controller is developed to improve the capture performance by combining the estimated velocity feedforward and refined PID controller. The motion estimation method combined Kalman filter and interpolation operation is proposed to achieve accurate and smooth estimation for target's pose and motion. Therefore, it provides feedforward velocity to the controller for a high response speed. The refined PID controller is designed, by the dead zone constraint, to reduce the manipulator's jitter caused by controller's frequent adjustment, and by fuzzy control to adaptively tune PID coefficients for rejecting noise or external disturbance. Finally, the developed capture and placement control system is experimentally verified by the constructed ROS–Gazebo simulation platform.

2. ROS–Gazebo Simulation Platform

To facilitate the development of capture and placement control system, a simulation platform was built by combining robot operating system (ROS) and Gazebo simulator. All the experiments were performed on this platform. ROS is an open-source meta-operating system that provides many services for robots, including hardware abstraction, underlying device control, implementation of common functions, inter-process messaging, and package management [27]. It can be used to drive real or simulated robots to perform various tasks. Gazebo is a mature 3D physical simulator that can be closely combined with ROS for simulation [28]. To obtain high simulation accuracy, Gazebo provides various underlying physics engine such as Ode, Bullet, Dart, and Simbody. Besides, it has rich interfaces for modeling, control, and visualization. In the virtual visual simulation environment, the robot model with physical properties can be created, and the robot motion and sensing data can also be simulated. Compared with the direct hardware operation, the simulation platform based on ROS and Gazebo is more suitable for the development and test of complex system due to the convenient parameter adjustment, fast operation speed, and low cost. The simulation framework of ROS–Gazebo is shown in Figure 1a, including the following processes:

(1) Gazebo updates the model state information in real time by the simulation model plugin;
(2) ROS control layer regularly obtains and calculates the state information from the simulation model;
(3) The behavior algorithm obtains the calculated state information and performs behavior derivation calculation;
(4) The node of generating control instruction obtains the behavior information and generates the control instructions;
(5) Gazebo simulation plugin gets control instructions to control the motion of simulation model.

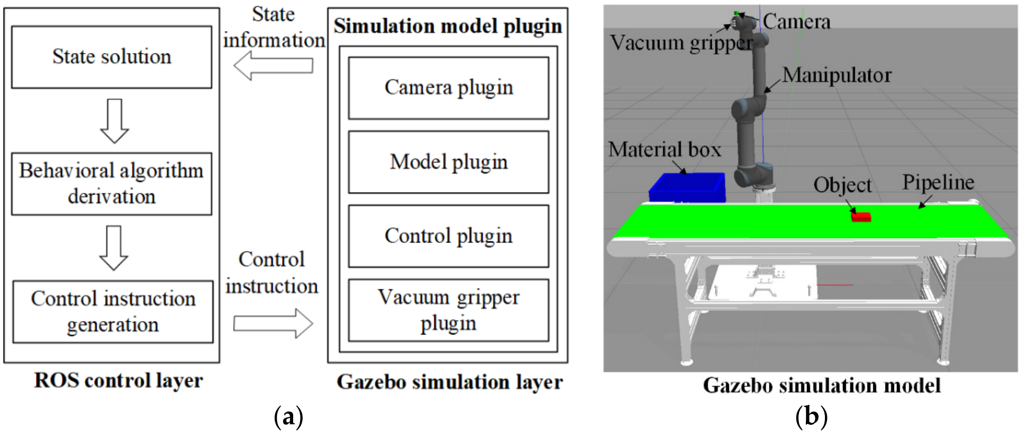


Figure 1. Simulation platform based on ROS–Gazebo. (**a**) Simulation framework. (**b**) Simulation model.

The robot simulation model was built as presented in Figure 1b. It mainly consists of a 6-DOF manipulator equipped with an eye-in-hand camera and vacuum gripper, the object moving on pipeline and a material box. Thereinto, UR5 with 0.85 m operating radius and +/−0.1 mm repeatable accuracy was selected as the manipulator model. Kinect V1 with 640 × 4180 resolution and 30 fps are selected as the camera model. The accurate and efficient simulation for the capture and placement of the moving target can be achieved by Gazebo simulator due to its powerful physics engine. Thereinto, the vacuum gripper will perform capture when the tracking error of manipulator end-effector relative to target is within a certain range. The visual servo control program is developed by C++ and Python in ROS control layer.

3. Kinematic Model of 6-DOF Manipulator Capturing Moving Target

The position-based visual seroving manipulator with an eye-in-hand camera provides an effective method for the autonomous operation of moving target, due to its advantages of follow-up view and natural relationship of end-effector's pose with respect to target. In the operation process, the velocity of moving target may be unknown or changed dynamically. It is required to track the target in real time based on the feedback of the machine vision system. Besides, soft grasping is preferred for avoiding the damage to moving target. Therefore, not only the pose but also the velocity of manipulator end-effector should be consistent with those of moving targets. It is necessary to analyze the kinematic relationship between the manipulator and moving target. In this paper, the autonomous operation for the target moving on pipeline is conducted by a 6-DOF manipulator considering its high flexibility, as shown in Figure 2.

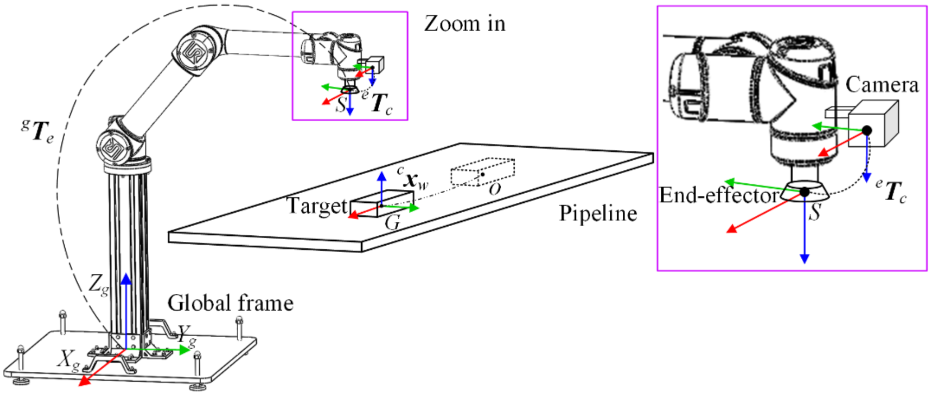


Figure 2. Schematic diagram of capturing target by the visual servoing manipulator.

The camera was mounted next to the end-effector of manipulator and used to measure target's pose ${}^{c}\boldsymbol{x}_w$ relative to camera frame in real time. The calibrations, including the intrinsic parameters for camera, the extrinsic parameters for hand–eye, i.e., the transformation matrix ${}^{e}\boldsymbol{T}_c$ from camera to end-effector frame (denoted by dashed line in Figure 2) and the model parameters for manipulator, i.e., the transformation matrix ${}^{g}\boldsymbol{T}_e$ from end-effector to global frame (marked with dotted line in Figure 2), were carried out in advance. The manipulator has six revolute joints. The first three, including shoulder pan joint, shoulder lift joint, and elbow joint, were used to control the position of manipulator end-effector. The last three include wrist-1 joint, wrist-2 joint, and wrist-3 joint for controlling the orientation of manipulator end-effector. During the autonomous operation, manipulator end-effector starts from the fixed initial point S and tracks the target moving from the point O on pipeline. When the tracking error of manipulator end-effector from moving target is within the allowed range, the capture operation will be performed at a certain grasping point G. It should be noted that point G is not a fixed point, but the one that meets the requirement of the allowed tracking error. A double dashed line in Figure 2 presents the motion trajectory of the moving target.

The kinematics of manipulator presents the transformation relationship from joint space to workspace as expressed below:

$$ {}^{g}\boldsymbol{x}_e = {}^{g}\boldsymbol{T}_e(\boldsymbol{\theta}) \tag{1}$$

where ${}^{g}\boldsymbol{x}_{e}$ denotes the end-effector's pose in a global frame, i.e., the base coordinate system of manipulator, and $\boldsymbol{\theta}= (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ represents the angle of six joints in joint space. The forward kinematics model of the manipulator is given in Appendix A. Target's pose ${}^{g}\boldsymbol{x}_{w}$ in global frame can be obtained according to the pose ${}^{c}\boldsymbol{x}_{w}$ in camera frame by:

$$ {}^{g}\boldsymbol{x}_{w} = {}^{g}\boldsymbol{T}_{e}(\boldsymbol{\theta}){}^{e}\boldsymbol{T}_{c}{}^{c}\boldsymbol{x}_{w} \tag{2}$$

The purpose of PBVS controller is to minimize the error between the desired and the actual pose of manipulator end-effector, which is expressed as:

$$ \boldsymbol{e}(k) = {}^{g}\boldsymbol{x}_{w}(k) - {}^{g}\boldsymbol{x}_{e}(k) \tag{3}$$

where $\boldsymbol{e}(k)$ denotes the error in the k-th control cycle, including the position and orientation error; ${}^{g}\boldsymbol{x}_{w}(k)$ and ${}^{g}\boldsymbol{x}_{e}(k)$ are the estimated pose obtained by the proposed motion estimation method, which will be presented in Section 4.2.1, and the actual pose of end-effector in the k-th control cycle, respectively.

While minimizing the error, manipulator end-effector moves at the same velocity as the moving target, which facilitates the soft capture. In this case, the velocity control mode of manipulator is preferred for the real time tracking of the moving target, thus the output of the developed controller should be defined as the end-effector's velocity in workspace.

During the actual control of robotic manipulator, end-effector's motion speed in workspace should be transformed to joint space. Assume that in the k-th control cycle, ${}^{g}\boldsymbol{v}_{e}(k)$ is the end-effector's velocity in global frame and it is output by the designed controller, including linear velocity ${}^{g}\boldsymbol{v}_{le}(k)$ and angular velocity ${}^{g}\boldsymbol{v}_{ae}(k)$ expressed as:

$$ {}^{g}\boldsymbol{v}_{e}(k)= [{}^{g}\boldsymbol{v}_{le}(k), {}^{g}\boldsymbol{v}_{ae}(k)]^{T} \tag{4}$$

Both the linear and angular velocity are limited to the maximum velocity range. Finally, instantaneous inverse kinematic are performed, as given in Appendix B to obtain the desired joint velocity as:

$$ \boldsymbol{\omega}(k) = \boldsymbol{J}^{-1}{}^{g}\boldsymbol{v}_{e}(k) \tag{5}$$

where $\boldsymbol{\omega}(k)= \{\omega_1(k), \omega_2(k), \omega_3(k), \omega_4(k), \omega_5(k), \omega_6(k)\}$ is the joint angular velocity and $\boldsymbol{J}^{-1}$ is the pseudo inverse of Jacobian matrix. Based on the obtained joint angular velocity, the angle of each joint can be expressed as:

$$ \theta_i(k) = \theta_i(k-1) + \omega_i(k)T, i= 0, 1, 2, \ldots\ldots, 6 \tag{6}$$

where $\theta_i(k)$ is the control angle of the i-th joint in the k-th control cycle and T is the control period. The angles and angular velocities of all joints are finally input to the underlying controller of the manipulator to perform tracking and capture.

