The Robotic Interception of Moving Objects in Industrial Settings: Strategy Development and Experiment

Damir Hujic, *Associate Member, IEEE*, Elizabeth A. Croft, *Member, IEEE*, Gene Zak, Robert G. Fenton, James K. Mills, *Member, IEEE*, and Beno Benhabib, *Member, IEEE*

***Abstract*— A novel active prediction, planning, and execution (APPE) system is presented herein for the robotic interception of moving objects. The primary feature of the proposed APPE system is the ability to intercept the object at an optimal rendezvous point, anywhere along its predicted trajectory, within the robot's workspace. For the interception of objects in industrial settings, the motion of which allows long-term predictability, this feature is a significant improvement over earlier APPE systems. These could only select a rendezvous point among a few nonoptimal interception points considered. An APPE system's objective is simply to move the robot to the earliest pregrasping location. A fine-motion tracking algorithm can take over the motion control at that point, utilizing proximity sensors mounted on the robot's end-effector. This approach eliminates the necessity of tracking the motion of the object, as required by conventional tracking-based techniques, where the distance between the robot's end-effector and the object is reduced continuously. In this paper, the proposed APPE system is first briefly introduced, and its individual modules are thereafter discussed in detail. Simulation and experimental results are presented in support of the developed optimal-interception strategy.**

***Index Terms*— Active prediction, planning, and execution system, robotic interception.**

I. Introduction

A KEY FEATURE OF intelligent robotic systems is the ability to perform autonomously a multitude of tasks without complete *a priori* information, while adapting to continuous changes in the working environment. An important problem in this field is the robotic interception of moving objects. A common approach to the object-interception problem is the utilization of a prediction, planning, and execution (PPE) strategy [1], [2]. In a PPE strategy, the motion of an object through a robot's workspace is predicted. Robot motion to intercept the object is then planned and executed. This approach can be used in an "active" mode (APPE), e.g., [3], where the three stages may be repeated as necessary to ensure the successful completion of the interception task. In this context, a novel APPE system developed and implemented in our laboratory will be described in this paper.

APPE-based approaches constitute an alternative to tracking-based techniques, which essentially minimize the difference between the state of the robot's end-effector and the state of the moving object, [4], [5]. The principal advantage of APPE systems over tracking-based systems is their ability to find an optimal solution to the interception-point planning problem. However, most APPE-based techniques reported in the literature target nonindustrial settings. They normally sacrifice time optimality in favor of a guarantee of interception, either for fast-moving objects or for objects of which the Cartesian path topology and/or velocity profiles vary significantly over short periods of time. Therefore, they choose a rendezvous point among a limited number of potential interception points considered, where these usually correspond to the intersection of the (most current) estimated target trajectory with fixed planes (or lines in the planar cases).

In this paper, the proposed novel technique allows the interception of the moving object at the earliest possible time, given the constraints of the robot's motion capabilities. Namely, the interception is not restricted to a choice among a few potential points, as with other APPE systems, but targeted toward the selection of the best rendezvous point anywhere on the target's predicted trajectory. Prior to the description of our research results, some APPE systems are briefly reviewed below.

Hove and Slotine [6] used a hybrid PPE-tracking system for robotic ball catching. The PPE approach is used to plan the initial motion of the robot. A point on the ball's path (parabolic in nature) closest to the initial position of the end-effector is chosen as the first potential rendezvous point. The robot is immediately sent to this point. A tracking strategy takes over the control of the robot once the ball passes the first potential rendezvous point. Thus, in this paper, the PPE technique is utilized to improve upon the principal tracking-based method.

Andersson [3] used an APPE approach in the development of a robotic ping-pong player. Based on the predicted motion of the ball, three possible interception points are identified

Manuscript received November 1, 1996; revised Sptember 19, 1997. Recommended by Technical Editor T. J. Tarn. This work was supported by the Natural Sciences and Engineering Research Council of Canada. The work of E. A. Croft was supported by the Canadian Federation of University Women through the Margaret McWilliams Pre-doctoral Scholarship.

D. Hujic was with the Computer Integrated Manufacturing Laboratory, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ont., M5S 3G8 Canada. He is now with Celestica Inc., North York, Ont., M3C 1V7 Canada.

E. A. Croft is with the Industrial Automation Laboratory, Department of Mechanical Engineering, University of British Columbia, Vancouver, B.C., V6T 1Z4 Canada.

G. Zak, R. G. Fenton, J. K. Mills, and B. Benhabib are with the Computer Integrated Manufacturing Laboratory, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ont., M5S 3G8 Canada.

Publisher Item Identifier S 1083-4435(98)06943-9.

within the workspace of the robot. These correspond to the intersection of the ball's trajectory with three fixed "hit planes" selected *a priori* perpendicular to the ball's direction of travel. An expert system selects the most suitable rendezvous point. As new information about the ball's trajectory becomes available, the planner can modify the rendezvous point within the selected plane.

