Journal of Intelligent and Robotic Systems **33:** 1–23, 2002.

Navigation-Guidance-Based Robotic Interception of Moving Objects in Industrial Settings

J. M. BORG, M. MEHRANDEZH, R. G. FENTON and B. BENHABIB
Computer Integrated Manufacturing Laboratory, University of Toronto, 5 King's College Road, Toronto, ON M5S 3G8, Canada; e-mail: beno@mie.utoronto.ca

(Received: 4 August 2000; in final form: 13 February 2001)

Abstract. An important task for autonomous industrial robotic systems is the interception of moving objects. In order to achieve this objective, an on-line robot-motion planning technique that utilizes real-time sensory feedback about the object's motion is needed. In this paper, an Ideal Proportional Navigation Guidance (IPNG) based technique is utilized for on-line robot-motion planning. One must note, however, that navigation-guidance techniques were originally developed for bringing the interceptor into a collision course with (hostile) airborne targets. Therefore, in our case, a conventional tracking technique must be utilized as a subsequent phase to an initial IPNG-based robot-motion planning phase in order to ensure smooth interception.

The implementation of the hybrid scheme in industrial settings, where one may not have access to the robot's dynamic model nor to the joints' controllers, is discussed. Real-time experimental results using an industrial robot and a computer-vision system are presented, confirming the (interception-time) superiority of our proposed scheme over conventional tracking techniques.

Key words: moving object interception, robot motion planning, proportional navigation guidance.

1. Introduction

Recent trends in industry, moving from mass production to mass customization, have contributed to the urgent call for the development of intelligent robotic systems that can operate autonomously. In this context, robotic interception is defined as approaching a moving object and matching its location and velocity (i.e., bringing the end-effector of the manipulator to a pre-grasp rendezvous location with the moving object), in the shortest possible time. On-line robot-motion planning is required for successful interception if the motion of the object is not completely known a priori. Various motion-planning strategies have been reported in the literature for this purpose, depending on the object's motion type. In this paper, we present the implementation of a navigation-based robot-motion planning technique for the interception of "maneuvering" and "non-maneuvering" objects using an industrial robotic system. A target is considered to be *maneuvering* if it varies its motion randomly and quickly, making time-optimal interception a difficult task. In contrast, a target is considered to be *non-maneuvering* if it moves on a contin-

uous path with constant velocity or acceleration, enabling the accurate long-term prediction of the target's motion for time-optimal interception.

Prediction-based techniques have been proposed for robotic interception when the object's motion is of "non-maneuvering" type. In such systems, the object's motion through the robot's workspace is predicted and, subsequently, the robot's trajectory to an anticipated rendezvous point on the (long-term-predicted) object path is planned and executed. Such systems are referred to as Prediction, Planning and Execution (PPE) Systems, e.g., [1]. The stages of PPE can be used in an "active" mode to ensure the successful completion of the interception task, hence, APPE. The cornerstone of APPE techniques is their capability of re-planning the rendezvous-point, in response to changes in the predicted object trajectory.

An APPE approach was used by Buttazzo et al. [2] to catch a planar toy-mouse, and by Fernandes and Lima [3] to catch ping-pong balls rolling on a table. In the APPE approach proposed by Croft et al. [4], the potential rendezvous points are not restricted to a small set of candidate points. Instead, they are optimally selected (anywhere) along the predicted object's trajectory subject to the robot's motion constraints.

Visual-servoing systems have been proposed to track and intercept maneuvering objects (e.g., [5, 6]). Such systems continually minimize the difference in states of the object and the end-effector. This constitutes position-based servoing. Alternatively, in image-based servoing systems, feedback control values are computed directly from feature vectors in the camera image plane (e.g., [7]). Due to their computational efficiency, visual-servoing systems are well suited for tracking maneuvering objects. However, trying to match the object's state when the robot is far may not be advantageous. Lin et al. [8] addressed this problem by using a heuristic coarse-tuning method to bring the end-effector to the vicinity of the moving object and subsequently switching to a fine-tuning method when the robot is within a pre-defined distance of the object.

Navigation-guidance laws have also been widely used for tracking and intercepting maneuvering targets. However, they have been predominantly implemented in tracking and intercepting airborne missiles and evasive aircrafts. Due to its simplicity of onboard implementation, Proportional Navigation Guidance (PNG) has been the most widely researched and used guidance law (e.g., [9]). There have been very few attempts, however, to use guidance laws for robotic-interception of moving objects. The utilization of a navigation-based technique in robotics was first reported by Piccardo and Hondred [10], who utilized a PNG law to intercept an object moving on a straight line. No experimental results were reported. More recently, Su and Xi [11] have also proposed a guidance-based strategy for moving-object interception. In this method, the angle between the Line-of-Sight (LOS) (i.e., a line connecting the interceptor to the target) and the object velocity is changed at a constant rate. In their experimental work, both maneuvering and non-maneuvering objects were successfully intercepted.

In both [10, 11], however, velocity matching was not considered as an issue. In contrast, this issue was addressed in the Ideal Proportional Navigation Guidance (IPNG) based interception scheme recently developed in our laboratory [12]. IPNG is a navigation scheme in which the acceleration command is applied in the direction normal to the relative velocity between the interceptor and the target, and tries to turn the relative velocity to the direction of the LOS with utmost effort. It was originally developed in [18] for the control of missiles tracking airborne targets.

The IPNG-based robotic interception technique was developed to intercept maneuvering and non-maneuvering objects, making full use of the robot's capabilities. The proposed approach is a hybrid method, in which a tracking method takes over control of the robot from the navigation-guidance method at an optimal instant to bring the end-effector to a pre-grasp situation (matching object's location and velocity). In this paper, the implementation of this scheme for the interception of moving objects in industrial settings is presented.

2. System Overview

Navigation-based techniques nullify the time-rate of change of the LOS angle (i.e., the angle that the LOS makes with a reference-frame axis). Subsequently, the interceptor always turns toward an interception point ahead of the target yielding fast interceptions. However, navigation techniques have been designed to bring the interceptor into a collision course with the object, whereas a smooth grasp is necessary in robotic applications. Thus, at a suitable time during the intercept course, a tracking technique must take over the on-line planning of the robot's motion.

The motion-planning algorithms presented in this paper aim at bringing the end-effector of the robot to a pre-grasp location. Interception is assumed to be achieved when the relative position and velocity errors between the object and the end-effector are within user-defined tolerances tol_p and tol_v, respectively:

$$|\boldsymbol{r}| < \mathrm{tol}_p \quad \text{and} \quad |\dot{\boldsymbol{r}}| < \mathrm{tol}_v, \tag{1}$$

where $\boldsymbol{r}$ is the relative position vector between the interceptor and the object.

A schematic diagram of the IPNG moving-object interception system developed in our laboratory is illustrated in Figure 1. The vision module provides real-time object tracking. The object's positional data is fed into a prediction module which provides the motion-planner with one-step-ahead object-state prediction. The motion-planning module uses the latest information about the robot and object states to generate the robot's motion. Finally, the planned motion is executed by the manipulator.

In the following two sections, the object-trajectory prediction and robot-motion planning algorithms will be discussed in greater detail, respectively, as a preamble to the discussion of the implementation issues. Several experiments carried out in our laboratory will be presented subsequently. In these verification tests, it is

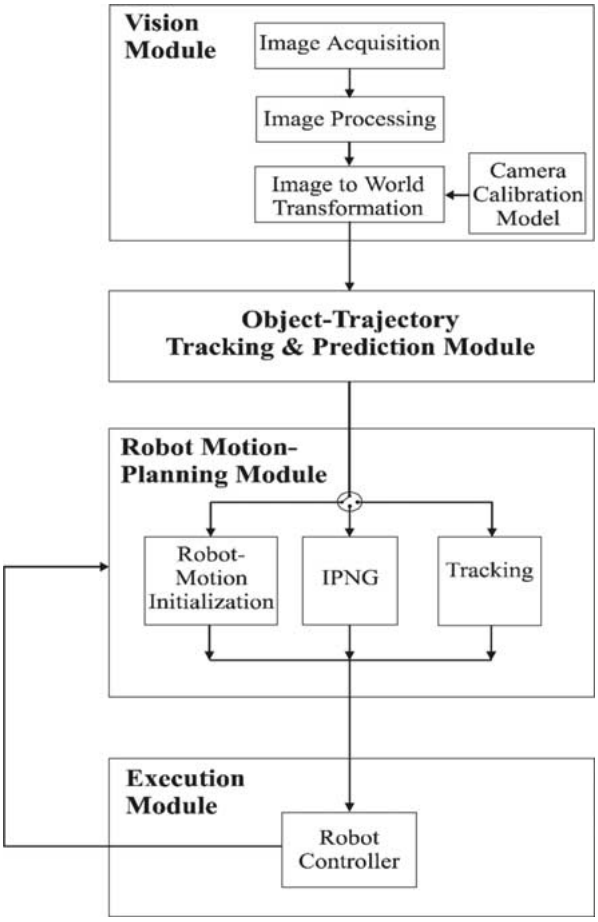

Figure 1. IPNG-based moving-object-interception system.

assumed that an industrial user would not have access to the robot's dynamic model nor to the manipulator arm's joint controllers.