4. Capture and Placement Control System

A capture and placement control system was developed for the 6-DOF manipulator to autonomously operate the moving target on pipeline. This control system consists of two parts: an operation strategy of online capture and offline placement is proposed to improve the overall operation efficiency; a new PBVS controller is developed based on estimated velocity feedforward and refined PID control. Details are presented below.

4.1. Capture and Placement Strategy

During the capture process of the target moving on pipeline in industrial production or warehouse logistics, the target's velocity is usually unknown or dynamically changed. Therefore, the robotic manipulator is required to track the target in real time. At the placement stage, the placed pose of target is usually fixed, and thus the motion path of manipulator end-effector can be planned offline using the predefined waypoints. According to the requirement for system operation, a capture and placement strategy is proposed as shown in Figure 3. Manipulator end-effector starts tracking the moving target from the initial pose when it appears in the camera's field of view. Subsequently, the grasping operation is performed when the tracking error between end-effector and moving target is within the allowed error threshold. The motion path of the above capture process is online, planned based on the visual feedback, and the velocity control mode of manipulator is adopted for high real-time performance. The velocity control signal of manipulator capturing the moving target is output by the developed controller, which is mainly studied in this paper.

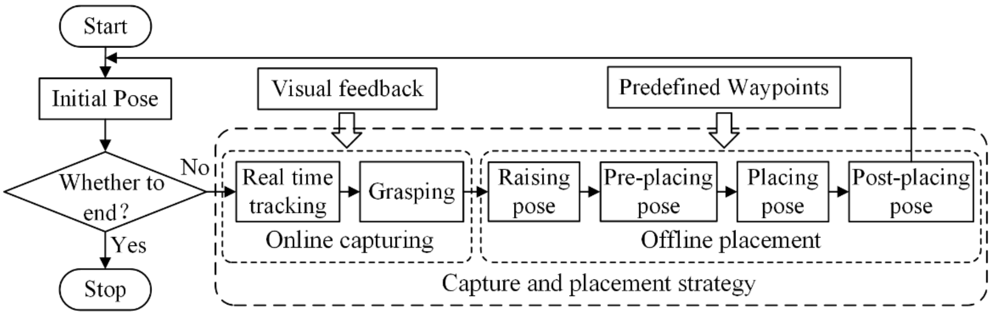


Figure 3. Flowchart of capture and placement strategy for the moving object.

The capture process will cost large amounts of computing resource by the machine vision and manipulator control system. However, the motion path of placement process can be offline planned based on the predefined waypoints, including pre-placing pose, placing pose, and post-placing pose, for reducing the computing cost. The schematic diagram of offline placement is shown in Figure 4. The target's pose ${}^{g}\boldsymbol{x}_w$ in global frame is obtained by the proposed motion estimation method, and it will be given in Section 4.2.1. The raising pose $\boldsymbol{x}_{rai}$ can be determined only by changing the z axis coordinate value of target's pose. The placing pose of all targets in material box is predetermined according to the size of material box and target. In Figure 4, $\boldsymbol{w}_i$ is the placing pose of the i-th target. At the same time, the pre-placing pose $\boldsymbol{w}_{pre}$ can be determined only by modifying the z axis coordinate value of placing pose. The joint angle control signal for placement operation can be obtained by inverse kinematics based on the pose of the key waypoints. As a result, the placement process avoids the online calculation, which helps to improve the operation efficiency.

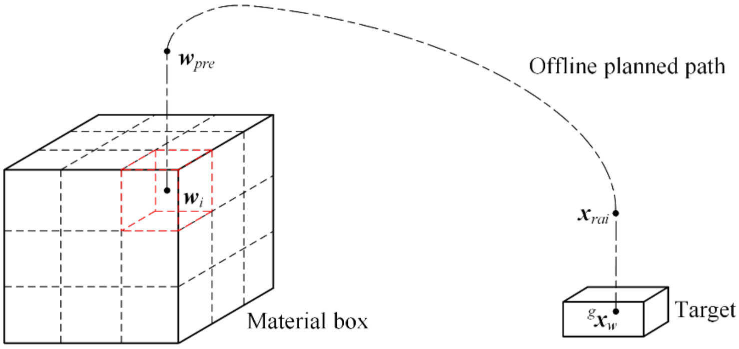


Figure 4. Schematic diagram of offline planning placement path.

4.2. Capture Control Scheme

Although control theory has developed rapidly, PID control strategy is still widely used in the industrial control systems due to its advantages of simple control structure, good stability, and high reliability. To track the moving target in real-time, it is required to have high dynamic response speed for the controller. Using the large proportional coefficient can improve the dynamic response of PID controller, but it easily causes excessive torque in some joints, leading the manipulator to stop working or even suffer damage.

Besides, the target's velocity may be changed by external disturbance during the tracking process. To achieve the fast, stable, and robust capture for the moving target, a new PBVS control scheme is developed, as shown in Figure 5. For high response speed, the velocity feedforward control is designed based on the target's velocity estimated by the proposed motion estimation method. For PID closed-loop control, the target's pose estimated by the motion estimation method is input as the desired value and the actual one of manipulator end-effector is used as the feedback value. What is more, PID controller is improved by dead zone module to reduce manipulator's jitter caused by the frequent adjustment of controller, and by fuzzy control module to adjust PID parameters online for rejecting the noise and disturbances. The feedforward velocity is combined with the output of the improved PID controller, which is used as the end-effector's velocity $\boldsymbol{V}$. Finally, the joint velocity $\boldsymbol{\omega}$ and angle $\boldsymbol{\theta}$ can be obtained to conduct the capture operation. Details are as follows.

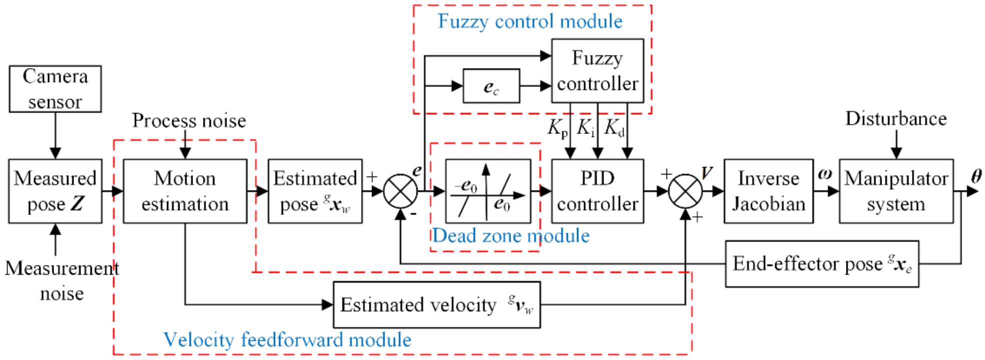


Figure 5. Block diagram of feedforward PID controller with dead zone and fuzzy control module.

4.2.1. Feedforward Control by the Motion Estimation Method

The velocity of moving target on pipeline may be unknown or variable in the dynamic environment. Besides, the control instructions sent to manipulator are generally required at 50 Hz frequency. However, the real-time performance of issuing control commands is directly affected by machine vision system. According to the literature [29], most of the current machine vision system is time-consuming, even using the camera with high sampling frequency. As a result, the update frequency of control instructions is lower than the required one, which will cause the manipulator to move unevenly. To address the above problems, a motion estimation method combined Kalman filter and interpolation operation is proposed as shown in Figure 6. The target's pose $\boldsymbol{Z}(k)$ measured by machine vision system is input to perform motion estimation calculation, where k represents the current k-th cycle, and the other symbols in this section have the same meanings as those in Section 3. According to the transformation from camera to global frame and the back projection based on pinhole imaging principle, the measured pose $\boldsymbol{Z}(k)$ of target in the global frame can be obtained by:

$$\boldsymbol{Z}(k) = {}^g\boldsymbol{T}_e {}^e\boldsymbol{T}_c \boldsymbol{K}^{-1} \boldsymbol{U}(k) s(k) \tag{7}$$

where $s(k)$ is the distance from the optical center of camera to target. $\boldsymbol{U}(k)$ is the homogeneous coordinate value of target in pixel frame. $\boldsymbol{K}^{-1}$ is the inverse matrix of camera intrinsic parameter matrix. ${}^e\boldsymbol{T}_c$ and ${}^g\boldsymbol{T}_e$ are the transformation matrix from camera to end-effector frame and from end-effector to global frame, respectively.

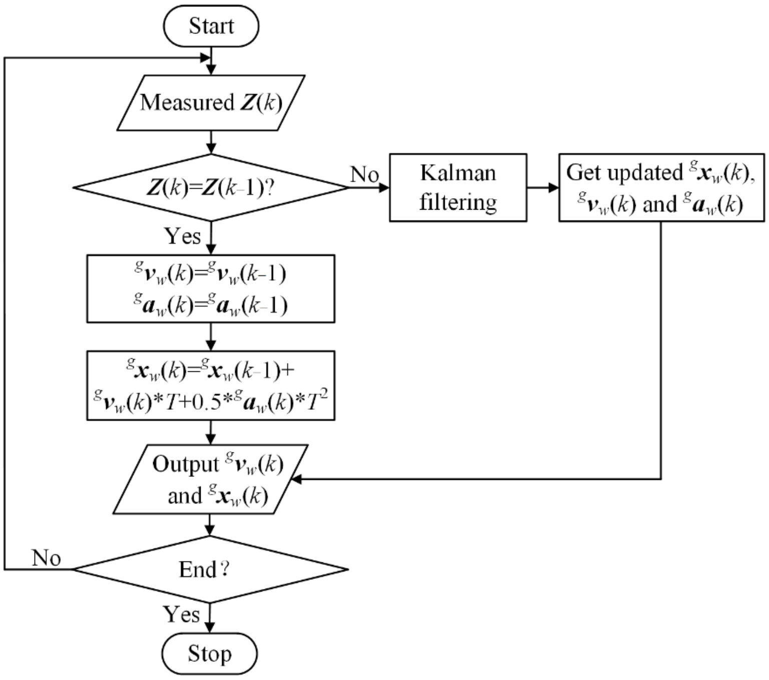

Start
Measured $Z(k)$
$Z(k)=Z(k-1)$?
No
Kalman filtering
Get updated ${}^g x_w(k)$, ${}^g v_w(k)$ and ${}^g a_w(k)$
Yes
${}^g v_w(k)={}^g v_w(k-1)$
${}^g a_w(k)={}^g a_w(k-1)$
${}^g x_w(k)={}^g x_w(k-1)+{}^g v_w(k)*T+0.5*{}^g a_w(k)*T^2$
Output ${}^g v_w(k)$ and ${}^g x_w(k)$
No
End?
Yes
Stop

Figure 6. Flow chart of the proposed motion estimation method.