Butazzo *et al.* [7] developed an APPE system for the interception of a planar object. The motion of the object is much less predictable than in the other cases described above. The interception is always performed on the object plane along a virtual catching line. This constraint defines the rendezvous point as the intersection of the object's trajectory and the catching line. The starting position of the robot end-effector is at a fixed height above the catching line. The motion of the robot in [7] (for the vertical motion of the end-effector only), as well as in [3], is planned using quintic polynomials, where robot motion time is defined by the arrival time of the target at the rendezvous point.

Lastly, Mikesell *et al.* [8] developed an APPE system for the robotic interception of an object moving in a plane and bouncing off the walls of a square enclosure. As in [7], a rendezvous point is determined by first selecting one of the two fixed intercept lines which the object's trajectory intersects. Once an intercept line is chosen, the actual rendezvous point on it is chosen based on continuously updated object-trajectory prediction data.

II. System Overview and Problem Definition

APPE approaches to robotic interception of moving objects use both *a priori* and on-line information to create interception strategies. The cornerstone of the planning process is the selection, evaluation, and updating of a rendezvous point. This, in turn, depends on the ability of the planning module to plan robot trajectories in an on-line mode.

The APPE approach assumes that the object trajectory does not vary significantly as a function of time and, thus, it is predictable. This principal feature of "long-term" predictability of the object motion can be exploited for finding the earliest interception point, rather than solely minimizing the difference between the states of the robot end-effector and the object.

The general APPE system developed in our laboratory is depicted in Fig. 1. The role of the "prediction module" is to provide the planning and replanning modules with on-line predictions of the target's trajectory. The role of the "planning module" is to determine the first *optimal* rendezvous point on the initially predicted target trajectory and initiate the robot's end-effector motion to meet the object at this point. The role of the "replanning module" is to locally modify the location of the rendezvous point and generate new robot trajectories as needed to ensure interception. This module is provided with on-line information about the robot's current state and the latest predicted target trajectory. The "execution module" implements the robot trajectories received from the planning and replanning modules.

The primary objective of this paper is the development of a motion-planning and execution strategy for a robot to rendezvous with a moving target, where the target is defined as the "pregrasping" location for the object.[1] It is desired that the rendezvous point be the "earliest" possible interception point on the target trajectory. This optimality, however, can only be achieved by evaluating all potential rendezvous points on the target's trajectory, and selecting the one to which the robot would take the shortest time to travel.

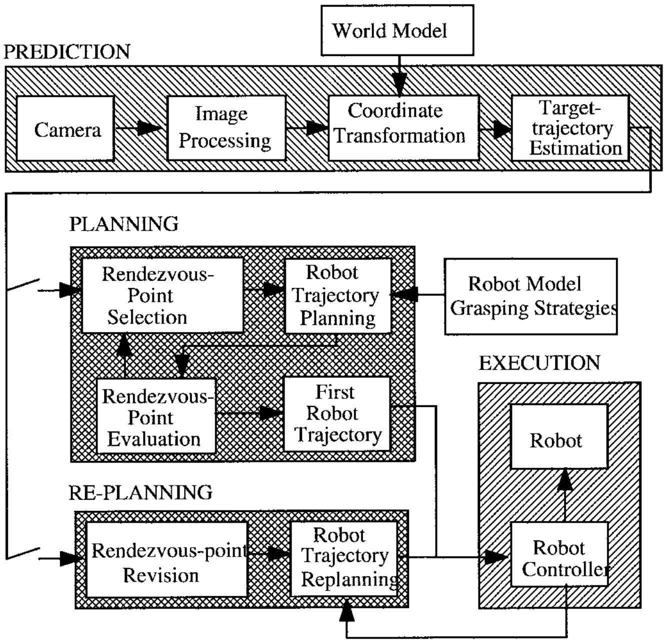

World Model
PREDICTION
Camera
Image Processing
Coordinate Transformation
Target-trajectory Estimation
PLANNING
Rendezvous-Point Selection
Robot Trajectory Planning
Robot Model Grasping Strategies
Rendezvous-Point Evaluation
First Robot Trajectory
EXECUTION
Robot
RE-PLANNING
Rendezvous-point Revision
Robot Trajectory Replanning
Robot Controller

Fig. 1. APPE system implementation.

According to the APPE system proposed in Fig. 1, the primary optimization problem at hand is, therefore, determining the "initial" optimal rendezvous point (planning module). It is anticipated that a large portion of the computational time required to intercept the object would be spent at this stage. This optimization problem can be formulated as a two-level procedure.

Outer loop: Evaluate potential rendezvous points on the target's predicted trajectory with respect to robot-motion time and choose the earliest interception point. (This substage is a one-dimensional optimization problem.)