3. Object-Trajectory Prediction

The purpose of the prediction system within the hybrid IPNG-based interception scheme shown in Figure 1 is to provide object tracking and one-step-ahead state prediction as required by the robot-motion planning module. In our laboratory, extensive work has been carried out on the development of a Kalman Filter (KF)-based object-motion-tracking and -prediction system [13, 14]. This prediction system was also utilized in this work. Other prediction techniques proposed in the literature for robotic-interception schema include ARMAX filters (e.g., [5]) and α-β-γ filters (e.g., [15]).

The KF is a computationally-efficient recursive filter that generates an optimal least-squares estimate from a sequence of noisy observations. A new optimal least-squares estimate is obtained from an additional observation without having to re-process past data. The KF can also be used to obtain multiple-step-ahead predictions by propagating the KF extrapolation equation [16].

The 2-dimensional system model for the KF is given as [17]:

$$\boldsymbol{x}_k = \boldsymbol{\Phi}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}, \quad \boldsymbol{w}_k \sim N(0, \boldsymbol{Q}_k), \tag{2}$$

where $\boldsymbol{x}$ is the state vector (which, in our case, is the Newtonian state of the moving object), $\boldsymbol{\Phi}$ is the state transition matrix (which transitions the state vector $\boldsymbol{x}_{k-1}$ at time $k-1$ to the state vector $\boldsymbol{x}_k$ at time k), and $\boldsymbol{w}$ is zero-mean, white Gaussian noise with covariance $\boldsymbol{Q}$, representing the dynamic model driving noise vector. The observation model is given as:

$$\boldsymbol{z}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k, \quad \boldsymbol{v}_k \sim N(0, \boldsymbol{R}_k), \tag{3}$$

where $\boldsymbol{z}$ is the noisy measurement obtained from the vision system, $\boldsymbol{H}$ is the observation matrix relating the observed measurement to the object state and $\boldsymbol{v}$ is zero-mean, white Gaussian noise with covariance $\boldsymbol{R}$, denoting the observation error.

The procedure for the recursive formulation of the KF is carried out in two stages: the prediction stage (Equations (4) and (5)) and the update stage (Equations (6)–(8)):

(i) *State-estimate extrapolation equation* (one-step ahead predictor):

$$\hat{\boldsymbol{x}}_k(-) = \boldsymbol{\Phi}_{k-1}\hat{\boldsymbol{x}}_{k-1}(+). \tag{4}$$

(ii) *Error-covariance extrapolation equation*:

$$\boldsymbol{P}_k(-) = \boldsymbol{\Phi}_{k-1}\boldsymbol{P}_{k-1}(+)\boldsymbol{\Phi}_{k-1}^{\mathrm{T}} + \boldsymbol{Q}_{k-1}, \tag{5}$$

where $\boldsymbol{P}$ denotes the covariance of $\boldsymbol{x}$. (Equation (5) can be used to obtain a measure of the accuracy in predicting $\boldsymbol{x}$ at time k based on the measurements made at time $k-1$ and before.)

(iii) *Gain-matrix update*:

$$\boldsymbol{K}_k = \boldsymbol{P}_k(-)\boldsymbol{H}_k^{\mathrm{T}}\left[\boldsymbol{H}_k\boldsymbol{P}_k(-)\boldsymbol{H}_k^{\mathrm{T}} + \boldsymbol{R}_k\right]^{-1}. \tag{6}$$

The Kalman gain $\boldsymbol{K}_k$ is used to obtain the optimal estimate of the current state, $\hat{\boldsymbol{x}}_k(+)$, from new measurement data $\boldsymbol{z}_k$ and the predicted state $\hat{\boldsymbol{x}}_k(-)$.

(iv) *Filtering equation*:

$$\hat{\boldsymbol{x}}_k(+) = \hat{\boldsymbol{x}}_k(-) + \boldsymbol{K}_k\left[\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k(-)\right]. \tag{7}$$

(v) *System covariance update*:

$$\boldsymbol{P}_k(+) = [\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k]\boldsymbol{P}_k(-). \tag{8}$$

4. Robot-Motion Planning – Theory

The IPNG-based interception scheme, as originally was proposed by Mehrandezh et al. in [12], is briefly outlined below as a preamble to the discussion of its implementation in industrial settings. Herein, an industrial setting refers to the absence of a comprehensive dynamic model of the robot as well as lack of the user's ability to access the joint controllers for torque-level control. (If the user does have both, however, it would be preferable to have a dynamic control of the robot. In this case, the IPNG-based method presented in [12] must be utilized with no approximations and/or simplifications.)

It should be noted that in the proposed hybrid scheme, interception is achieved when the position and velocity of the origin of the end-effector's frame matches the position and velocity of the *object's frame's*⋆ origin. This is a pre-grasping stage interception. Namely, the objective is to bring the robot's end-effector to the closest vicinity of the object at the fastest possible time and subsequently allow a tracking algorithm to carry out the final fine-tuning for grasping.

Herein, and in other interception projects, it is assumed that positional interception would be carried out primarily by the first three joints (of a six degree-of-freedom robot) – the *slow* joints, which could be considered as the bottle-neck operation in comparison to orientation matching. Namely, during robotic object interception, position matching consumes significantly more time than would the concurrent orientation matching process and, thus, an efficient algorithm should be utilized for this purpose, such as the one proposed in this paper. Orientation matching, is not the focus of this paper and could be carried out by any other tracking method proposed in the literature (rather than extending the IPNG-based technique for orientation matching as well).

The proposed on-line, IPNG-based robot-motion planning scheme comprises three sub-phases: motion initialization, navigation guidance and tracking (see Figure 2), which will be individually discussed in the following subsections, respectively. Prior to the discussion of each of these phases, however, the following subsection will first address the basics of the proposed interception scheme.

4.1. BASICS OF NAVIGATION-GUIDANCE-BASED ROBOTIC INTERCEPTION

The IPNG law was developed by Yuan and Chern [18]. In their proposed scheme, the acceleration command turns the relative velocity between the interceptor and the object onto the direction of the LOS with utmost effort. The IPNG acceleration command, applied in the direction normal to the relative velocity, has a magnitude proportional to the product of the LOS rate and the relative velocity:

$$\boldsymbol{a}_{\mathrm{IPNG}} = \lambda \dot{\boldsymbol{r}} \times \dot{\boldsymbol{\theta}}_{\mathrm{LOS}}, \tag{9}$$

⋆ This frame is actually not physically located on the object, but a short distance away from it, in order to prevent any potential collision between the robot's end effector and the object to be grasped.

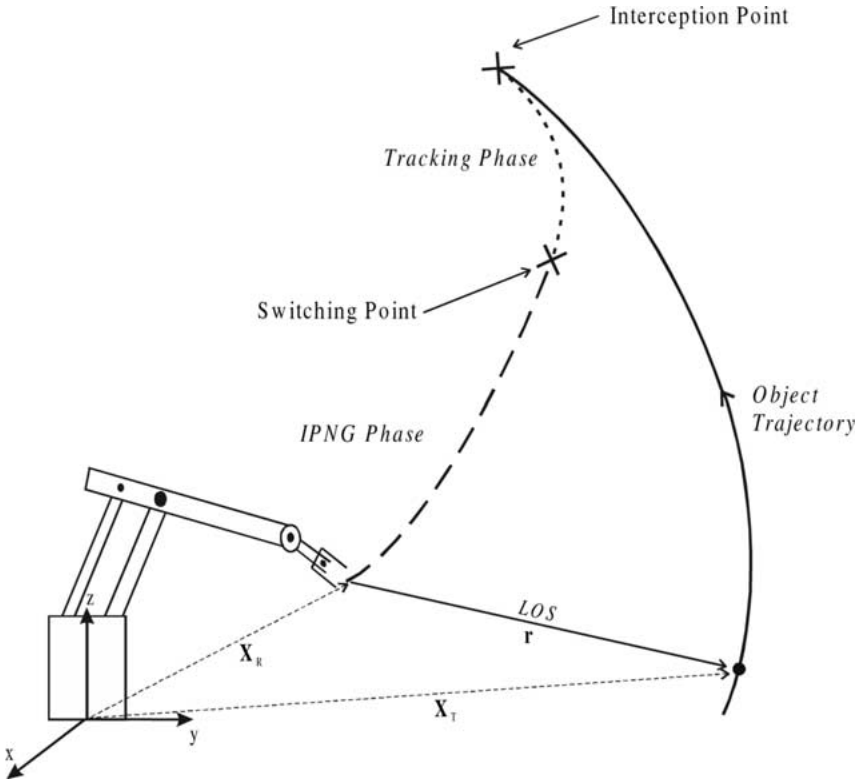

Figure 2. Interception of a moving object via IPNG.