For the condition that $\mathbf{Z}(k)$ is different from the previous $\mathbf{Z}(k-1)$, the Kalman filter is used to predict the upcoming measurement result. It computes the current estimation based on all past measurements and usually converges in a few iterations. Thus, the motion including velocity ${}^g\boldsymbol{v}_w(k)$, acceleration ${}^g\boldsymbol{a}_w(k)$, and pose ${}^g\boldsymbol{x}_w(k)$ can be estimated by the same process, as described in literature [9]. One example of calculating the system state along x axis in global frame by Kalman filter method is given in Appendix C.

However, $\mathbf{Z}(k)$ is usually not updated in time because the actual output frequency of machine vision system is lower than the required signal release frequency. As a result, the current sampled pose $\mathbf{Z}(k)$ is equal with the previous $\mathbf{Z}(k-1)$. In this case, the interpolation operation will be conducted using the predicted velocity, acceleration, and position by $(k-1)$-th Kalman filtering to estimate k-th motion, as expressed in Equation (8). From k-th to $(k+i-1)$-th cycle, the interpolation operation is recursively performed, until receiving the updated measurement $\mathbf{Z}(k+i)$. At this time, the interpolated results in $(k+i-1)$-th cycle are applied to Kalman filtering process. The obtained velocity and position will be used as the feedforward value and the input for PID controller, respectively.

$$\begin{cases} {}^g\boldsymbol{v}_w(k) = {}^g\boldsymbol{v}_w(k-1) \\ {}^g\boldsymbol{a}_w(k) = {}^g\boldsymbol{a}_w(k-1) \\ {}^g\boldsymbol{x}_w(k) = {}^g\boldsymbol{x}_w(k-1) + {}^g\boldsymbol{v}_w(k) * T + 0.5 * {}^g\boldsymbol{a}_w(k) * T^2 \end{cases} \quad (8)$$

Under the condition that the camera keeps stationary and the target moves on the pipeline plane, the comparative experiments are performed, in which all experimental data are represented in global coordinate system. Figure 7 gives target's position measured by machine vision system based on the photogrammetry method and the ones by original Kalman filter and proposed motion estimation method. It is noted in Figure 7 that the measured data are composed of horizontal lines and polylines, which have the ladder-line shape. This discontinuity of measured data is caused by the update lag of target location, which will lead to the unstable movement of the manipulator. For the result obtained by the proposed method, phase a denotes the convergence process from the initial value to the exact one, and phase b indicates that the interpolated position achieves the stable and accurate estimation. The result estimated by original Kalman filter has the same adjustment process as the above, but it has poor smoothness and a larger difference with the measured result. According to the results, it can be concluded that the proposed method provides more smooth and accurate estimation for velocity feedforward and PID control.

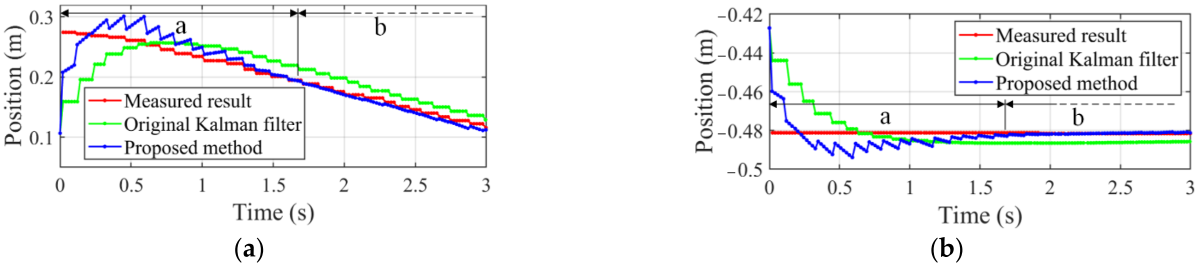


Figure 7. Measured result and estimated ones by original Kalman filter and proposed method. (**a**) Position along x axis. (**b**) Position along y axis.

4.2.2. PID Controller with Dead Zone Constraint

In this paper, PID controller is used for adjusting the error $\boldsymbol{e}(k)$ between the desired and actual pose of manipulator end-effector in steady-state tracking process. The estimated pose ${}^{g}\boldsymbol{x}_w(k)$ in Section 4.2.1 is the desired one and used as the input of PID controller. The current pose $\boldsymbol{F}(k)$ of manipulator end-effector is served as the feedback value of PID controller. The output $\boldsymbol{V}_{PID}(k)$ of PID controller is defined as the end-effector's velocity to reduce the steady-state error. It is expressed as:

$$\boldsymbol{V}_{PID}(k) = K_p\boldsymbol{e}(k) + K_i\sum_{n=0}^{k}\boldsymbol{e}(k) + K_d[\boldsymbol{e}(k) - \boldsymbol{e}(k-1)] \tag{9}$$

In which $\boldsymbol{e}(k) = \boldsymbol{X}(k) - \boldsymbol{F}(k)$; K_p, K_i, and K_d denote the proportional, intergral, and differential coefficients of PID controller, respectively. The output of PID controller is used to correct the tracking velocity of end-effector in small range, and the large part of tracking velocity is adjusted by the feedforward value obtained by the improved Kalman filter. Therefore, the control velocity of manipulator end-effector can be expressed as:

$$\boldsymbol{V}(k) = \boldsymbol{V}_{PID}(k) + {}^{g}\boldsymbol{v}_w(k) \tag{10}$$

where $\boldsymbol{V}_{PID}(k)$ and ${}^{g}\boldsymbol{v}_w(k)$ denote the output of PID controller and feedforward value, respectively.

Generally, PID controller will continuously adjust the deviation until it disappears. However, the deviation always exists due to the influence of the manipulator model error and the coordinate transformation error between the camera and global frame. As a result, PID controller will perform frequent adjustments for the fine error when manipulator end-effector approaches the target, which leads to the jitter of manipulator. To solve this problem, the deviation between the desired and feedback pose is pre-processed by the dead zone constraint. If the deviation is too large so that it exceeds the threshold of dead zone constraint, it will stay unchanged to enable the operation of PID controller. When the deviation is small enough, it will be set to zero and the PID controller will stop working. The dead zone constraint and control signal are expressed as Equations (11) and (12), respectively.

$$\boldsymbol{e}(k) = \begin{cases} \boldsymbol{e}(k) & |\boldsymbol{e}(k)| > |\boldsymbol{e}_0| \\ 0 & |\boldsymbol{e}(k)| \le |\boldsymbol{e}_0| \end{cases} \tag{11}$$

$$\boldsymbol{V}(k) = \begin{cases} \boldsymbol{V}_{PID}(k) + {}^{g}\boldsymbol{v}_w(k) & |\boldsymbol{e}(k)| > |\boldsymbol{e}_0| \\ {}^{g}\boldsymbol{v}_w(k) & |\boldsymbol{e}(k)| \le |\boldsymbol{e}_0| \end{cases} \tag{12}$$

where $\boldsymbol{e}_0$ denotes the threshold of dead zone constraint. It is noted that the setting for the dead zone value is very important. The small value has little effect on reducing manipulator's jitter. On the contrary, it will cause large control error. In this paper, the dead zone threshold with 0.1 mm is determined according to the control accuracy of manipulator system and the results of pre-experiments.

Specifically, a step response experiment was conducted to analyze the stability of PID controller with dead zone constraint (DPID). The results of the DPID controller and that of the general PID controller for comparison are shown in Figure 8. The desired positions of end-effector in global frame are set at 0.13 s, as marked by the black dotted lines. It can be noted that the adjustment time of DPID controller is smaller than that of the general one. Besides, the motion trajectory obtained by DPID controller is also smoother in the adjustment process, indicating that DPID controller benefits by reducing the jitter of manipulator. This is owed to the DPID controller avoiding frequent adjustment under the dead zone constraint.

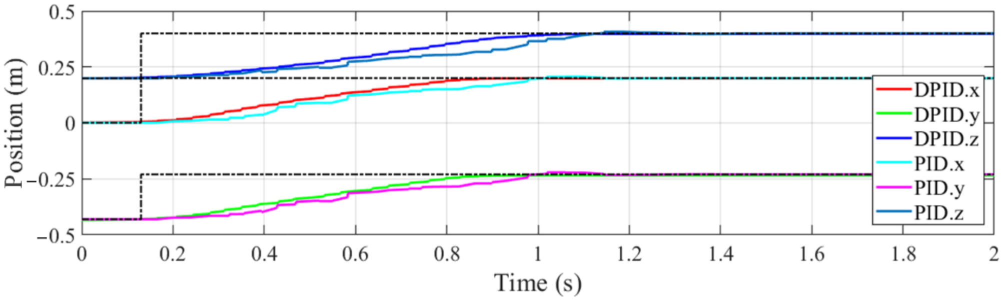


Figure 8. Step response of the position of manipulator end-effector.