Inner loop: Determine the minimum robot-motion time from the initial state of the robot $(\boldsymbol{q}_o, \dot{\boldsymbol{q}}_o)$ to a potential rendezvous point $(\boldsymbol{q}_f, \dot{\boldsymbol{q}}_f)$ considered by the outer loop. (As will be discussed later, for this substage, we recommend the use of any one of the on-line time-optimization techniques reported in the literature.)

The subsequent replanning stage of the rendezvous point, in response to new target-trajectory prediction data, assumes that the optimality achieved above can be maintained if the target's trajectory does not change significantly (replanning module). Although, at this stage, the interception time can be further

[1] Once the end-effector reaches the pregrasping rendezvous point, it is expected that a fine-motion tracking strategy would be utilized to grasp the object. Most likely, proximity sensors would be employed for active sensing.

reduced, the primary concern is the successful interception of the target. This second problem can be formulated as a two-step procedure.

1) Consider a new rendezvous point on the *newly predicted* target's trajectory and, if needed,
2) modify the robot's original trajectory to guarantee interception.

One must note, however, that extensive optimization cannot be carried out at this replanning stage (as was the case in the planning of the initial rendezvous point), due to severe computational time restrictions. Herein, this is acceptable, since, in industrial settings, objects' paths would not vary significantly anyway (e.g., grasping an object from a moving automatically guided vehicle).

In regard to the prediction of the target trajectory, the first optimization problem stated above requires long-term prediction, as is the case with all APPE systems, while the replanning stage requires short-term prediction, as is the case with all tracking-based systems and the replanning modules of other APPE systems. Thus, in the next section, we first present a novel long-term target-trajectory prediction algorithm (prediction module). The rendezvous-point planning issues are addressed in the subsequent section.

III. Target-Trajectory Prediction

The proposed prediction system is envisioned to have two principal objectives within the framework of the APPE; it provides the planning and replanning modules with predictions of the target's trajectory and estimates of the uncertainty associated with these predictions. Providing the replanning module with a measure of confidence is essential, so that it can decide whether a replan is required.

A. Kalman Filter

The three most common prediction approaches reported in the literature are the autoregressive models (ARM's) [9], α–β–γ filters [2], and Kalman filters (KF's) [10]. Papanikolopoulos *et al.* [11] investigated several prediction approaches and found that stochastic approaches, such as the KF, are very robust in the face of noise.

The central element of our prediction module is the KF algorithm. The KF algorithm receives its information from a computer-vision system that continuously tracks the object's position in world coordinates. The KF is a computationally efficient algorithm which generates an optimal least-squares estimate from a sequence of noisy observations [12]. For linear systems, it produces a new optimal estimate from an additional observation without having to reprocess past data. The KF can also be used to obtain multiple-step-ahead predictions [13] by propagating the KF extrapolation equation (i.e., one-step-ahead predictor).

In our system, the function of the KF is to obtain optimal estimates of the target's present position, as well as predictions of the target's future trajectory.

The KF is a state-space formulation; thus, the model describing the target must also be expressed in state-space form. For example, the two-dimensional system model for the KF is defined as [14]

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

where $\boldsymbol{x}$ is the state vector, which in our case, is the Newtonian state of the object being intercepted, and ϕ is the state transition matrix relating the object's past state to its current.

The measurement model is given as

$$\boldsymbol{z}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k, \quad \boldsymbol{v}_k \approx N(0, \boldsymbol{R}_k) \tag{2}$$

where $\boldsymbol{z}$ is the noisy measurement obtained with the camera, and $\boldsymbol{H}$ is the measurement matrix relating the observed measurement to the true state of the target.

In (1) and (2), $\boldsymbol{w}$ and $\boldsymbol{v}$ are zero-mean, mutually uncorrelated, white noise with $\boldsymbol{Q}$ and $\boldsymbol{R}$ covariances (herein, assumed constant).

Given the initial expected value of the state and its covariance as $\hat{\boldsymbol{x}}_o$ and $\boldsymbol{P}_0$,

$$E[x(0)] = \hat{x}_0, \quad E[(x(0) - \hat{x}(0))(x(0) - \hat{x}(0))^T] = P_0. \tag{3}$$

The digital KF is given by the following five equations [14]:

1) initial state-estimate extrapolation (one-step-ahead predictor)

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

2) error-covariance extrapolation

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

3) Kalman gain matrix update

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

4) optimal state-estimate extrapolation

$$\hat{\boldsymbol{x}}_k(+) = \hat{\boldsymbol{x}}_k(-) + \boldsymbol{K}_k[z_K - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k(-)] \tag{7}$$