where λ is the navigation gain and $\dot{\boldsymbol{\theta}}_{\mathrm{LOS}}$ is the angular velocity of the LOS. $\dot{\boldsymbol{\theta}}_{\mathrm{LOS}}$ is expressed as:

$$\dot{\boldsymbol{\theta}}_{\mathrm{LOS}} = \frac{\boldsymbol{r} \times \dot{\boldsymbol{r}}}{|\boldsymbol{r}|^2}. \tag{10}$$

By substituting Equation (12) into Equation (11), the IPNG acceleration command can be re-defined as

$$\boldsymbol{a}_{\mathrm{IPNG}} = \frac{\lambda}{|\boldsymbol{r}|^2}\{\dot{\boldsymbol{r}} \times (\boldsymbol{r} \times \dot{\boldsymbol{r}})\}. \tag{11}$$

In [18], it was proven that for IPNG, the capture criterion is $\lambda > 1$, no matter what the initial condition of $\dot{\boldsymbol{r}}$ and object maneuver are. Moreover, $\dot{\boldsymbol{\theta}}_{\mathrm{LOS}}$ approaches infinity if $\lambda < 2$, whereas (for non-maneuvering objects) $\dot{\boldsymbol{\theta}}_{\mathrm{LOS}}$ approaches zero when $\lambda > 2$.* Thus, λ should be chosen to be greater than 2, for successful interception.

* The optimal value for λ is computed by optimizing an objective function that takes into account the energy expended during the interception on top of the interception time. For IPNG, this yields the range $3 < \lambda < 5$ for optimal performance. However, since in robotic manipulators the expended energy is not crucial, the range for λ can be relaxed.

For robotic interception, the acceleration command of the IPNG (in task space, for the robot's end effector) can be modified, as will be discussed below, to reflect the maneuvering superiority of a robotic manipulator over an airborne interceptor, though, subject to the robot's dynamic constraints

$$\left|T_i\right| \leqslant \left|T_{i_{\max}}\right|, \quad i = 1, \ldots, n, \tag{12}$$

where $T_{i_{\max}}$ is the maximum torque available in the ith actuator.

4.2. ROBOT-MOTION INITIALIZATION

From Equation (11), it can be noted that, if the manipulator is initially at rest, the initial acceleration command will be in a direction perpendicular to the object's velocity. This is clearly non-optimal. Thus, a motion initialization scheme is required prior to the start of the IPNG phase.

The robot motion can be initiated using the maximum permissible acceleration of the robot towards the interception point. However, since the interception point is not known a priori, it is proposed herein to initialize the robot in the direction of the LOS, while taking into account the direction of the object's velocity $\boldsymbol{e}_{\dot{X}_T}$, Figure 3:

$$\boldsymbol{e}_d = \frac{k\boldsymbol{e}_r + \boldsymbol{e}_{\dot{X}_T}}{|k\boldsymbol{e}_r + \boldsymbol{e}_{\dot{X}_T}|}, \tag{13}$$

where $\boldsymbol{e}_d$ represents the direction of motion initialization, $\boldsymbol{e}_r$ is a unit vector along the LOS, $\boldsymbol{e}_{\dot{X}_T}$ is a unit vector along the object's velocity and k is a pre-defined coefficient. (The selection of $k > 1$ prevents $\boldsymbol{e}_d$ from collapsing to zero, when the direction of the object's velocity is directly opposite to $\boldsymbol{e}_r$.) A more conservative approach could be the use of a higher value of k ($k \geqslant 3$), to place more emphasis

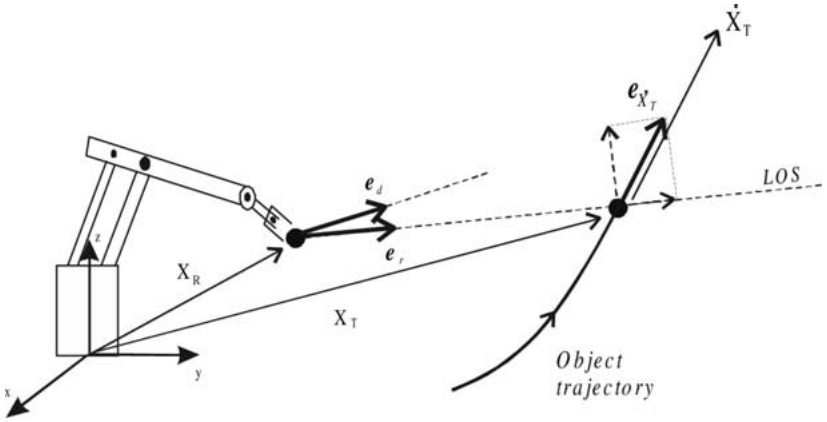

Figure 3. Direction of robot-motion initialization.

on initializing the robot towards the current location of the object with less consideration to the direction of motion of the object. This would be a safer strategy, in case the object maneuvers and changes completely its direction of motion. If the relative velocity is turned onto the LOS during motion initialization (i.e., if $|\dot{\boldsymbol{\theta}}_{\mathrm{LOS}}| <$ threshold), the direction of motion initialization is set as:

$$\boldsymbol{e}_d = \boldsymbol{e}_r \tag{14}$$

(i.e., along the LOS), for the remaining duration of robot-motion initialization.

Robot motion-initialization is terminated when either the robot reaches its maximum permissible velocity or when a pre-defined time threshold has passed.

4.3. NAVIGATION-GUIDANCE PHASE

IPNG was originally developed for airborne interception, where the interceptor is normally assumed to be capable of maneuvering in a direction normal to its current direction of motion. However, the end effector of a robotic manipulator can be maneuvered in any direction, given that the dynamic constraints of the actuators are not violated. To reflect this superiority, the acceleration command of the robotic interceptor should be upgraded while maintaining the torques within a pre-defined percentage α of their maximum values (for $0 < \alpha < 1$). The IPNG acceleration command is boosted up herein by adding a component in the LOS direction, i.e.,

$$\boldsymbol{a}_c = \boldsymbol{a}_{\mathrm{IPNG}} + \beta \boldsymbol{e}_r, \tag{15}$$

where β is a constant that is computed on-line, such that,

$$\left|T_i\right| \leqslant \alpha \left|T_{i_{\max}}\right|, \quad i = 1, \ldots, n. \tag{16}$$

It should be noted that, the effect of upgrading the acceleration command along the LOS is to increase the closing velocity.

Conversely, if the dynamic constraints of the manipulator are violated, the IPNG acceleration command must be limited. The acceleration command can be limited by scaling $\boldsymbol{a}_{\mathrm{IPNG}}$, i.e., reducing its magnitude while keeping its original direction,

$$\boldsymbol{a}_c = K \boldsymbol{a}_{\mathrm{IPNG}}. \tag{17}$$

In Equation (17) above, the coefficient K is determined on-line, such that, given the current robot configuration, none of the actuator limits is exceeded.

4.4. TRACKING PHASE AND PHASE SWITCHING

As discussed above, in the proposed hybrid interception scheme, the IPNG-based motion-planning brings the robot to the vicinity of the object as fast as possible (namely, the robot's end effector is in a collision course with object). A tracking technique is, thus, necessary to slow the robot and ensure velocity matching at

the interception point. (A commonly used trajectory-tracking technique for the control of robotic manipulators is the Computed-Torque (CT) scheme [19]. The basic concept employed by the CT scheme is to achieve *dynamic decoupling* of all joints using *nonlinear feedback*.)