4.2.3. Online Tuning of PID Coefficients

In the presence of external disturbance, random noise, and model's uncertainty, it is hard to track the moving target accurately by the general PID controller [9]. According to the demand of system operation, fuzzy control is used to modify PID controller in this paper, considering its advantages of independent mathematical model, imitating human logical thinking, and strong robustness. According to the general PID controller as expressed in Equation (9), three parameters, K_p, K_i, and K_d, should be tuned by fuzzy tuners. The detailed structure is shown in Figure 5 for the fuzzy control module. There are two inputs for fuzzy controller: the error $\boldsymbol{e}$ between the pose estimated by the improved Kalman filter and the feedback one of end-effector, and its change rate $\boldsymbol{e_c}$. According to fuzzy set rules, the modified values ΔK_p, ΔK_i, and ΔK_d are adjusted on the basis of the initial values K_{po}, K_{io}, and K_{do}. Finally, the fuzzy control module dynamically outputs the changed PID parameters K_p, K_i, and K_d under different operation conditions.

In fuzzy control module, fuzzification is firstly conducted, namely the above five variables are transformed to seven fuzzy linguistic parameters NB (negative big), NM (Negative medium), NS (Negative small), Z (zero), PS (Positive small), PM (Positive medium), PB (Positive big). The corresponding domain of the input and output variables are defined as $\{-6, -4, -2, 0, 2, 4, 6\}$ and $\{-3, -2, -1, 0, 1, 2, 3\}$, respectively. Then, the membership functions (MFs) with triangular distribution are adopted for all variables, and the membership value can be obtained based on the MFs. Considering the demand for accuracy, stability, and overshoot of control system, the fuzzy rules to adjust the PID parameters are determine by the knowledge of the practical experiments, as shown in Table 1. For the large $\boldsymbol{e}$ and $\boldsymbol{e_c}$, larger K_p, K_i, and K_d should be taken to achieve fast tracking. For the medium $\boldsymbol{e}$, small K_p is used to reduce the overshoot, and K_i should be appropriately selected because it has a significant impact on system response. When $\boldsymbol{e}$ is small, small K_p and medium K_i are taken to avoid overshoot and reduce steady-state error. For large $\boldsymbol{e_c}$, the large value of K_d should be taken to obtain predictive compensation.

Table 1. Fuzzy rules of online tuning (ΔK_p, ΔK_i, and ΔK_d).

e \ e_c	**NB**	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
NB	(PB, NB, PS)	(PB, NB, NS)	(PM, NM, NB)	(PM, NM, NB)	(PS, NS, NB)	(Z, Z, NM)	(Z, Z, PS)
NM	(PB, NB, PS)	(PB, NB, NS)	(PM, NM, NB)	(PS, NS, NM)	(PS, NS, NM)	(Z, Z, NS)	(NS, Z, Z)
NS	(PM, NB, Z)	(PM, NM, NS)	(PM, NS, NM)	(PS, NS, NM)	(Z, Z, NS)	(NS, PS, NS)	(NS, PS, Z)
Z	(PM, NM, Z)	(PM, NM, NS)	(PS, NS, NS)	(Z, Z, NS)	(NS, PS, NS)	(NM, PM, NS)	(NM, PM, Z)
PS	(PS, NM, Z)	(PS, NS, Z)	(Z, Z, Z)	(NS, PS, Z)	(NS, PS, Z)	(NM, PM, Z)	(NM, PB, Z)
PM	(PS, Z, PB)	(Z, Z, NS)	(NS, PS, PS)	(NM, PS, PS)	(NM, PM, PS)	(NM, PB, PS)	(NB, PB, PB)
PB	(Z, Z, PB)	(Z, Z, PM)	(NM, PS, PM)	(NM, PM, PM)	(NM, PM, PS)	(NB, PB, PS)	(NB, PB, PB)

According to the above setting and fuzzy rules, the output surface of ΔK_p, ΔK_i, and ΔK_d on the domain are obtained as presented in Figure 9. The modified parameters ΔK_p, ΔK_i, and ΔK_d can be obtained by checking Table 1. Centroid defuzzification method is used to calculate the specific variation of PID parameters from the obtained fuzzy results. Finally, the self-tuning PID parameters are determined by Equation (13).

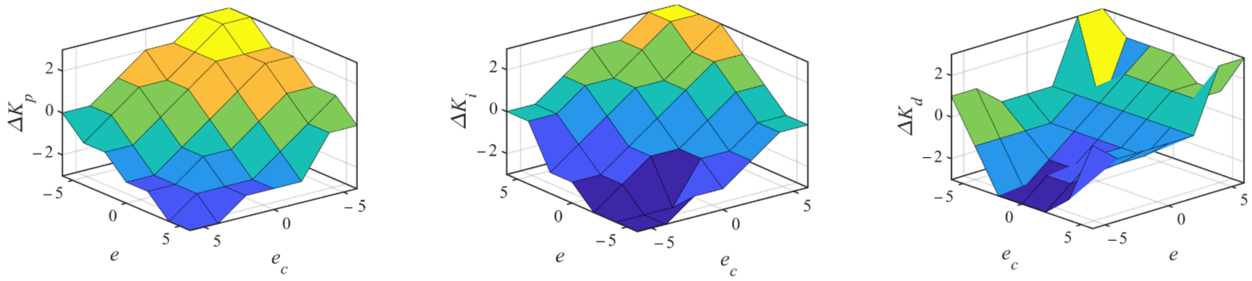


Figure 9. Output surface for self-tuning ΔK_p, ΔK_i, and ΔK_d.

$$\begin{cases} K_p(k) = K_{po} + \Delta K_p(k) \\ K_i(k) \ = \ K_{io} + \Delta K_i(k) \\ K_d(k) = K_{do} + \Delta K_d(k) \end{cases} \tag{13}$$

where $\Delta K_p(k)$, $\Delta K_i(k)$, and $\Delta K_d(k)$ are the defuzzified variation of proportional, integral, and differential coefficient in the k-th control. The initial parameters, including K_{po} with 28, K_{io} with 0.008, and K_{do} with 0.0015, are offline determined by GA-II optimization algorithm, as described in the literature [30].

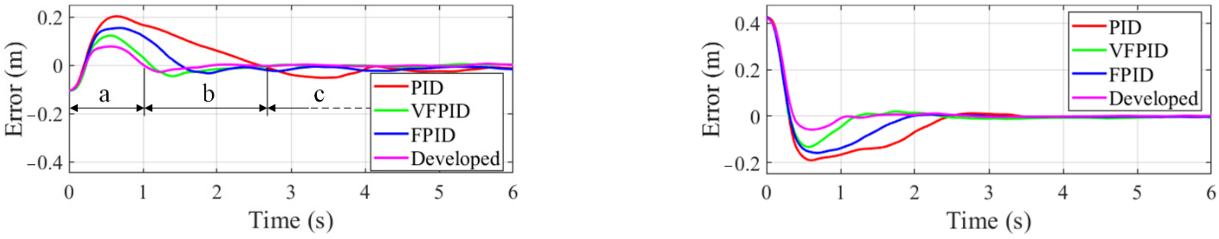
5. Experimental Results and Discussions

In this section, the experiments are performed on the simulation platform to test the dynamic response speed and robustness of the developed controller. Meanwhile, the autonomous operation experiment is conducted to verify the effectiveness of the proposed control system. The experimental scenario is the same as described in Section 3. The pose of manipulator end-effector and moving target are all represented in the global coordinate system, and the tracking error is the difference between the manipulator end-effector and moving target, as expressed in Equation (3).

5.1. Dynamic Response Speed Analysis

To test the dynamic response speed of the developed controller, experiments were performed under two conditions of tracking the objects moving at high-speed with 1 m/s and low-speed with 0.6 m/s along x-axis, respectively. The results obtained by the developed controller were compared with those obtained by conventional PID, velocity feedforward PID (VFPID), and fuzzy PID (FPID).

Firstly, the tracking error of manipulator end-effector for the low-speed moving target is shown in Figure 10. Taking the results obtained by the developed controller as an example, phase a featured with large changes in tracking error is the transition process of end-effector from the initial point to the near region of the moving target. Phase b denotes that end-effector approaches the near region of the moving target, which is characterized by a small variation in tracking error. Phase c indicates that the stable tracking is achieved, and the capture operation can be performed. From the results in Figure 10a,b, it can be observed that the FPID controller takes less time to reach the stable tracking stage than PID controller due to using the adaptively tuned PID parameters. VFPID controller has a faster response speed than PID and FPID controllers. Compared to the above controllers, the developed controller takes the least time to approach the stable tracking stage. The detailed time spent in phase a and b are presented in Table 2.



(a)

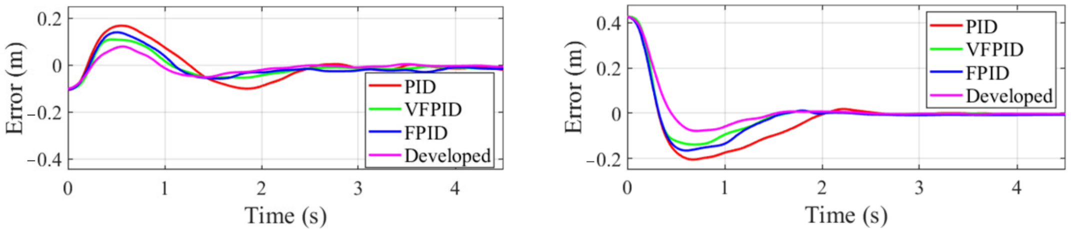
(b)

Figure 10. Tracking error of manipulator end effector to low-speed moving target. (**a**) Error along *x* axis. (**b**) Error along *y* axis.

Table 2. The time spent in phase a and b for low-speed moving target.

	PID	VFPID	FPID	Developed
Phase a (s)	2.573	1.098	1.495	1.010
Phase b (s)	5.752	3.121	4.930	2.852

Besides, a comparative experiment is also carried out by tracking the high-speed moving object. Figure 11 presents the tracking error under different controllers and the same conclusion can be obtained as the above analysis. Table 3 presents the time spent in phase a and b for the high-speed moving target. It is noted that all the controllers in Table 3 take less time in phase a and b than that in Table 2. That is because the object moves at high speed, which will result in larger tracking error in the tracking process. The output of all the controllers will increase, thus reducing the adjustment time.



(a)

(b)

Figure 11. Tracking error of manipulator end effector to high-speed moving object. (**a**) Error along *x* axis. (**b**) Error along *y* axis.

Table 3. The time spent in phase a and b for high-speed moving target.