5) covariance update

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

In modeling a two-dimensional system, the noise in the x and y directions are assumed to be independent and, thus, the x and y states are decoupled. A constant-velocity target motion model is used for each direction, $\ddot{x}(t) = 0$. In practice, however, the velocity does undergo at least slight changes. This can be modeled by a continuous-time white noise $\tilde{v}(t)$ [15]:

$$\ddot{x}(t) = \tilde{v}(t) \tag{9}$$

where

$$\boldsymbol{E}[\tilde{v}(t)] = 0 \quad \boldsymbol{E}[\tilde{v}(t)\tilde{v}(\tau)] = \sigma^2\delta(t - \tau).$$

Herein, the use of a fading memory filter (FMF) [14] is also proposed to place more emphasis on the newest data. With an FMF, old data is discarded by increasing the covariance of the measurement noise for past measurements:

$$\boldsymbol{R}_k^* = s^{j-k}\boldsymbol{R}_k, \qquad k = j, j-1, j-2, \cdots \quad s \geq 1, \quad j \geq 1 \tag{10}$$

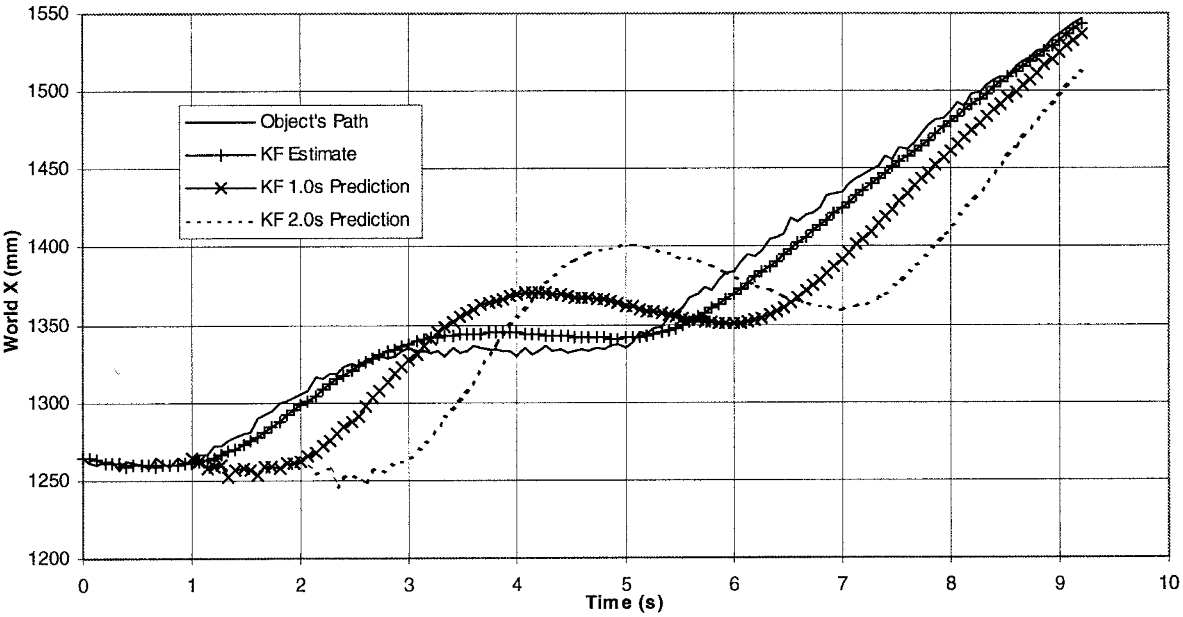


Fig. 2. Best overall constant-velocity KF for stop-and-go motion.

where $\boldsymbol{R}_k$ is the regular noise covariance, $\boldsymbol{R}_k^*$ is the new noise covariance, and

$$s = e^{\Delta t/\tau}. \tag{11}$$

In (11) above, Δt is the measurement interval and τ is the age-weighting time constant. It then follows that

$$\boldsymbol{R}_{k-m}^* = (e^{m\Delta t/\tau})\boldsymbol{R}, \qquad m = 0, 1, 2, \cdots. \tag{12}$$

A recursive KF can be constructed under these assumptions. It is identical to the standard KF given by (4)–(8), except for a modified error-covariance-extrapolation equation (i.e., the plant's covariance term is now multiplied by the factor s)

$$\boldsymbol{P}_k'(-) = s\boldsymbol{\Phi}_{k-1}\boldsymbol{P}_{k-1}'(+)\boldsymbol{\Phi}_{k-1}^T + \boldsymbol{Q}_{k-1}. \tag{13}$$

A KF must be "tuned," via the adjustment of all the variable parameters, by analyzing the system's performance (in the form of the residual series). The parameters to be tuned are problem specific but usually include the decorrelation time and the standard deviation of the white noise driving the system covariance.

Most KF tuning criteria simply compare the target's observed behavior to the estimates of the KF. A well-tuned KF is one that tracks the target well. However, through simulations, one would note that a KF well tuned for one-step-ahead tracking will usually not provide good long-term predictions. Thus, using a similar line of reasoning, a long-term predictive KF must be selected by comparing the KF's predictive and tracking performance to the true behavior of the target.