For time-optimal interception, the tracking method must take over control of robot-motion planning at an optimal time instant. Therefore, while the robot is under IPNG control, the decision whether to switch to tracking control or proceed with IPNG must be considered at each planning instant. In order to do this, the decision mechanism must be capable of evaluating the overall interception time t_{int} if a switch would occur at that instant. The overall interception time t_{int} is defined in this paper as:

$$t_{\text{int}} = t_{\text{IPNG}} + t_{\text{tracking}}, \tag{18}$$

where t_{IPNG} and t_{tracking} denote the times during which the robot is under IPNG control and tracking control, respectively. (The time during which the robot is being initialized is included in t_{IPNG}.)

Normally, t_{int} is a single minimum curve, Figure 4. The problem at hand, therefore, is to determine on-line when the tracking technique should take over the control of the on-line planning of the robot's motion for minimum-time interception. (However, if the curve is not a single-minimum curve, as in every other optimization case, a switch could happen at a local-optimal point. Though, one can appreciate that even a local optimal, with some IPNG-phase at the robot's initial motion stage, is better than a pure tracking process, where the switch is carried out at time zero.)

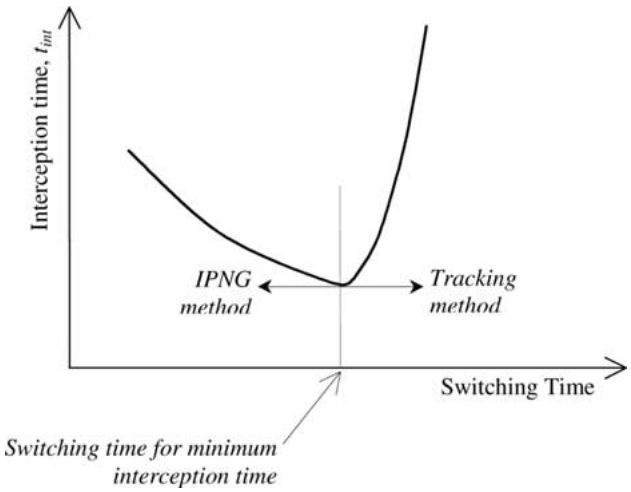

Figure 4. Interception time versus the switching time.

5. Implementation Using an Industrial Robotic System

The implementation of the proposed IPNG-based interception scheme using an industrial robotic system would normally face two primary challenges: lack of availability of the robot's dynamic model, as well as, inaccessibility to the joint controllers. In response to these challenges, several modifications to the robot-motion planning schema are presented in this section. In addition, numerous issues related to the development of the four software modules, previously introduced in Section 2, are discussed, Figure 1.

Since our current experimental test-bed includes a GMFanuc S-100 industrial manipulator with its standard Karel Controller, an NC XY-table used for object motion and a computer vision system, some references to these are also made during the discussion of the software modules.

The GMFanuc S-100 is a six degree-of-freedom manipulator, with primarily three revolute joints to govern end-effector translation, and three revolute joints mounted at the wrist for orientation. The Karel controller performs the joint-level control of the manipulator, to which the user does not have access.

5.1. VISION MODULE

The function of the vision module is to provide real-time object tracking. However, since our emphasis is on visual-tracking in an industrial setting, a semi-structured environment can be assumed, where certain simple geometric features on the object, such as a planar circular marker in our case, rather than the object itself, is tracked.

In our experiments, the white circular marker is displaced on a darker background, permitting target localization in the scene by using gray-level thresholding and centroid computation. The threshold value utilized to convert the gray-scale images into binary is a priori determined experimentally. The windowing approach (e.g., [2]) was adopted, whereby only a small rectangular window located at the current marker location is processed to save time. The window is multiplied in size if the target is not located within the window. Once the centroid of the target has been located in the image plane, its coordinates are transformed into the world coordinate system of the robot. This is achieved using a priori knowledge of the camera-calibration model [20].

5.2. PREDICTION MODULE

The object's positional data obtained from the vision module can be efficiently processed by a recursive KF algorithm. In modeling our system, the noise in the x and y directions is assumed to be independent and, thus, the x and y states are decoupled. A first-order Gauss–Markov jerk motion model, as proposed by Singer [21], is used for each direction. The movement of the object is represented by a

point mass undergoing random acceleration:

$$\dot{p}_x = v_x, \qquad \dot{p}_y = v_y, \tag{19}$$

$$\dot{v}_x = a_x, \qquad \dot{v}_y = a_y, \tag{20}$$

$$\dot{a}_x = -\frac{1}{\tau}a_x + \sqrt{\frac{2}{\tau}}\sigma_x V_x, \qquad \dot{a}_y = -\frac{1}{\tau}a_y + \sqrt{\frac{2}{\tau}}\sigma_y V_y, \tag{21}$$

where p, v and a are the object's planar position, velocity and acceleration, respectively. τ is the acceleration decorrelation time (i.e., a measure of how quickly the object changes its trajectory) and V is Gaussian white noise with variance σ.

5.3. ROBOT-MOTION PLANNING MODULE

The problem of autonomous robot motion has been traditionally divided into two separate tasks: *trajectory planning* and *robot control*, whereby the latter is designed to reliably follow the trajectory planned by the former, e.g., [22]. The IPNG-based robot-trajectory planning schema presented in Section 4 may need to be modified for the effective control of industrial robots. In the absence of the robot's dynamic model, the function of our robot-motion planning module is to compute the desired trajectory setpoints of the end-effector based on the IPNG method. The robot controller would then utilize these setpoints to produce the inputs to the manipulator so that the end-effector tracks the desired path, Figure 5.

One must recall that the motion-planning algorithms presented herein aim at bringing the robot to a pre-grasping position. Once the end-effector is at a pre-

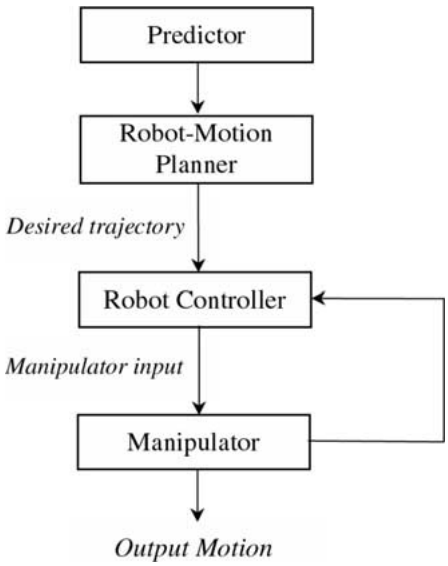

Figure 5. Architectural organization of an industrial robot control.

grasping position, it is assumed that a proximity-sensor-based tracker would take over control until the object is successfully grasped.

5.3.1. *Motion Initialization*

The robot's motion can be initialized as described in Section 4.3. The motion of the end-effector is defined by a displacement vector $\boldsymbol{d}$ in the direction defined by Equation (13). The magnitude of $\boldsymbol{d}$ is computed using the maximum permissible acceleration of the end-effector, A_{lim}.

5.3.2. *Navigation-Guidance Phase*

In IPNG, the calculation of the acceleration command is carried out using Equation (11). Herein, the implementation problem at hand is to determine the end-effector's future (desired) position as a result of applying the computed IPNG acceleration command for one motion interval Δt. Due to the possible absence of the robot's jerk-limit specifications and since Δt is normally small, a constant-acceleration model can be used to determine this next setpoint for the robot (i.e., $\boldsymbol{X}_R$ at $t = t_{k+1}$) as follows:

$$\boldsymbol{X}_R(t) = (\boldsymbol{X}_R)_k + \left(\dot{\boldsymbol{X}}_R\right)_k (t - t_k) + \frac{\boldsymbol{a}_{\text{IPNG}}}{2}(t - t_k)^2, \quad t \in [t_k, t_{k+1}]. \tag{22}$$

The corresponding robot velocity at $t = t_{k+1}$ is calculated as

$$\left(\dot{\boldsymbol{X}}_R\right)_{k+1} = \left(\dot{\boldsymbol{X}}_R\right)_k + \boldsymbol{a}_{\text{IPNG}}\Delta t. \tag{23}$$

As discussed in Section 4, the IPNG acceleration command may be modified to maximize the utilization of the manipulator's mobility, without exceeding its specified limits. In the absence of the robot's dynamic model, the manipulator's (task-space) kinematic constraints are utilized herein: the maximum end-effector velocity and acceleration limits V_{lim} and A_{lim}, respectively. These limits must be satisfied simultaneously.