	PID	VFPID	FPID	Developed
Phase a (s)	1.253	1.040	1.101	0.932
Phase b (s)	4.243	2.824	3.982	2.487

5.2. Robustness Test under Gauss Noise and Disturbance

To examine the robustness of the developed controller, experiments are performed for the first time changing the target's position by introducing large Gaussian noise in the measurement equation of Kalman filter, and for the second time introducing an angle disturbance to six joints, as shown in Figure 5. Gaussian noise with zero-mean value and 0.05 covariance is used as the measurement noise from camera. The target's position along x and y axis on pipeline in the tracking process using the developed controller is shown in Figure 12.

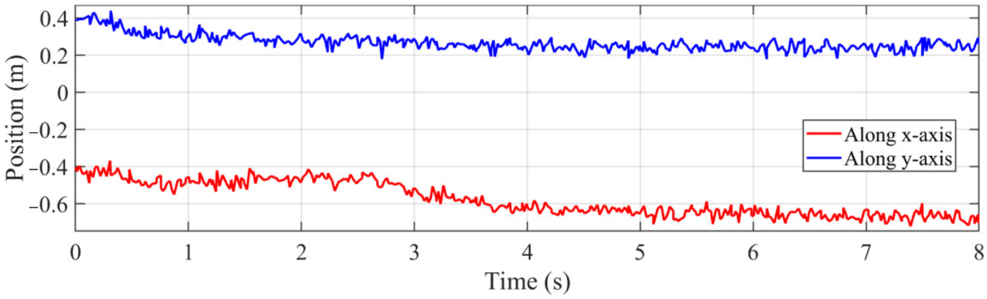


Figure 12. Target's position obtained by the proposed estimation method with Gaussian noise.

The tracking error from the initial pose of manipulator end-effector to the stable tracking phase is shown in Figure 13. It can be seen that the proposed VFPIDDF controller has better robustness and a higher response speed than other controllers.

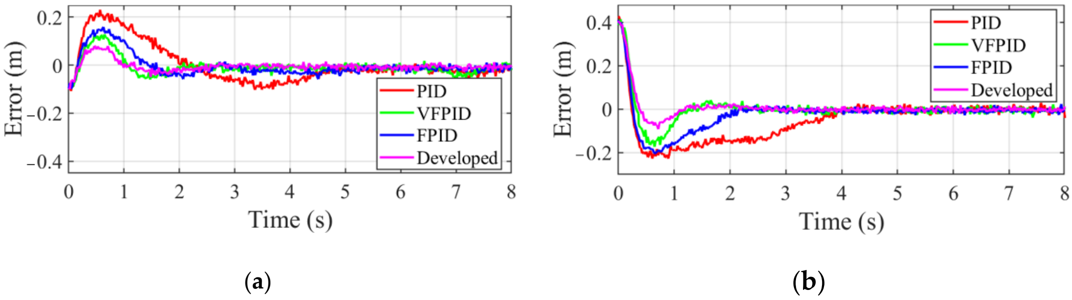


Figure 13. Variation of tracking error in the tracking process with Gaussian noise. (**a**) Error along x axis. (**b**) Error along y axis.

The disturbance with -0.15 rad is introduced to six joint angles between 3.052 s and 3.152 s, as shown in Figure 14. The tracking errors along x-axis and y-axis are presented in Figure 15a,b, respectively. It is seen that the tracking errors increase sharply under the influence of interference. Take the results obtained by the developed controller as an example, a_{ex} and a_{ey} denote the error variation amplitude caused by disturbance, and t_{ex} and t_{ex} indicate the adjustment time from appearing disturbance to return to the steady tracking state along x-axis and y-axis, respectively. Table 4 gives the results of error variation amplitude and adjustment time. It can be noted that through the adjustment of the developed controller, the tracking error not only has smaller amplitude, but also reaches the stable state more quickly than other controllers.

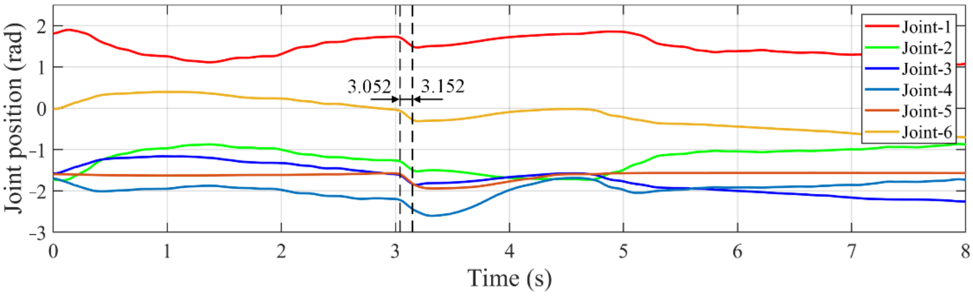
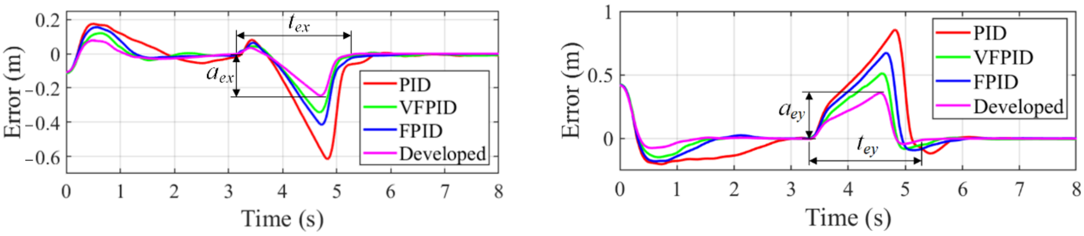


Figure 14. Variation of joint position in the tracking process with −0.15 radian disturbance.



(**a**) (**b**)

Figure 15. Variation of tracking error in the tracking process with −0.15 radian disturbance. (**a**) Error along *x*-axis. (**b**) Error along *y*-axis.

Table 4. The results of error variation amplitude and adjustment time.

	PID	VFPID	FPID	Developed
a_{ex} (m)	0.615	0.342	0.415	0.244
a_{ey} (m)	4.244	2.717	3.98	2.484
t_{ex} (s)	3.872	2.756	3.392	2.285
t_{ey} (s)	3.767	2.682	3.317	2.217

The PID coefficients of the developed controller are adaptively tuned during the tracking process in the presence of disturbance, which are presented in Figure 16. When there are large tracking errors, the velocity feedforward control plays a major role. Thus, a faster response is obtained by the developed controller than the PID controller. At this time, the proportional module plays a minor adjustment role with small proportional coefficient P. Near the stable tracking phase i.e., 1~2 s, the role of velocity feedforward decreases gradually as the tracking of errors decreases. At this time, the coefficient P tends to increase to further reduce the tracking error. During the stable tracking phase, the proportional coefficient P tends to be unchanged for high stability, and the integral coefficient I fluctuates to eliminate the steady-state error. The differential coefficient D is almost zero, because there is no significant position tracking lag under the control of velocity feedforward module.

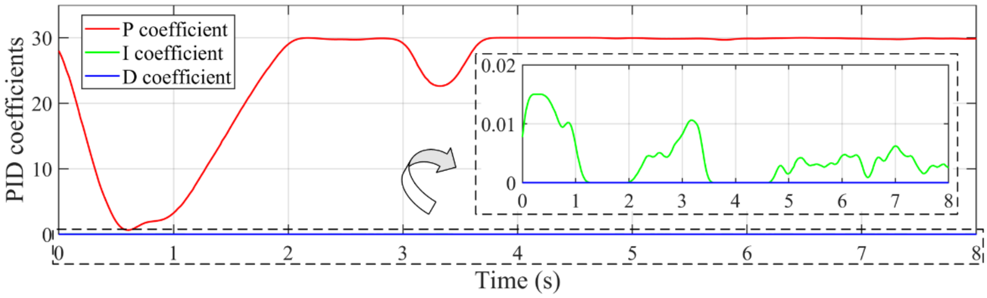


Figure 16. Variation of PID coefficients in the tracking process with −0.15 rad disturbance.

5.3. Autonomous Operation Experiment

To test the practical performance of the proposed control system, the autonomous operation experiment is performed, including tracking and grasping the object moving on pipeline and placing it into the material box. At first, the manipulator receives the output signal of the developed controller to track the moving target. To ensure the real-time performance, the trajectory of manipulator end-effector from the initial point S to a certain grab point G is online, planned based on the proposed strategy, as presented by the dotted line in Figure 17. When the tracking error is within the allowed 5 mm error threshold, the manipulator end-effector performs a grasping operation. Finally, the captured object is placed into the material box along the offline planned trajectory, as shown by the dashed line between point G and E in Figure 17. The trajectory of moving object is the one from start point O to grab point G, as shown by the solid line in Figure 17, and the corresponding tracking error is given in Figure 18.

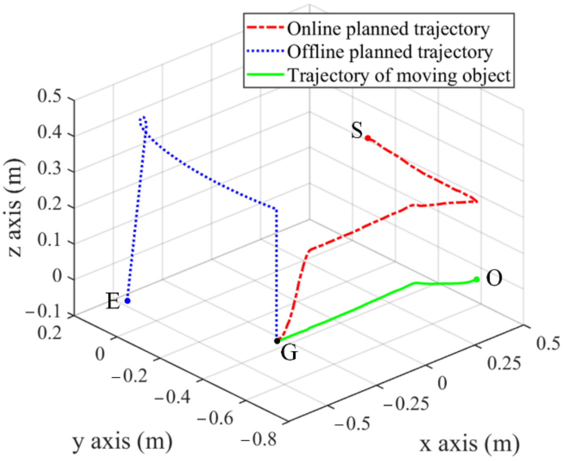


Figure 17. Trajectory of manipulator end-effector and moving object in experiment.

The key frames of the autonomous operation using 6-DOF manipulator are presented in Figure 19. The manipulator end-effector firstly starts from the starting point and passes through the transition point to reach the tracking point, as shown in Figure 19a–c. When the tracking error is within the allowed 5 mm error threshold, the grasping operation is conducted as shown in Figure 19d,e. The offline placement operation will be performed when the moving target is completely captured. At this time, manipulator end-effector lifts the object, moves it to the pre-placement pose and performs the placement operation, as shown in Figure 19f–h. Finally, manipulator end-effector returns to the initial point represented in Figure 19i and prepares for the next round of autonomous operation. The above experimental results confirm the practicality of the proposed capture and placement control system.