B. Simulations

Besides the constant-velocity model described above, constant-acceleration, constant-jerk, and first-order Gauss Markov filters were also examined for various planar object paths. Through simulations, improvements in long-term-prediction performance were shown to be obtainable by decreasing the FMF factor s. However, by decreasing the FMF factor, the KF places less emphasis on newer data. This makes the filter less responsive to sudden changes in the object's path. Thus, in selecting a KF for the proposed APPE, a compromise had to be achieved. A KF that provides reasonable one-step-ahead tracking and good, stable long-term predictions was selected.

In one exemplary "worst case scenario" simulation, addressed herein, the object accelerated uniformly at 50 mm/s^2 to a maximum velocity of 125 mm/s. After traveling at this velocity for 0.3 s, it decelerated to rest, remained stationary for 2.7 s, and then accelerated to its maximum velocity, at which it traveled until it left the camera's field of view (an unlikely motion in industrial environments). Superimposed onto the object's trajectory was Gaussian white noise with a 3-mm standard deviation, approximately twice the magnitude of the noise present in our current system. The object's trajectory was sampled at standard video rate, 15 Hz.

Fig. 2 shows the simulation results based on a constant-velocity target motion model which has a 1.11-s time constant. Plotted on each curve, besides the object's path and the one-step (i.e., 0.067 s) KF estimate of its current position, are predictions of the object's future position, 1.0 and 2.0 s into the future.

IV. Robot-Motion Planning

Both the optimal planning of the initial rendezvous-point stage and the subsequent replanning of the modified points stage will be addressed in this section.

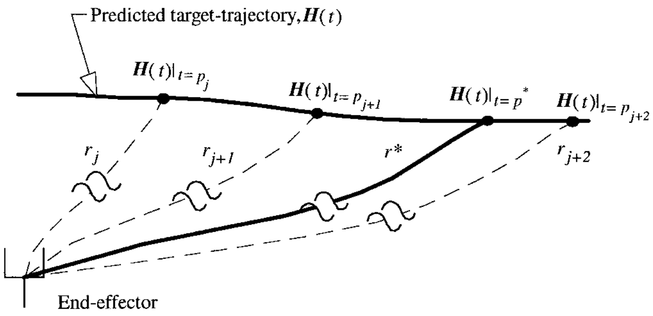

Predicted target-trajectory, $\boldsymbol{H}(t)$
$\boldsymbol{H}(t)|_{t=p_j}$
$\boldsymbol{H}(t)|_{t=p_{j+1}}$
$\boldsymbol{H}(t)|_{t=p^*}$
$\boldsymbol{H}(t)|_{t=p_{j+2}}$
r_j
r_{j+1}
r^*
r_{j+2}
End-effector

Fig. 3. Predicted target trajectory and planned time-optimal robot trajectories.

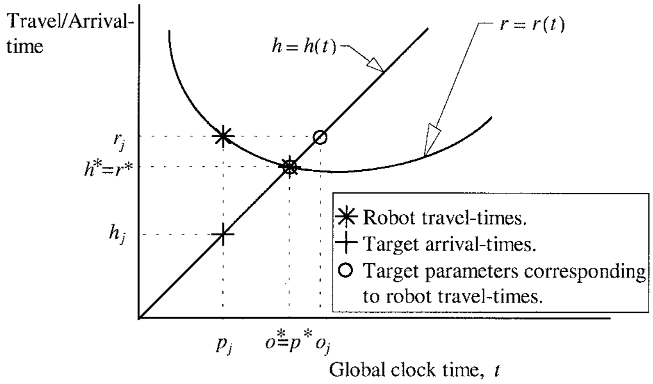


Fig. 4. Travel/arrival-time diagram.

A. Search for the Initial Rendezvous Point

1) Problem Formulation: The determination of the optimal (initial) rendezvous point involves a one-dimensional search along the predicted target trajectory, defined herein as $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}$, for the earliest interception point. The predicted target trajectory is parameterized by a global clock time t. A rendezvous point is characterized by both the potential *rendezvous time* p_j and the corresponding *rendezvous state* $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t-p_j}$.[2]

Potential rendezvous points on $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}$ can be evaluated using any on-line point-to-point (PTP) optimal robot-trajectory planning technique (see Section V). Such a calculation would provide the minimum robot travel time r_j to $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=p_j}$ (Fig. 3), as well as the robot trajectory to that state.[3]

2) Proposed Solution: Most solutions to the global time-optimal PTP trajectory-planning problem (with two fixed boundary points) reported in the literature are computationally intensive and, thus, not suitable for on-line planning. In our case, since only the initial boundary point $(\boldsymbol{q_o}, \dot{\boldsymbol{q}}_{\boldsymbol{o}})$ is specified, the solution of the time-optimal rendezvous problem further requires the additional search of the target trajectory for an end point $(\boldsymbol{q_f}, \dot{\boldsymbol{q}}_{\boldsymbol{f}})$ consistent with the minimization process. Thus, for on-line planning, a computationally inexpensive, near-optimal solution is necessary.