(i) Upgrading the acceleration command. The IPNG acceleration command is upgraded, if both its magnitude and the magnitude of robot velocity, $(\dot{\boldsymbol{X}}_R)_{k+1}$, are less than the specified percentages α_A and α_V, respectively, of their maximum values. This is achieved by adding a component in the direction of the LOS, $\boldsymbol{e}_r$, as described in Equation (15). β is computed such that

$$\left|\left(\dot{\boldsymbol{X}}_R\right)_{k+1}\right| \leqslant \alpha_v V_{\text{lim}} \quad \text{and} \quad |\boldsymbol{a}_c| \leqslant \alpha_A A_{\text{lim}}. \tag{24}$$

(ii) Limiting the acceleration command. The IPNG acceleration is limited, if either its magnitude or the magnitude of the end-effector velocity would exceed their respective limits during the execution of the $\boldsymbol{a}_{\text{IPNG}}$. A sequential process can

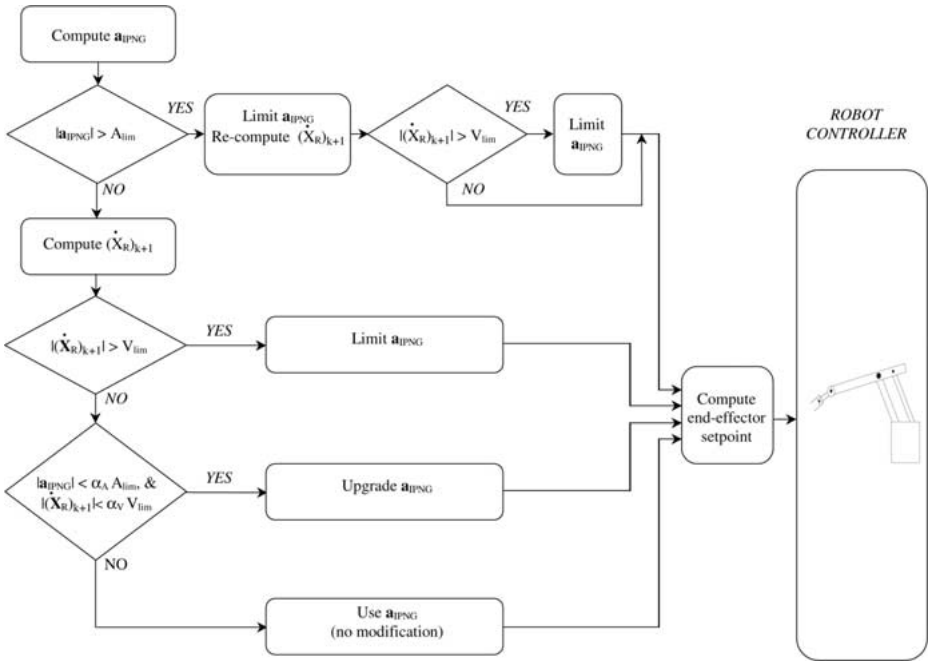

Figure 6. Modifying the $\boldsymbol{a}_{\text{IPNG}}$.

be utilized to achieve this objective: If the acceleration limit is violated, the $\boldsymbol{a}_{\text{IPNG}}$ is limited by simply scaling down its magnitude,

$$\boldsymbol{a}_c = \frac{A_{\text{lim}}}{|\boldsymbol{a}_{\text{IPNG}}|}\boldsymbol{a}_{\text{IPNG}}. \tag{25}$$

Subsequently, it is checked whether the end-effector velocity at $t = t_{k+1}$ is expected to exceed the speed limit V_{lim}. If the answer is yes, the robot velocity magnitude is scaled down to V_{lim}. This amounts to the utilization of an acceleration command that produces a final velocity at $t = t_{k+1}$ in the same direction as that which would be produced by $\boldsymbol{a}_{\text{IPNG}}$; however, with a smaller (permissible) magnitude.

The overall algorithm for modifying the $\boldsymbol{a}_{\text{IPNG}}$ for an industrial robot, such as the GMFanuc S-100, is depicted in Figure 6.

5.3.3. *Tracking Phase*

In the absence of torque-control capability, object tracking can be achieved using point-to-point robot-trajectory generation, (e.g., [5]), based on the current state of the robot and the object's one-step-ahead predicted state. Task-space quintic polynomials have been used in the past for real-time applications (e.g., [14]). Quintic polynomials are inherently one-dimensional and, thus, three quintic polynomials are necessary to describe the robot's 3D positional trajectory.

In the proposed hybrid interception scheme, tracking is employed only in the last phase of motion in order to decelerate the robot. Thus, only short-term quin-

tic polynomial trajectories would be utilized. However, in order to determine the coefficients of such quintic polynomial trajectories, the motion time is required. For time-optimal quintic motion trajectory, the motion time must be minimized subject to the velocity and acceleration limits. Minimum motion time is achieved when either V_{lim} or A_{lim} is reached at some point during the planned trajectory. The problem can, thus, be formulated as:

Given the current state of the robot and the predicted state of the object one motion-time step ahead, and given the velocity and acceleration limits of the robot, V_{lim} *and* A_{lim}*, find the minimum quintic polynomial trajectory time* t_{qp}:

$$t_{\text{qp}} = f(V_{\text{lim}}, A_{\text{lim}}). \tag{26}$$

Since the initial and final boundary conditions are not stationary in our case, a numerical-solution algorithm needs to be used to calculate the t_{qp} [14].

5.3.4. *Switching from IPNG to QP Tracking*

For time optimal interception, an on-line selection of the optimal instant at which motion-planning should be switched from IPNG to QP tracking is necessary. A predictive approach can be used when using a QP Tracker (QPT). However, such an approach is not suitable for the IPNG-based interception scheme, since IPNG is targeted for maneuvering objects, whose long-term trajectory cannot be reliably predicted without a priori information. A near-optimal solution can be adopted instead, in which the current quintic-trajectory motion time t_{qp} is used as the criterion for switching, Figure 7.

This approach is based on the assumption that IPNG is superior to the tracking method in bringing the robot towards the object and, thus, tracking is utilized only in the final portion of the intercept period to decelerate the robot to the current state of the object. During the IPNG phase of motion-planning, the value of t_{qp} is computed at each planning instant (using Equation (26)), based on the (current) end-effector state and (one step-ahead) object state. As the robot closes the distance to the object, the value t_{qp} would normally decrease in tandem with the (real) interception time, whose value cannot be accurately estimated. As is the case of the t_{int} curve, there would be a sudden increase in t_{qp}, if the planned quintic trajectory overshoots the target trajectory, Figure 7.

The minimum point of the t_{qp} curve (e.g., t_2 in Figure 7) can, thus, be utilized to determine the switching point in an on-line manner. (In order to reduce the effect of noise, the values of t_{qp} computed on-line may be smoothed using a recursive exponential smoothing technique [23].)

5.4. EXECUTION MODULE

The robot-motion planner sends trajectory set-points to the execution module, which is normally hosted on the robot's dedicated controller (such as the Karel

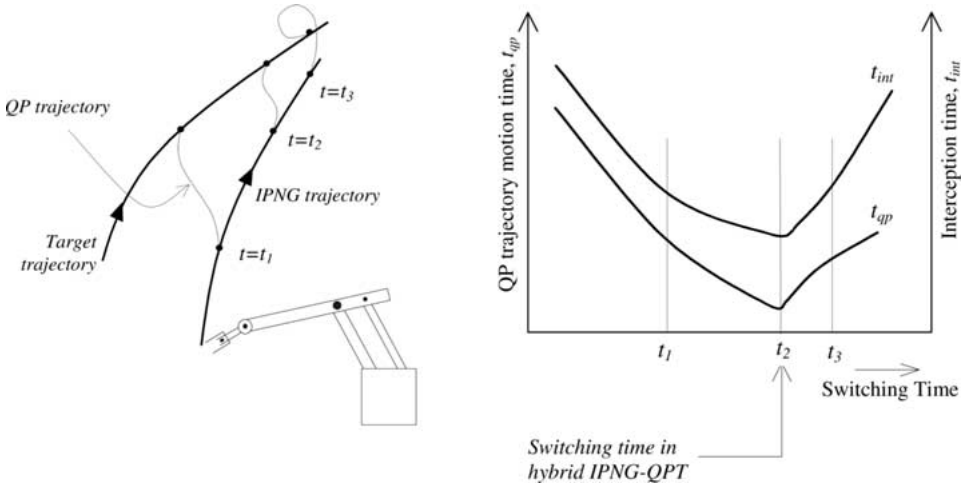

Figure 7. Selection of switching instant.

controller for the GMFanuc S-100). The controller would, then, be instructed to perform joint-interpolated, continuous motion between successive end-effector setpoints. In our case, trajectory setpoints are provided by the motion-planner at the Karel controller's request (at the minimum possible ∼220 ms intervals).