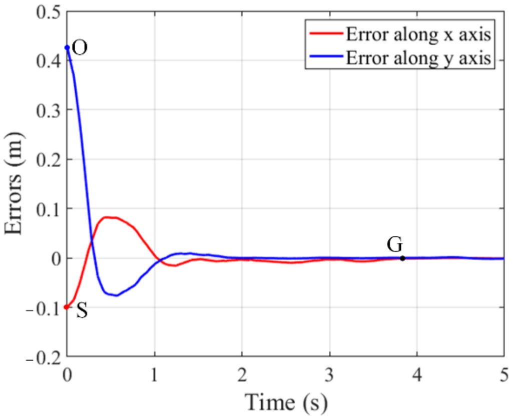


Figure 18. Tracking error of manipulator end-effector to the moving target.

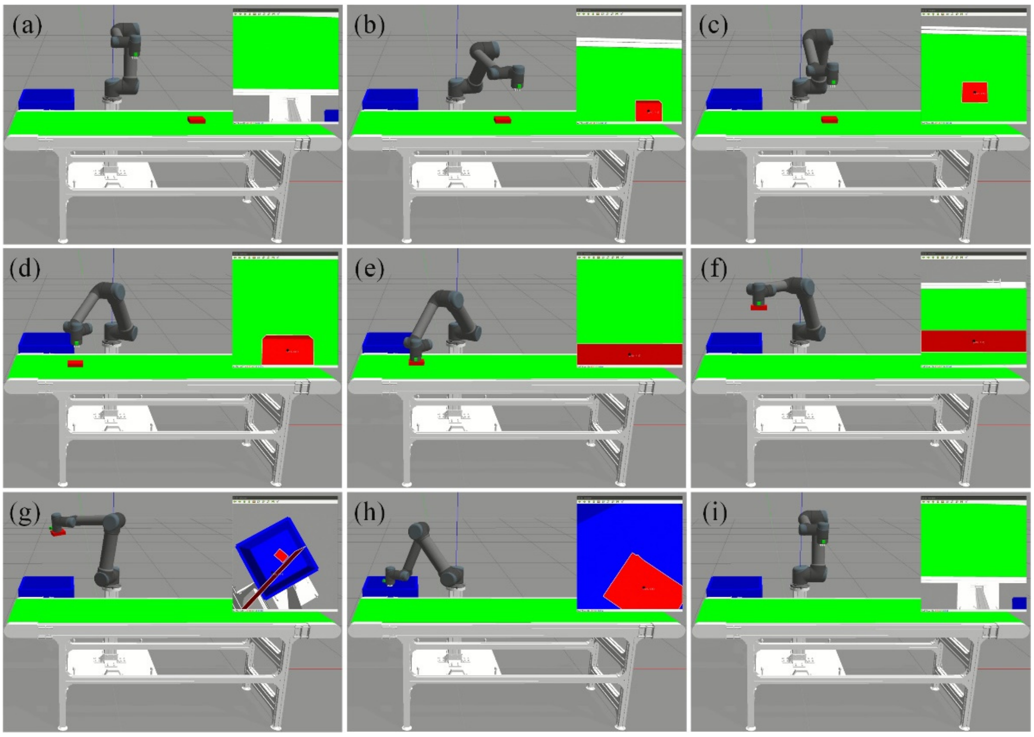


Figure 19. The key frames in the autonomous operation experiment. (**a**) Initial pose. (**b**) Transition phase. (**c**) Start tracking. (**d**) Before grasping. (**e**) Perform grasping. (**f**–**h**) Lift object, perform pre-placement and placement. (**i**) Return to initial pose.

In addition, the autonomous operation experiments with and without using the capture and placement strategy are performed. For the experiments without using the proposed strategy, all the trajectories of manipulator end-effector are online planned. In particular, the trajectory in placement stage is obtained by inverse kinematics, which is different from the one using a proposed strategy. The overall time spent in the experiments with and without using the proposed strategy is recorded, which is denoted by t_i and t_o, respectively. Table 5 presents the result obtained by the developed control system and the traditional ones for comparison, including PID, VFPID, and FPID. It can be noted that the overall time spent in the autonomous operation by the developed control system is less than others. Besides, for all controllers, the time spent in the operation with using the proposed strategy is smaller than that without using the strategy. That is because for the operation without using the proposed strategy, the inverse kinematic calculation is conducted online based on the pre-defined waypoints in the placement process, which will cost more time. The above experiments demonstrate that the developed control system has a high operation efficiency.

Table 5. Overall time spent in autonomous operation.

	PID	VFPID	FPID	Developed
t_i (s)	10.541	6.435	8.347	5.825
t_o (s)	11.260	7.129	9.052	6.516

6. Conclusions

This paper develops a capture and placement control system for 6-DOF manipulator to capture the moving target. An online capture and offline placement strategy is adopted in the developed system, which improves the overall manipulation efficiency compared with the existing control system. In addition, a new PBVS controller is designed by combining the estimated velocity feedforward module and refined PID controller. The motion estimation method combined Kalman filter and interpolation operation is proposed and the accurate and smooth estimation for target's pose and velocity are thus obtained, which provides velocity value to feedforward module. The PID controller is refined by dead zone constraint to reduce the manipulator's jitter, and by fuzzy control to adaptively tune PID parameters. Validation experiments prove that the developed controller achieves faster response speed and stronger anti-interference ability compared to the existing PID-based controller. In the future, the autonomous operation for moving target in 3D space will be conducted on the real experimental setup, and the end-to-end visual servo manipulation will be studied to further improve the robustness of the autonomous operation system.

Author Contributions: Conceptualization, X.C. and P.L.; methodology, F.W.; software, R.Y.; validation, X.C. and F.W.; formal analysis, X.C.; investigation, X.C.; resources, F.W.; data curation, X.C.; writing—original draft preparation, X.C.; writing—review and editing, P.L.; visualization, X.C.; supervision, P.L.; project administration, R.Y.; funding acquisition, P.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the scientific research project of our laboratory, but it is inconvenient to disclose the project information here due to confidentiality.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the funded project's scope of deliverables.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Forward Kinematics Model

The link coordinate system of 6-DOF manipulator based on D-H method is shown in Figure A1. a_{i-1} and α_{i-1} denote the distance and angle of axis Z_{i-1} with respect to axis Z_i. d_i is the distance between axis X_{i-1} and X_i along Z_i. θ_i is the angle of axis X_{i-1} and X_i with respect to Z_i. According to D-H method, the transformation matrix from $(i-1)$-th joint frame to i-th joint frame can be obtained, and its homogeneous coordinate expression is expressed in Equation (A1).

$$\boldsymbol{T}_{i-1}^{i} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_{i-1} & s\theta_i s\alpha_{i-1} & \alpha_{i-1}c\theta_i \\ s\theta_i & c\theta_i c\alpha_{i-1} & -c\theta_i s\alpha_{i-1} & \alpha_{i-1}s\theta_i \\ 0 & s\alpha_{i-1} & c\alpha_{i-1} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A1}$$

where c and s denote cosine and sine function, respectively. The transformation matrix of end-effector frame relative to global frame, i.e., base coordinate system of the manipulator can be expressed as:

$$^{g}\boldsymbol{T}_e = \boldsymbol{T}_5^6\boldsymbol{T}_4^5\boldsymbol{T}_3^4\boldsymbol{T}_2^3\boldsymbol{T}_1^2\boldsymbol{T}_0^1 \tag{A2}$$

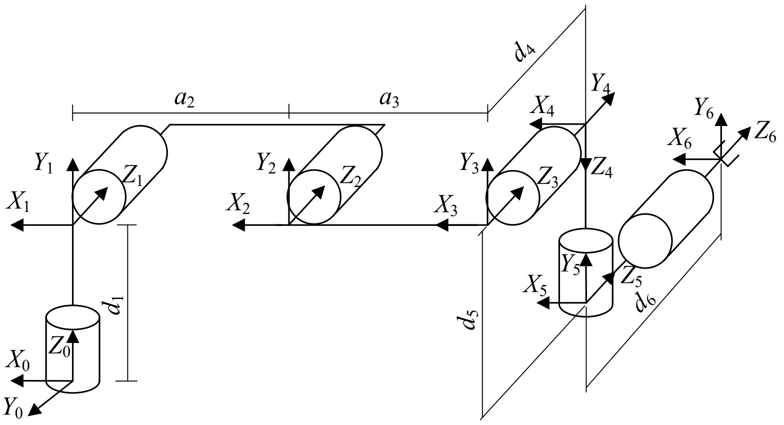


Figure A1. Schematic diagram of 6-DOF manipulator link coordinate system.

So long as the joint position is known, the pose of end-effector in global frame can be determined. Suppose that the homogeneous coordinate transformation matrix of end-effector relative to global frame is:

$$^{g}\boldsymbol{T}_e = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A3}$$

where (p_x, p_y, p_z) denotes the position of end-effector, and $\begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}$ is the rotation matrix of the end-effector's orientation. According to Equations (A1)–(A3), the expression for the elements in $^{g}\boldsymbol{T}_e$ can be obtained by:

$$\begin{cases} n_x = & c\theta_6\{s\theta_1 s\theta_5 + c\theta_5 c\theta_1 c(\theta_2+\theta_3+\theta_4)\} - s\theta_6 c\theta_1 s(\theta_2+\theta_3+\theta_4) \\ n_y = & c\theta_6\{c\theta_5 s\theta_1 c(\theta_2+\theta_3+\theta_4) - c\theta_1 \mathrm{s}\theta_5\} - s\theta_6 s\theta_1 s(\theta_2+\theta_3+\theta_4) \\ n_z = & c\theta_5 c\theta_6 s(\theta_2+\theta_3+\theta_4) + s\theta_6 c(\theta_2+\theta_3+\theta_4) \\ o_x = & -s\theta_6\{c\theta_5 c\theta_1 c(\theta_2+\theta_3+\theta_4) + s\theta_1 s\theta_5\} - c\theta_6 c\theta_1 s(\theta_2+\theta_3+\theta_4) \\ o_y = & -s\theta_6\{c\theta_5 s\theta_1 c(\theta_2+\theta_3+\theta_4) - c\theta_1 s\theta_5\} - c\theta_6 s\theta_1 s(\theta_2+\theta_3+\theta_4) \\ o_z = & c\theta_6 c(\theta_2+\theta_3+\theta_4) - c\theta_5 s\theta_6 s(\theta_2+\theta_3+\theta_4) \\ a_x = & -s\theta_5 c\theta_1 c(\theta_2+\theta_3+\theta_4) + c\theta_5 s\theta_1 \\ a_y = & -s\theta_5 s\theta_1 c(\theta_2+\theta_3+\theta_4) - c\theta_1 c\theta_5 \\ a_z = & -s\theta_5 s(\theta_2+\theta_3+\theta_4) \\ p_x = & d_5 c\theta_1 s(\theta_2+\theta_3+\theta_4) + d_4 s\theta_1 + d_6\{c\theta_5 s\theta_1 - s\theta_5 c\theta_1 c(\theta_2+\theta_3+\theta_4)\}+ \\ & a_1 c\theta_1 c\theta_2 + a_2 c\theta_1 c(\theta_2+\theta_3) \\ p_y = & d_5 s\theta_1 s(\theta_2+\theta_3+\theta_4) - d_4 c\theta_1 - d_6\{s\theta_5 s\theta_1 c(\theta_2+\theta_3+\theta_4) + c\theta_1 c\theta_5)\} \\ & +a_1 c\theta_2 s\theta_1 + a_2 s\theta_1 c(\theta_2+\theta_3) \\ p_z = & d_1 - d_5 c(\theta_2+\theta_3+\theta_4) + a_1 s\theta_2 + a_2 s(\theta_2+\theta_3) - d_6 s\theta_5 s(\theta_2+\theta_3+\theta_4) \end{cases} \tag{A4}$$