Once a PTP-motion time-optimal trajectory-planning technique is selected, the problem that remains to be addressed is finding the optimal rendezvous point on the predicted target trajectory. The former issue is addressed in Section V, while, herein, we assume that such a solution algorithm is available for finding the minimum robot-travel time r_j to a specific potential rendezvous state considered, within our one-dimensional optimization search.

A potential rendezvous point is characterized by the arrival times of both the robot and the target at the corresponding rendezvous state. To illustrate this evaluation, a *travel-time diagram* is constructed (Fig. 4). In this diagram, the global clock time t is plotted along the abscissa. Potential rendezvous times are represented on the abscissa by p_j, p_{j+1}, etc. Robot travel times r_j and target arrival times h_j to potential rendezvous states are plotted on the ordinate.

The travel-time diagram shown in Fig. 4 is constructed as follows.

1) Select a predicted state on the target trajectory $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=p_j}$ (Fig. 3) as a potential rendezvous state. The associated rendezvous time p_j is marked on the abscissa of the travel/arrival-time diagram (Fig. 4).
2) The target arrives at state $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=p_j}$ at time h_j. This is marked by the "+" symbol on the travel/arrival-time diagram.
3) The robot, on the other hand, takes time r_j to arrive at $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}\}|_{t=p_j}$, as marked by the "$*$" symbol on the travel/arrival-time diagram. (It should be noted that only $r_j = h_j$ indicates the minimum-time rendezvous point).
4) The actual target state at the time the robot arrives at state $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=p_j}$ is designated by $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=oj}$ where $o_j = r_j$. (This assumes that $r_j \neq h_j$; namely, the proposed rendezvous point is not the minimum-time point).

To reiterate, *clock times* p_j and o_j, associated with different target states, are plotted on the abscissa. On the other hand, *travel/ arrival times* h_j and r_j, associated with the *time* that the target and robot take to reach a particular state, are plotted on the ordinate.

Since the target trajectory is parameterized by t, the target arrival-time function $h = h(t)$ is simply a straight line with slope 1. A function describing the robot travel times to points on the target trajectory is defined herein as $r = r(t)$. The intersection of $r(t)$ and $h(t)$ designates the optimal rendezvoustime p^*, namely, the earliest time that the robot end-effector can reach the same location with the velocity as the target. As expected, except for the intersection of $r(t)$ and $h(t)$ at time p^*, a difference will exist between r_j and h_j.

Using the above definitions, the time-optimal rendezvous-point planning problem can be stated as follows.

"Find the earliest feasible rendezvous point $(\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=p_j}, p_j)$, such that $r_j = h_j$."

[2] Each rendezvous-point is defined as a $2n + 1$-dimensional vector (where n is the number of degrees of freedom of the target's mobility).

[3] Thus, the robot trajectory to each potential rendezvous state, including the optimal rendezvous state, is found as a byproduct of the search for the optimal rendezvous-point.

Although the use of a specific robot-trajectory planning method would vary the value of the achievable optimal motion time r^*, this could be simply seen as obtaining a different $r(t)$ curve, which would naturally intersect $h(t)$ in Fig. 4 at a different location. Thus, the success of the interception strategy proposed in this paper is independent of the specific robot-trajectory planning technique selected.

As the principal justification for the formulation presented in this paper, it must be noted that the robot travel time $r(t)$ cannot be analytically obtained, nor can it be exactly evaluated via numerical methods in an on-line manner. Thus, herein, as part of the general APPE strategy, a numerical approximation of $r(t)$ based only on the calculations of optimal robot motion times to few potential rendezvous points is proposed. Since an *approximation* of $r(t)$ is used to find the intersection of $r(t)$ and $h(t)$, the true value of p^* will never be exactly determined. In other words, the value of the approximated $r(t)$ curve at the intersection point r^* will always be different than the true (minimum) robot-travel time to the rendezvous point denoted by p^*. Thus, in order to determine the "achievable" optimal rendezvous point, an iterative search procedure has been devised. In this algorithm, the robot-travel time function $r(t)$ is iteratively improved based on the new calculations of the true robot-motion times to potential rendezvous states found in earlier iterations. The central element of this algorithm is an objective function, defined below, which is calculated at every iteration to evaluate the potential rendezvous point at hand.