6. Experiments

6.1. EXPERIMENTAL SYSTEM CONFIGURATION

The experimental IPNG-QPT-based moving-object interceptor test-bed in our laboratory comprises three primary subsystems, Figure 8.

Within the object-tracking subsystem, tracking is achieved using a (Hitachi 30 Hz) CCD camera (equipped with a Canon 25 mm lens) and a (Matrox 640B) frame grabber, which acquire 640×480 pixel digital images with 256 grey-level resolution. The host is an Intel-CPU-based PC. The CCD camera is set up in a fixed-camera configuration. It is mounted at 1.75 m above the plane of motion of the target, at an angle of approximately 8° with the normal to the plane. The current configuration yields a field of view of approximately 0.47×0.58 m^2 on the object's motion plane, which corresponds to a resolution of approximately 0.98 mm/pixel along the x direction of object motion, and 0.9 mm/pixel along the y direction of motion. On the object-tracking subsystem, the process of acquiring an image and processing it to generate fresh world coordinates of the object's centroid utilizes approximately 66 ms.

The interception-planning subsystem is also an Intel-CPU-based PC. This subsystem hosts both the target-motion prediction and the robot-motion planning modules. This enables the predictor module to efficiently provide the planner module with estimated object motion at the latter's request. Whenever fresh object data

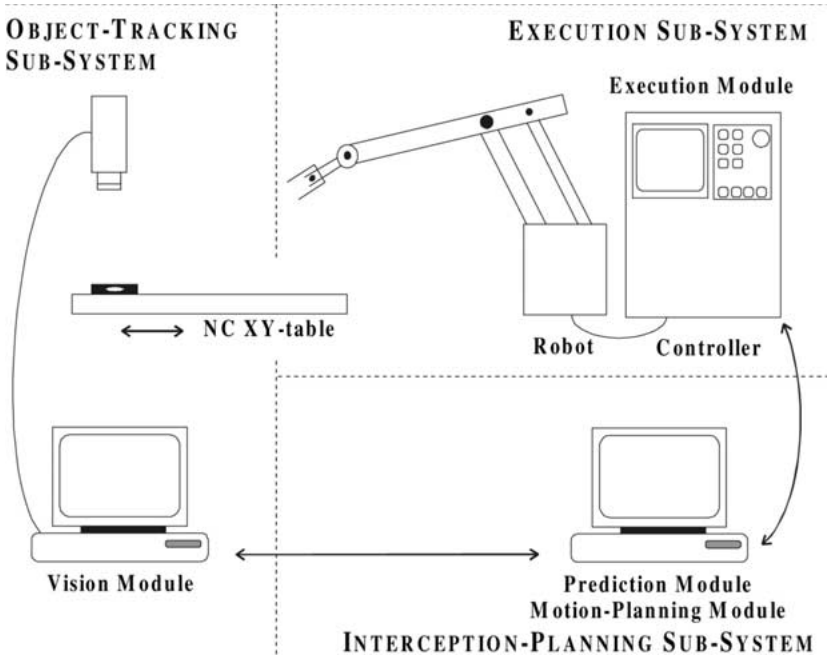

Figure 8. Architecture of the experimental test-bed.

becomes available, the object's trajectory estimates are updated by the KF and the end-effector setpoints are regenerated using the most recent object-trajectory data.

The execution subsystem is the GMFanuc S-100 robotic system. The GMFanuc robot is controlled by the Karel controller, which runs its operating system [24]. Programs are written in the Karel high-level language to control the operation of the robot's end-effector. The controller allows serial communication with a host-computer during the execution of a program. This feature is used herein to instruct the robot to execute real-time trajectories generated by the interception-planning subsystem. The latter provides the Karel controller with the trajectory set-points using 9.600 baud serial communication, the fastest rate which the controller permits.

6.2. EXPERIMENTAL RESULTS

The experimental set-up discussed above was employed to verify the proposed implementation of the hybrid IPNG-QPT method to intercept moving objects. Due to the velocity limitation of the NC XY-table ($\sim$35 mm/s maximum), via which the object is displaced, the robot's speed limit was set at a much smaller value than its actual permissible limit. The tolerances of interception (i.e., achievement of a pre-grasping robot state), tol_p and tol_v, were set at 10 mm and 10 mm/s, respectively.

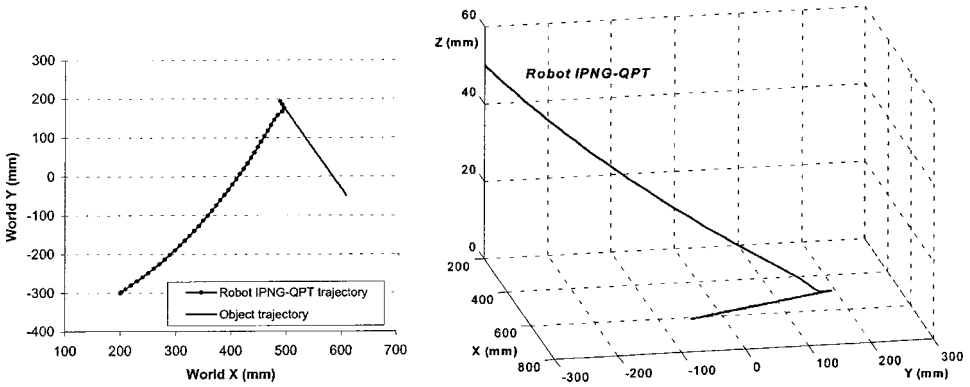

Figure 9. Interception of an object moving on a linear trajectory.

EXAMPLE 1 (Non-maneuvering object motion). Figure 9 shows the interception of a non-maneuvering object moving on a planar linear trajectory with a cruising speed of approximately 30 mm/s. The robot velocity and acceleration limits were correspondingly set at 70 mm/s and 150 mm/s^2. The end-effector was initially at rest at $[200–300\ 50]^{\mathrm{T}}$ mm.

The IPNG-QPT method yielded an interception time of 9.6 s for a switching time of 8 s, whereas the QPT method would have yielded 13.1 s (for a switching time of 0 s). The x, y and z components of the position of the robot's end-effector under IPNG-QPT control (as sent by the planner to the robot controller) and of the object (as received from the object-tracking subsystem) are shown in Figure 10. Figure 10 also shows the velocity components of the object (estimated via the KF) and of the end-effector. The variation of the interception time versus the switching time is shown in Figure 11.

EXAMPLES 2 and 3 (Maneuvering object motion). In the following two examples, the interception of a maneuvering object is considered. Figure 12 shows the interception of an object which initially moves along a circular trajectory, then, stops and moves along a (different) circular trajectory in the opposite direction. For this example, the IPNG-QPT yielded an interception time of 9.1 s (for a switching time of 7.6 s), whereas the QPT method would have yielded an interception time of 12.5 s. Figure 13 shows the x, y and z components of the position and velocity of the robot's end-effector under IPNG-QPT control and of the object.

Figure 14 shows the variation of the interception time versus the switching time. It can be noted that switching at 5.5 s would have yielded a (global-optimal) interception time of 8.5 s, whereas the automatic switching algorithm yielded an interception time of 9.1 s (at a switching time of 7.6 s, i.e., at the minimum value of the t_{qp} curve). However, the interception time obtained by the automatic switching method, though not globally optimal, still represents a significant decrease from the interception time which the pure QPT method would have yielded (i.e., 12.5 s).