Appendix B. Instantaneous Kinematics Model

The velocity in global frame of end-effector is output by the designed controller. To control the manipulator, it is necessary to transform end-effector's velocity into joint velocity of manipulator. The instantaneous kinematics analysis is carried out to achieve the above transformation based on the forward kinematics model in Appendix A. The instantaneous motion of the manipulator can be expressed by the differential equation of position and orientation:

$$\frac{d^g\boldsymbol{x}_e}{dt} = \frac{df(\boldsymbol{\theta})}{dt} = \frac{df(\boldsymbol{\theta})}{d\boldsymbol{\theta}}\frac{d\boldsymbol{\theta}}{dt} \tag{A5}$$

where ${}^g\boldsymbol{x}_e$ is end-effector's pose in global frame, $\boldsymbol{\theta}= (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ is the joint position and $f(\boldsymbol{\theta}) = {}^gT_e$. Equation (A5) can be simplified to ${}^g\boldsymbol{v}_e = \boldsymbol{J\omega}$, and $\boldsymbol{J} = \frac{df(\boldsymbol{\theta})}{d\boldsymbol{\theta}}$ is the Jacobian matrix, which describes the mapping from joint velocity $\boldsymbol{\omega}$ to end-effector's velocity ${}^g\boldsymbol{v}_e$. Through the pseudo inverse transformation of Jacobian matrix, the expression of joint velocity can be obtained by $\boldsymbol{\omega} = \boldsymbol{J}^{-1}{}^g\boldsymbol{v}_e$.

For the convenience of derivation, the Jacobian matrix is divided into two parts, i.e., $\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_v \\ \boldsymbol{J}_w \end{bmatrix}$. Thereinto, $\boldsymbol{J}_v$ and $\boldsymbol{J}_w$ denote the linear velocity and angular velocity Jacobian matrix, respectively. Firstly, $\boldsymbol{J}_v$ is deduced by the differential derivation for joint vector $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ according to the expression (p_x, p_y, p_z) in forward kinematics model:

$$\boldsymbol{J}_v = \begin{bmatrix} \frac{\partial p_x}{\partial\theta_1} & \frac{\partial p_x}{\partial\theta_2} & \frac{\partial p_x}{\partial\theta_3} & \frac{\partial p_x}{\partial\theta_4} & \frac{\partial p_x}{\partial\theta_5} & \frac{\partial p_x}{\partial\theta_6} \\ \frac{\partial p_y}{\partial\theta_1} & \frac{\partial p_y}{\partial\theta_2} & \frac{\partial p_y}{\partial\theta_3} & \frac{\partial p_y}{\partial\theta_4} & \frac{\partial p_y}{\partial\theta_5} & \frac{\partial p_y}{\partial\theta_6} \\ \frac{\partial p_z}{\partial\theta_1} & \frac{\partial p_z}{\partial\theta_2} & \frac{\partial p_z}{\partial\theta_3} & \frac{\partial p_z}{\partial\theta_4} & \frac{\partial p_z}{\partial\theta_5} & \frac{\partial p_z}{\partial\theta_6} \end{bmatrix} \tag{A6}$$

According to Equation (A4), the expression for each element in $\boldsymbol{J}_v$ can be obtained. Manipulator joint rotates around its own rotation axis, which will correspondingly cause the end-effector to rotate around this axis. The joint's angular velocity is a vector pointing to the rotation axis, and each rotation joint rotates around its own Z axis. To obtain $\boldsymbol{J}_w$, it is necessary to transform each rotation joint reference system to the global frame. Therefore, the expression of $\boldsymbol{J}_w$ can be obtained by:

$$\boldsymbol{J}_w = \begin{bmatrix} \boldsymbol{Z}_0^1 & \boldsymbol{Z}_0^2 & \boldsymbol{Z}_0^3 & \boldsymbol{Z}_0^4 & \boldsymbol{Z}_0^5 & \boldsymbol{Z}_0^6 \end{bmatrix} \tag{A7}$$

where $\boldsymbol{Z}_0^i$ is the transformation from base frame to i-th joint frame. According to the derivation process of the forward kinematics model, it can be deduced that $\boldsymbol{Z}_0^i$ is the first three rows of the third column in transformation matrix $\boldsymbol{T}_0^i$. Therefore, the element in $\boldsymbol{J}_w$ can be obtained by $\boldsymbol{Z}_0^1 = \begin{bmatrix} s\theta_1 \\ -c\theta_1 \\ 0 \end{bmatrix}$, $\boldsymbol{Z}_0^2 = \begin{bmatrix} s\theta_1 \\ -c\theta_1 \\ 0 \end{bmatrix}$, $\boldsymbol{Z}_0^3 = \begin{bmatrix} s\theta_1 \\ -c\theta_1 \\ 0 \end{bmatrix}$, $\boldsymbol{Z}_0^4 = \begin{bmatrix} c\theta_1 s(\theta_2+\theta_3+\theta_4) \\ -s\theta_1 s(\theta_2+\theta_3+\theta_4) \\ c(\theta_2+\theta_3+\theta_4) \end{bmatrix}$, $\boldsymbol{Z}_0^5 = \begin{bmatrix} s\theta_1 c\theta_5 - c\theta_1 c(\theta_2+\theta_3+\theta_4)s\theta_5 \\ -c\theta_1 c\theta_5 - s\theta_1 c(\theta_2+\theta_3+\theta_4)s\theta_5 \\ -s(\theta_2+\theta_3+\theta_4)s\theta_5 \end{bmatrix}$, $\boldsymbol{Z}_0^6 = \begin{bmatrix} -s\theta_5 c\theta_1 c(\theta_2+\theta_3+\theta_4) + c\theta_5 s\theta_1 \\ -s\theta_5 s\theta_1 c(\theta_2+\theta_3+\theta_4) - c\theta_1 c\theta_5 \\ -s\theta_5 s(\theta_2+\theta_3+\theta_4) \end{bmatrix}$.

The above is the derivation process of Jacobian matrix. Based on this, the Jacobian pseudo inverse matrix $\boldsymbol{J}^{-1}$ can be obtained, and end-effector's velocity can be converted to joint velocity by $\boldsymbol{\omega} = \boldsymbol{J}^{-1}{}^g\boldsymbol{v}_e$.

Appendix C. Kalman Filter Model

The model of predicting a system state, including position orientation, can be established according to the Kalman filter method. Taking the system state along x axis in global frame, it can be expressed as:

$${}^g\hat{\boldsymbol{X}}_{wx}(k) = \boldsymbol{A}{}^g\boldsymbol{X}_{wx}(k-1) + \boldsymbol{Bu}(k-1) + \boldsymbol{q}(k) \tag{A8}$$

where ${}^{g}\hat{\boldsymbol{X}}_{wx}(k) = \left[{}^{g}\hat{x}_{xw}(k) \quad {}^{g}\hat{v}_{xw}(k) \quad {}^{g}\hat{a}_{xw}(k)\right]^{T}$, is the predicted system state value at k-th period, including position ${}^{g}\hat{x}_{xw}(k)$, velocity ${}^{g}\hat{v}_{xw}(k)$, and acceleration ${}^{g}\hat{a}_{xw}(k)$ along x axis in global frame. ${}^{g}\boldsymbol{X}_{wx}(k-1) = \left[{}^{g}x_{wx}(k-1) \quad {}^{g}v_{wx}(k-1) \quad {}^{g}a_{wx}(k-1)\right]^{T}$ is the optimal estimation state at $(k-1)$-th period. B is the matrix which can convert inputs to system states.$\boldsymbol{u}(k-1)$ is system's control variable at $(k-1)$-th period. Since the target moves on a pipeline, it can be assumed that there is no control variable, i.e., $\boldsymbol{u}(k-1) = \boldsymbol{0}$, and the target moves at a uniform acceleration in a small time period. Thus, the state transition matrix $\boldsymbol{A} = \begin{bmatrix} 1 & T & 0.5T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}$ can be obtained. It should be noted that T is the control period. $\boldsymbol{q}(k)$ is the process noise vector and is assumed to obey a zero-mean Gaussian distribution with covariance $\boldsymbol{Q}(k)$. In general, it is difficult to determine $\boldsymbol{Q}(k)$ due to the non-cooperative nature of the moving target and unknown motion of camera. In the current work, the constant process noise covariance matrix $\boldsymbol{Q}(k) = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix} \times 10^{-4}$ works well after tuning the Kalman filter in experiments.

The measurement result $Z_x(k)$ along the x axis can be expressed as:

$$Z_x(k) = \boldsymbol{H}_x {}^{g}\hat{\boldsymbol{X}}_{wx}(k) + r(k) \tag{A9}$$

where $\boldsymbol{H}_x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$ is the transition matrix from state variables to measurements. $r(k)$ is the measurement noise from camera. It is assumed as Gaussian noise i.e., $r(k) \sim N(0, R(k))$. The measurement noise covariance $R(k)$ with 5×10^{-4} is determined according to camera's simulation model.