a) The objective function: interception time: As stated above, examination of Fig. 4 indicates that potential rendezvous points for which $p_j < p^*$ would require the robot to "chase" the target, since the robot cannot reach these points prior to the target's arrival. If the robot could catch up to the target before it moves out of the robot's workspace, it would catch the target at an interception time greater than r_j. Points for which $p_j > p^*$, on the other hand, would require the robot to wait for the target, with an interception time also greater than r_j. Thus, the objective-function value of a potential rendezvous point $(\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=p_j}, p_j)$ can be defined as the corresponding estimated "real" interception time y_j at which the robot would intercept the target [16]

$$y_j \approx \begin{cases} r_j + \dfrac{r_j - h_j}{V_r/V_t^{-1}}, & r_j > h_j \\ h_j, & r_j \le h_j \end{cases} \tag{14}$$

where V_r is an average (*a priori* known) robot end-effector speed, and V_t is an average (on-line calculated) target speed. One can note that the calculation of the estimated interception time is simply a measure of the value of a particular planned rendezvous point and does not necessarily reflect a completed plan to intercept the target at this interception time. The actual interception time may be different due to replanning, as will be described in Section IV-B. For the set of rendezvous points $\{p_j\}$ considered so far, the current best achievable interception time $\hat{y}$ is then defined as

$$\hat{y} = \min\{y_j\}. \tag{15}$$

b) The temporal convergence criterion: Given the above formulation for the objective function, the solution to the rendezvous-point problem is achieved by iteratively increasing the accuracy of $r(t)$, in order to improve upon $\hat{y}$. The convergence criterion for the minimization of $\hat{y}$ can be determined by considering the cost of each minimization step. The time cost of improvement, namely, the period of time required to compute a new robot trajectory to the proposed rendezvous point t_p should not exceed the expected reduction in interception time due to an additional iteration.

The decision-making procedure for the convergence of the minimization is as follows.

1) Consider a new potential rendezvous time p_j, determined as the intersection of (the current approximation of) $r(t)$ and $h(t)$.
2) Consider h_j (the target arrival time), as the minimum possible interception time for p_j (14).
3) Determine the *potential* improvement in the interception time Δt_j with respect to the previous iteration

$$\Delta t_j = (\hat{y} - h_j). \tag{16}$$

4) If the time t_p that it takes to calculate the near-time-optimal PTP robot motion to $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=p_j}$ is greater than Δt_j, namely,

$$\Delta t_j < t_p \tag{17}$$

then, the optimization algorithm has converged.

Otherwise, $r(t)$ is reapproximated using the additional true robot-motion time r_j calculated for $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=p_j}$, and steps 1)–4) are repeated.

The above convergence criterion, (17), is "temporal" in nature and is only concerned with the iterative improvement estimation of $r(t)$.

c) Coping with uncertainties: Uncertainties in the estimation of the target's trajectory $h(t)$ complicate the search for the intersection of $r(t)$ and $h(t)$ at every iteration. Thus, for a certain $r(t)$ approximation at hand, the problem introduced by the uncertainty in the target position is the determination as to whether further effort is required to improve the spatial position of the end-effector with respect to the predicted target location at the rendezvous time (namely, deciding when to stop the search algorithm for the exact intersection of $h(t)$ and $r(t)$).

Given an estimate of the variance of a predicted target location $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=o_j}$ (provided by the prediction module), an uncertainty ellipse [17] and a corresponding tolerance region can be constructed around the predicted point. The front and rear limits of the tolerance region can be represented by two limiting points on the travel-time diagram, namely, o_j^- and o_j^+ (Fig. 5). The set of all such limiting points forms the tolerance-limit lines about the target arrival-time line.

The tolerance limits $T(o_j)$ provide the positional convergence constraint corresponding to a potential rendezvous time p_j, represented herein as

$$p_j \in T(o_j). \tag{18}$$

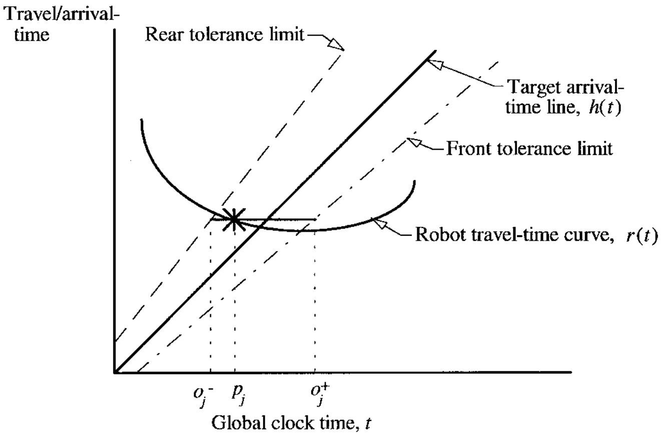


Fig. 5. Representation of the tolerance limits around a point on the predicted target trajectory.

Equation (18) is used as a necessary condition for the convergence of the rendezvous-point selection algorithm.

3) The Rendezvous-Point Planning Algorithm: The preceding section provided a basis for the formulation of the one-dimensional rendezvous-point planning optimization problem, which is *minimize the interception time y subject to the positional convergence criterion:*

$$z = \min\{y(p_j)\} \quad \text{where} \quad p_j \in \boldsymbol{T}(o_j). \tag{19}$$

The comprehensive planning algorithm to solve (19) is as follows.