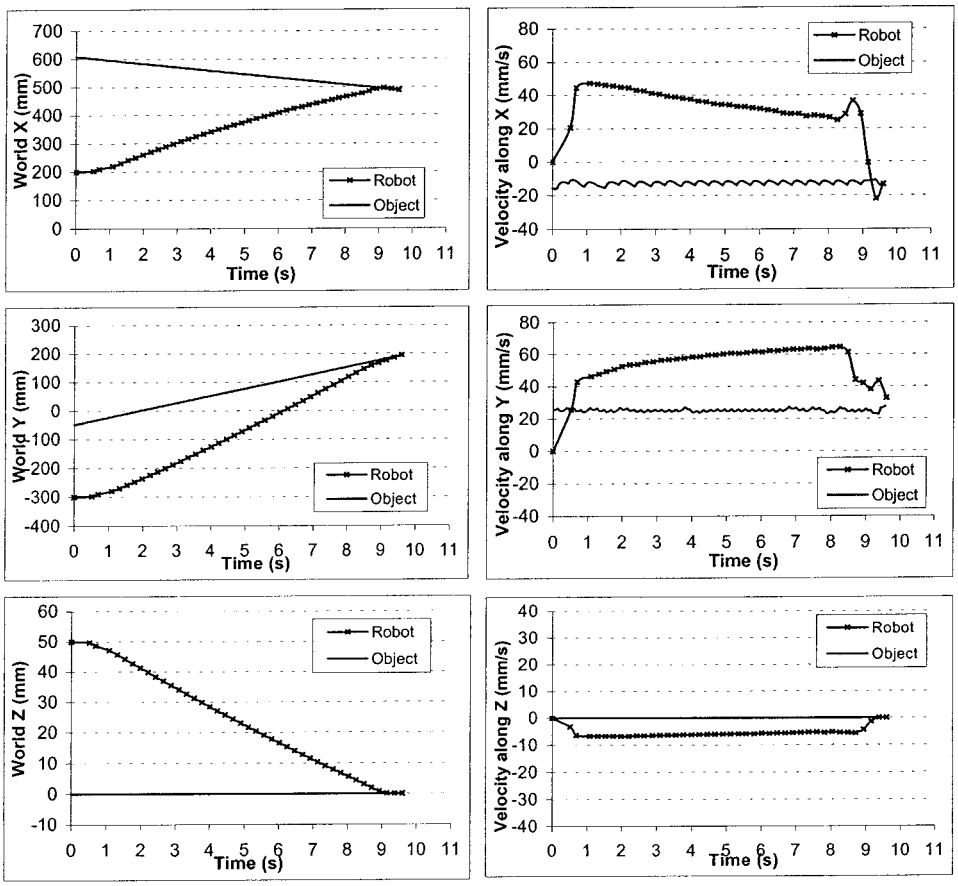

Figure 10. Position and velocity variation versus time for the end-effector and the target.

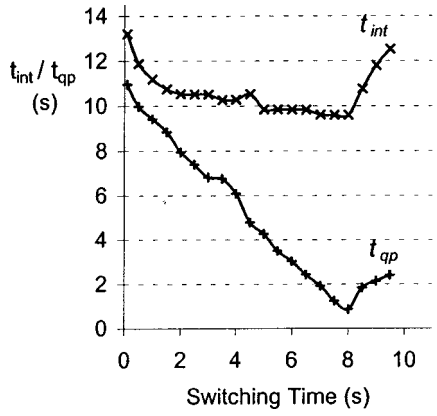

Figure 11. Variation of t_{int} and t_{qp} versus switching time.

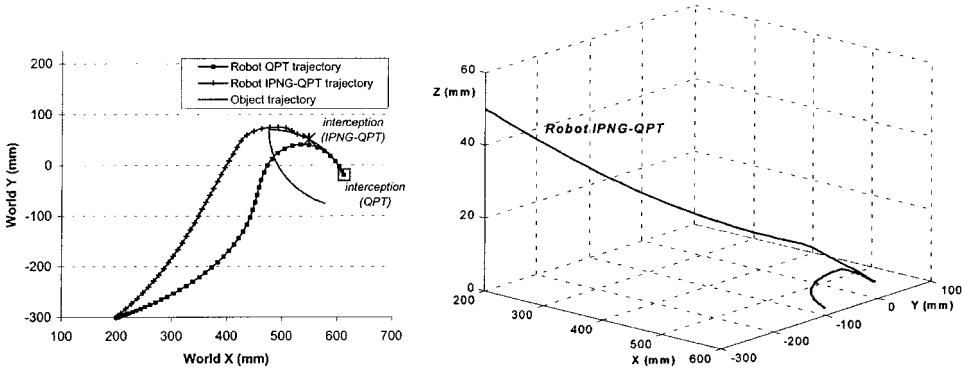

Figure 12. Interception of a target moving along a discontinuous circular trajectory.

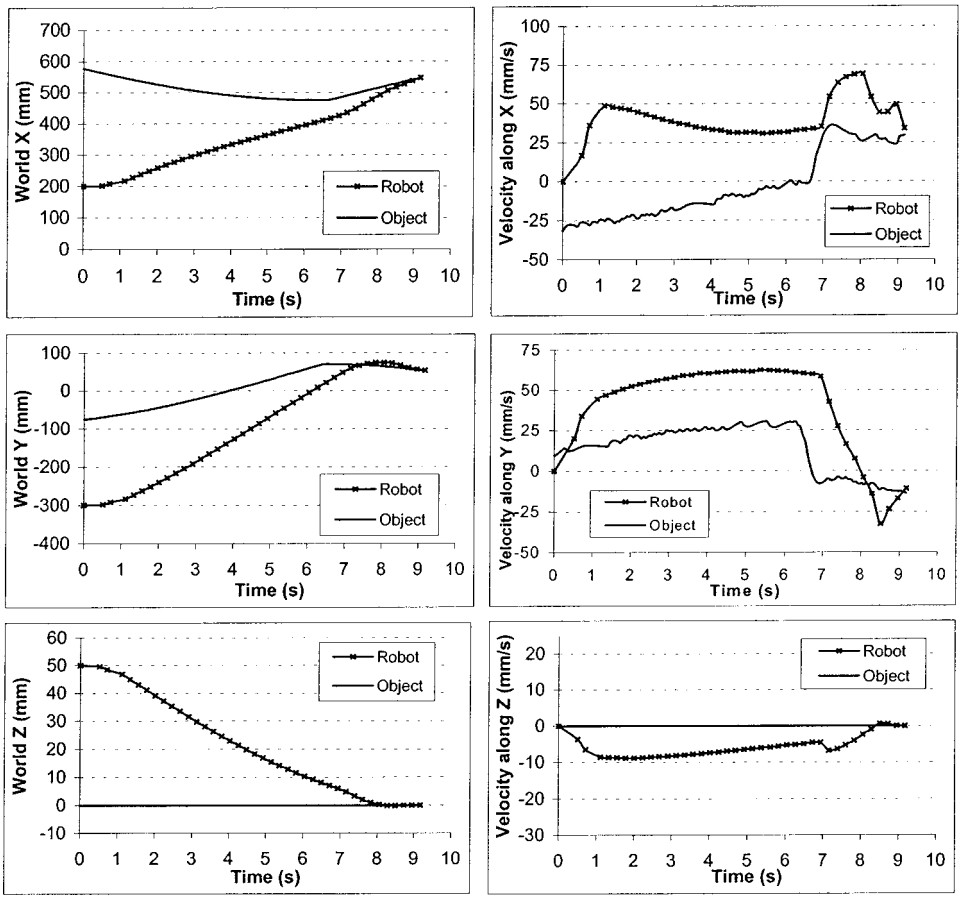

Figure 13. Position and velocity variation versus time for the end-effector and the object.

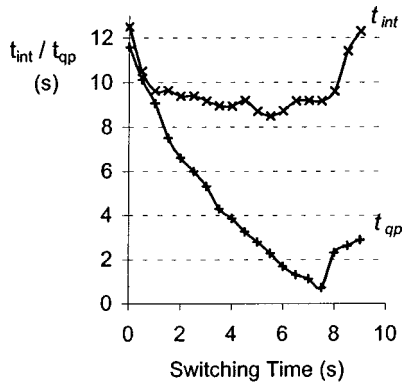

Figure 14. Variation of t_{int} and t_{qp} versus switching time.

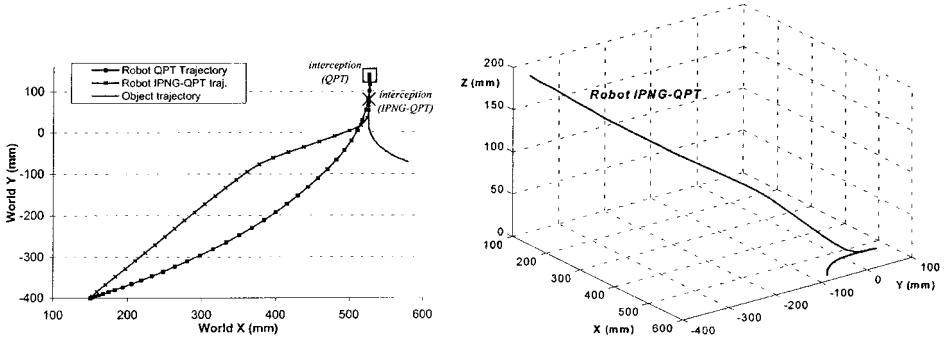

Figure 15. Interception of an object travelling along a stop-and-go trajectory.