The priori estimate covariance $\hat{\boldsymbol{P}}(k)$ at k-th period can be obtained by:

$$\hat{\boldsymbol{P}}(k) = \boldsymbol{A}\boldsymbol{P}(k-1)\boldsymbol{A}^{T} + \boldsymbol{Q}(k) \tag{A10}$$

where $\boldsymbol{P}(k-1)$ is the posteriori estimate covariance at $(k-1)$-th period, which represents the uncertainty of estimating ${}^{g}\boldsymbol{X}_{wx}(k-1)$. Kalman gain matrix $\boldsymbol{K}(k)$ can be obtained by:

$$\boldsymbol{K}(k) = \hat{\boldsymbol{P}}(k)\boldsymbol{H}_x{}^{T}\left[\boldsymbol{H}_x\hat{\boldsymbol{P}}(k)\boldsymbol{H}_x{}^{T} + R(k)\right]^{-1} \tag{A11}$$

Subsequently, the optically estimated state ${}^{g}\boldsymbol{X}_{wx}(k)$ can be updated by:

$${}^{g}\boldsymbol{X}_{wx}(k) = {}^{g}\hat{\boldsymbol{X}}_{wx}(k) + \boldsymbol{K}(k)\left[\boldsymbol{Z}_x(k) - \boldsymbol{H}_x {}^{g}\hat{\boldsymbol{X}}_{wx}(k)\right] \tag{A12}$$

Finally, to carry out the next round of optimal estimation, the posteriori estimate covariance $\boldsymbol{P}(k)$ is updated by:

$$\boldsymbol{P}(k) = [\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{H}_x]\hat{\boldsymbol{P}}(k) \tag{A13}$$

where $\boldsymbol{P}(k)$ is the matrix of covariance between system states. Its diagonal elements are the variances of each state, and other elements are the covariance of corresponding elements. Thus, $\boldsymbol{P}(k)$ can be expressed as:

$$\boldsymbol{P}(k) = \begin{bmatrix} \mathrm{cov}({}^{g}x_{wx}(k), {}^{g}x_{wx}(k)) & \mathrm{cov}({}^{g}x_{wx}(k), {}^{g}v_{wx}(k)) & \mathrm{cov}({}^{g}x_{wx}(k), {}^{g}a_{wx}(k)) \\ \mathrm{cov}({}^{g}v_{wx}(k), {}^{g}x_{wx}(k)) & \mathrm{cov}({}^{g}v_{wx}(k), {}^{g}v_{wx}(k)) & \mathrm{cov}({}^{g}v_{wx}(k), {}^{g}a_{wx}(k)) \\ \mathrm{cov}({}^{g}a_{wx}(k), {}^{g}x_{wx}(k)) & \mathrm{cov}({}^{g}a_{wx}(k), {}^{g}v_{wx}(k)) & \mathrm{cov}({}^{g}a_{wx}(k), {}^{g}a_{wx}(k)) \end{bmatrix} \tag{A14}$$

where $\mathrm{cov}(\cdot)$ is the covariance calculation formula. Generally, only $\boldsymbol{P}(0)$ needs to be set, and others will be calculated automatically. In this paper, determining $\boldsymbol{P}(0)$ works well in Kalman filter experiments according to the initial state, i.e., ${}^{g}x_{wx}(0) = -0.2$, ${}^{g}v_{wx}(0) = 0$, ${}^{g}a_{wx}(0) = 0$ and the state average value in tracking process, i.e., ${}^{g}\overline{x}_{wx}$= 0.15, ${}^{g}\overline{v}_{wx}$= 0.2, ${}^{g}\overline{a}_{wx}$= 2.0. After several rounds of iteration, the $\boldsymbol{P}(k)$ will get closer to the real value.

By the above way, the optimal state estimation ${}^{g}\boldsymbol{X}_{wx}(k)$ will converge after a few iterations. The calculation process of estimating other states is the same as that of x axis.

References

1. Castelli, F.; Michieletto, S.; Ghidoni, S.; Pagello, E. A machine learning-based visual servoing approach for fast robot control in industrial setting. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 172988141773888. [CrossRef]
2. Filipescu, A.; Mincă, E.; Filipescu, A.; Coandă, H.G. Manufacturing Technology on a Mechatronics Line Assisted by Autonomous Robotic Systems, Robotic Manipulators and Visual Servoing Systems. *Actuators* **2020**, *9*, 127. [CrossRef]
3. Muñoz-Benavent, P.; Solanes, J.E.; Gracia, L.; Tornero, J. Robust auto tool change for industrial robots using visual servoing. *Int. J. Syst. Sci.* **2019**, *50*, 432–449. [CrossRef]
4. Chaumette, F.; Hutchinson, S. Visual servo control. I. Basic approaches. *IEEE Robot. Autom. Mag.* **2006**, *13*, 82–90. [CrossRef]
5. Chaumette, F.; Hutchinson, S. Visual servo control. II. Advanced approaches. *IEEE Robot. Autom. Mag.* **2007**, *14*, 109–118. [CrossRef]
6. Kang, M.; Chen, H.; Dong, J. Adaptive visual servoing with an uncalibrated camera using extreme learning machine and Q-leaning. *Neurocomputing* **2020**, *402*, 384–394. [CrossRef]
7. Qiu, Z.; Hu, S.; Liang, X. Disturbance observer based adaptive model predictive control for uncalibrated visual servoing in constrained environments. *ISA Trans.* **2020**, *106*, 40–50. [CrossRef]
8. Li, K.; Wang, H.; Liang, X.; Miao, Y. Visual Servoing of Flexible-Link Manipulators by Considering Vibration Suppression Without Deformation Measurements. *IEEE Trans. Cybern.* **2021**, *3*, 1–10. [CrossRef]
9. Larouche, B.P.; Zhu, Z.H. Autonomous robotic capture of non-cooperative target using visual servoing and motion predictive control. *Auton. Robot.* **2014**, *37*, 157–167. [CrossRef]
10. Larouche, B.P.; Zhu, Z.H. Position-based visual servoing in robotic capture of moving target enhanced by Kalman filter. *Int. J. Robot. Autom.* **2015**, *30*, 267–277. [CrossRef]
11. Dong, G.; Zhu, Z.H. Position-based visual servo control of autonomous robotic manipulators. *Acta Astronaut.* **2015**, *115*, 291–302. [CrossRef]
12. Shi, H.; Chen, J.; Pan, W.; Cho, Y.Y. Collision Avoidance for Redundant Robots in Position-Based Visual Servoing. *IEEE Syst. J.* **2019**, *13*, 3479–3489. [CrossRef]
13. Wei, Y.; Wang, H.; Tian, Y. Adaptive sliding mode observer–based integral sliding mode model-free torque control for elastomer series elastic actuator–based manipulator. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2022**, *236*, 1010–1028. [CrossRef]
14. Tofigh, M.A.; Mahjoob, M.J.; Hanachi, M.R.; Ayati, M. Fractional sliding mode control for an autonomous two-wheeled vehicle equipped with an innovative gyroscopic actuator. *Robot. Auton. Syst.* **2021**, *140*, 103756. [CrossRef]
15. Grami, S.; Okonkwo, P.C. Friction Compensation in Robot Manipulator Using Artificial Neural Network. In Proceedings of the Advances in Automation, Signal Processing, Instrumentation, and Control, Singapore, 6 March 2021.
16. Li, S.; Wang, H.; Rafique, M.U. A novel recurrent neural network for manipulator control with improved noise tolerance. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 1908–1918. [CrossRef]
17. Su, H.; Hu, Y.; Karimi, H.R.; Knoll, A.; De, M.E. Improved recurrent neural network-based manipulator control with remote center of motion constraints: Experimental results. *Neural Netw.* **2020**, *131*, 291–299. [CrossRef]
18. Hu, Y.; Si, B.A. Reinforcement Learning Neural Network for Robotic Manipulator Control. *Neural Comput.* **2018**, *30*, 1983–2004. [CrossRef]
19. Yen, V.T.; Nan, W.Y.; Van, C.P. Robust Adaptive Sliding Mode Neural Networks Control for Industrial Robot Manipulators. *Int. J. Control Autom.* **2019**, *17*, 783–792. [CrossRef]
20. Li, P.; Zhu, G. IMC-based PID control of servo motors with extended state observer. *Mechatronics* **2019**, *62*, 102252. [CrossRef]
21. Jin, X.; Chen, K.; Zhao, Y.; Ji, J.; Jing, P. Simulation of hydraulic transplanting robot control system based on fuzzy PID controller. *Measurement* **2020**, *164*, 108023. [CrossRef]
22. Ahmed, T.; Islam, M.R.; Brahmi, B.; Rahman, M.H. Robustness and Tracking Performance Evaluation of PID Motion Control of 7 DoF Anthropomorphic Exoskeleton Robot Assisted Upper Limb Rehabilitation. *Sensors* **2022**, *22*, 3747. [CrossRef] [PubMed]
23. Loucif, F.; Kechida, S.; Sebbagh, A. Whale optimizer algorithm to tune PID controller for the trajectory tracking control of robot manipulator. *J. Braz. Soc. Mech. Sci.* **2019**, *42*, 1. [CrossRef]
24. Xue, F.; Fan, Z. Kinematic control of a cable-driven snake-like manipulator for deep-water based on fuzzy PID controller. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2022**, *236*, 989–998. [CrossRef]
25. Huang, P.; Zhang, Z.; Luo, X. Feedforward-plus-proportional–integral–derivative controller for agricultural robot turning in headland. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 172988141989767. [CrossRef]
26. Londhe, P.S.; Singh, Y.; Santhakumar, M.; Patre, B.M.; Waghmare, L.M. Robust nonlinear PID-like fuzzy logic control of a planar parallel (2PRP-PPR) manipulator. *ISA Trans.* **2016**, *63*, 218–232. [CrossRef] [PubMed]
27. Koubâa, A. *Robot Operating System (ROS)*; Springer: Cham, Switzerland, 2017; pp. 112–156.
28. Jalil, A. Robot Operating System Dan Gazebo Sebagai Media Pembelajaran Robot Interaktif. *ILKOM J. Ilm.* **2018**, *10*, 284–289. [CrossRef]
29. Wu, Y.; Yang, M.H. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013.
30. Chen, X.; Wang, Z.; Wang, Y.; Chi, G. Investigation on MRR and machining gap of micro reciprocated wire-EDM for SKD11. *Int. J. Precis. Eng. Manuf.* **2020**, *21*, 11–22. [CrossRef]