Figures 15 and 16 depict the interception of an object travelling along a stop-and-go trajectory: the object initially moves along a circular path, it then decelerates rapidly to rest, and, after being at rest for 1.5 s, it accelerates rapidly along a linear path. For this example, the robot's velocity and acceleration limits were set at 120 mm/s and 200 mm/s^2. Using the IPNG-QPT method, interception occurred at 7.0 s, whereas the pure QPT method would have yielded an interception time of 9.1 s.

7. Conclusions

In this paper, the implementation of a navigation-based interception scheme for moving-object interception has been presented. The interception scheme is a hybrid method, combining a navigation-based technique with a conventional tracking method. An industrial robotic system has been utilized in the implementation. The experimental results confirm that the hybrid interception scheme could yield faster interception times than would a pure (visual-servoing type) tracking method.

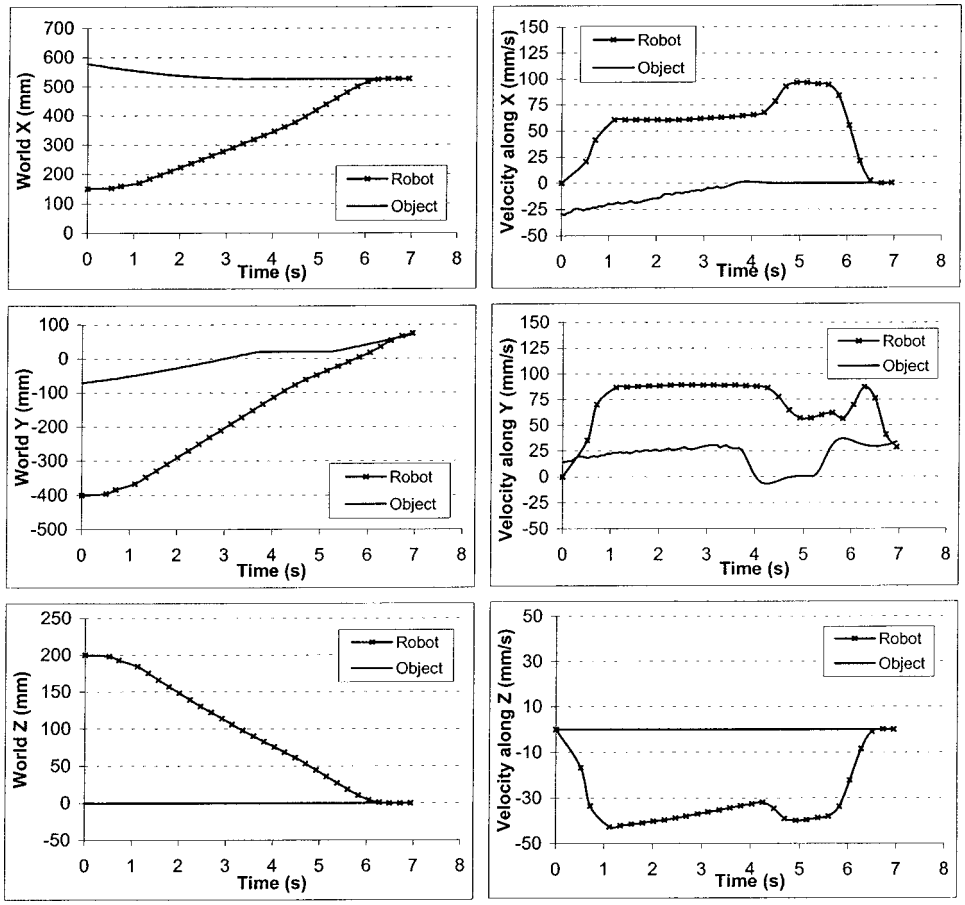

Figure 16. Position and velocity variation versus time for the end-effector and the object.

References

1. Kimura, H., Mukai, N., and Slotine, J. E.: Adaptive visual tracking and Gaussian network algorithm for robotic catching, *ASME Advances in Robust and Nonlinear Control Systems* **43** (1992), 67–74.
2. Buttazzo, G. C., Allotta, B., and Fanizza, F. P.: Mousebuster: A robot for real-time catching, *IEEE Control Systems Mag.* **14**(1) (1994), 49–56.
3. Fernandes, D. G. and Lima, P. U.: A testbed for robotic visual servoing and catching of moving objects, in: *IEEE Internat. Conf. on Electronics, Circuits and Systems*, Lisboa, Portugal, 1998, pp. 475–478.
4. Croft, E. A., Fenton, R. G., and Benhabib, B.: Time-optimal interception of objects moving along predictable paths, in: *IEEE Internat. Symp. on Assembly and Task Planning*, Pittsburgh, 1995, pp. 419–425.
5. Houshangi, N.: Control of a robotic manipulator to grasp a moving target using vision, in: *IEEE Internat. Conf. on Robotics and Automation*, Cincinnati, OH, 1990, pp. 604–609.

6. Kelly, R., Reyes, F., Moreno, J., and Hutchinson, S.: A two-loop direct visual control of direct-drive planar robots with moving target, in: *IEEE Internat. Conf. on Robotics and Automation*, Detroit, MI, 1999, pp. 599–604.
7. Hashimoto, K., Kimoto, T., Ebine, T., and Kimura, H.: Manipulator control with image-based visual servo, in: *IEEE Internat. Conf. on Robotics and Automation*, Sacramento, CA, 1991, pp. 2267–2271.
8. Lin, Z., Zeman, V., and Patel, R. V.: On-line robot trajectory planning for catching a moving object, in: *IEEE Internat. Conf. on Robotics and Automation*, Scottsdale, AZ, 1989, pp. 1726–1731.
9. Yang, C. D. and Yang, C. C.: A unified approach to proportional navigation, *IEEE Trans. Aerospace Electronic Systems* **33**(2) (1997), 557–567.
10. Piccardo, H. R. and Hondered, G.: A new approach to on-line path planning and generation for robots in non-static environment, *J. Robotics Autonom. Systems* (1991), 187–201.
11. Su, J. and Xi, Y.: Path planning for robotic hand/eye system to intercept moving objects, in: *IEEE Conf. on Decision and Control*, Phoenix, AZ, 1999, pp. 2963–2968.
12. Mehrandezh, M., Sela, M. N., Fenton, R. G., and Benhabib, B.: Robotic interception of moving objects using ideal proportional navigation guidance technique, *J. Robotics Autonom. Systems* **28**(4) (1999), 295–310.
13. Zak, G., Hexner, G., Croft, E. A., Benhabib, B., and Fenton, R. G.: A prediction based strategy for robotic interception of moving objects, in: *IEEE Canadian Conf. on Electrical and Computer Engineering*, Vancouver, BC, 1993, pp. 1069–1072.
14. Hujic, D., Croft, E. A., Zak, G., Fenton, R. G., Mills, J. K., and Benhabib, B.: The robotic interception of moving objects in industrial settings: Strategy development and experiment, *IEEE/ASME Trans. Mechatronics* **3**(3) (1998), 225–239.
15. Allen, P. K., Yoshimi, B., and Timcenko, A.: Real time visual servoing, in: *IEEE Internat. Conf. on Robotics and Automation*, Sacramento, CA, 1991, pp. 851–856.
16. Harvey, A. C.: *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge Univ. Press, New York, 1989.
17. Gelb, A., Kasper, J. F., Nash, R. A., Price, C. F., and Sutherland, A. A.: *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.
18. Yuan, P. J. and Chern, J. S.: Ideal proportional navigation, *J. Guidance Control Dynamics* **15**(5) (1992), 1161–1165.
19. Khosla, P. K. and Kanade, T.: Experimental evaluation of nonlinear feedback and feedforward control schemes for manipulators, *J. Robotics Res.* **7**(1) (1988), 18–28.
20. Tsai, R. Y.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the shelf TV cameras and lenses, *IEEE J. Robotics Automat.* **3**(4) (1987), 323–344.
21. Singer, R. A.: Estimating optimal tracking filter performance for manned maneuvering targets, *IEEE Trans. Aerospace Electronic Systems* **6**(4) (1970), 473–483.
22. Koivo, A. J.: *Fundamentals for Control of Robotic Manipulators*, Wiley, New York, 1989.
23. Brown, R. G.: *Smoothing, Forecasting and Prediction of Discrete-Time Series*, Prentice-Hall, Englewood Cliffs, NJ, 1963.
24. GMF Robotics Corporation, *Karel System Reference Manual*, GMF Robotics Corporation, Auburn Hills, MI, 1987.