1) Start the approximation of $r(t)$ by selecting two extreme states $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=p_1}, \{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=p_2}$ on the predicted target trajectory (for example, the entry and exit intersections of the predicted object trajectory with the workspace envelope of the robot). Set $j = 2$.
2) Calculate the robot motion times r_1 and r_2 to the two rendezvous states using a PTP near-time-optimal trajectory planning algorithm, record the values of h_1 and h_2, and set $\hat{y} = 0$.
3) Construct $r(t)$ using $\{r_j\}$.
4) Set $j = j + 1$.
5) Estimate the intersection of $r(t)$ and $h(t)$,[4] p_j, such that $p_j \in T(o_j)$.
6) Determine h_j and, if $\hat{y} \neq 0$, check whether $\hat{y} - h_j < t_p$; if true, set $p^* = \hat{p}$ as the optimal rendezvous time and stop, otherwise, continue. (Note that $\hat{p}$ is defined only if $\hat{y} \neq 0$, step 8.)
7) Calculate near-time-optimal motion to $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=p_j}$, with motion time r_j.
8) Determine y_j, set $\hat{y} = min\{y_j\}$, and record $\hat{p}$ corresponding to $\hat{y}$.
9) Go to step 3).

The above algorithm was proven to monotonically converge to the optimal solution $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=\hat{p}}$ in [16].

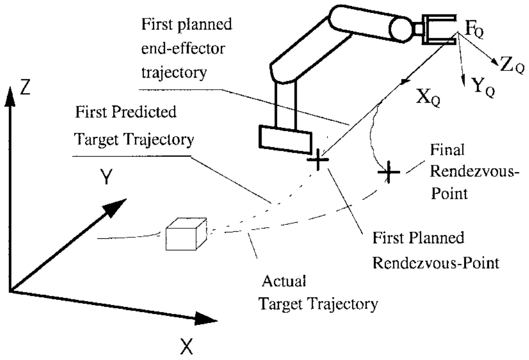

First planned end-effector trajectory
Z
First Predicted Target Trajectory
Y
F_Q
Z_Q
X_Q
Y_Q
Final Rendezvous-Point
First Planned Rendezvous-Point
Actual Target Trajectory
X

Fig. 6. The schematic view of the robotic interception system.

B. Replanning

If the rendezvous-point planner described in Section IV-A was provided with accurate information, then the robot could intercept the target at the first planned optimal rendezvous point. However, uncertainties in the model of the target motion result in inaccurate predictions of the target trajectory. In this context, the primary purpose of replanning is to locally modify the initial optimal rendezvous point, so that interception is guaranteed, and not to further improve the interception time. A new strategy is required for this task, since the repeated application of an *optimal* rendezvous-point selection algorithm for determining improved rendezvous points would not be feasible due to computation-time constraints.

In the proposed strategy, replanning entails altering the unexecuted portion of the robot trajectory by planning a trajectory patch to a newly selected rendezvous point (Fig. 6). Motion continuity is maintained at the patch starting point. Replanning includes two successive examinations:

1) determining whether a new rendezvous point, chosen on the newly predicted target trajectory, would potentially improve upon the robot-motion plan currently under execution;
2) determining whether the target can be intercepted at this new rendezvous point and, if not, how a modified rendezvous point can be chosen to guarantee interception.

a) Determining the potential for improvement: Replanning must be motivated by a potential improvement upon the currently aimed rendezvous point, considering the newly predicted target trajectory $\{\boldsymbol{H}'(t), \dot{\boldsymbol{H}}'(t)\}$. The currently aimed rendezvous state $\{\boldsymbol{H}(t), \dot{\boldsymbol{H}}(t)\}|_{t=p_j}$ must first be compared to a new potential rendezvous state on the newly predicted target trajectory [Fig. 7(a) and (b)].

The comparison procedure is as follows.

1) Find a state on the newly-predicted target trajectory closest in Cartesian distance to $\boldsymbol{H}(t)|_{t=\hat{p}}$. Denote this state as $\{\boldsymbol{H}'(t), \dot{\boldsymbol{H}}(t)\}|_{t=p'}$.
2) Determine whether $\boldsymbol{H}(t)|_{t=\hat{p}}$ is within the uncertainty ellipse surrounding the new rendezvous state

$$\boldsymbol{H}(t)|_{t=\hat{p}} \in U(\boldsymbol{H}'(t)|_{t=p'}). \tag{20}$$

[4] The travel-time line is shifted to the right an amount t_{calc} to accommodate the time required for an estimated maximum number of planning iterations. However, instead of fixing a bulk cost to planning, the time cost of planning could be incorporated into each calculation of the robot travel time and interception time. Such an adaptive approach is presented in [16].

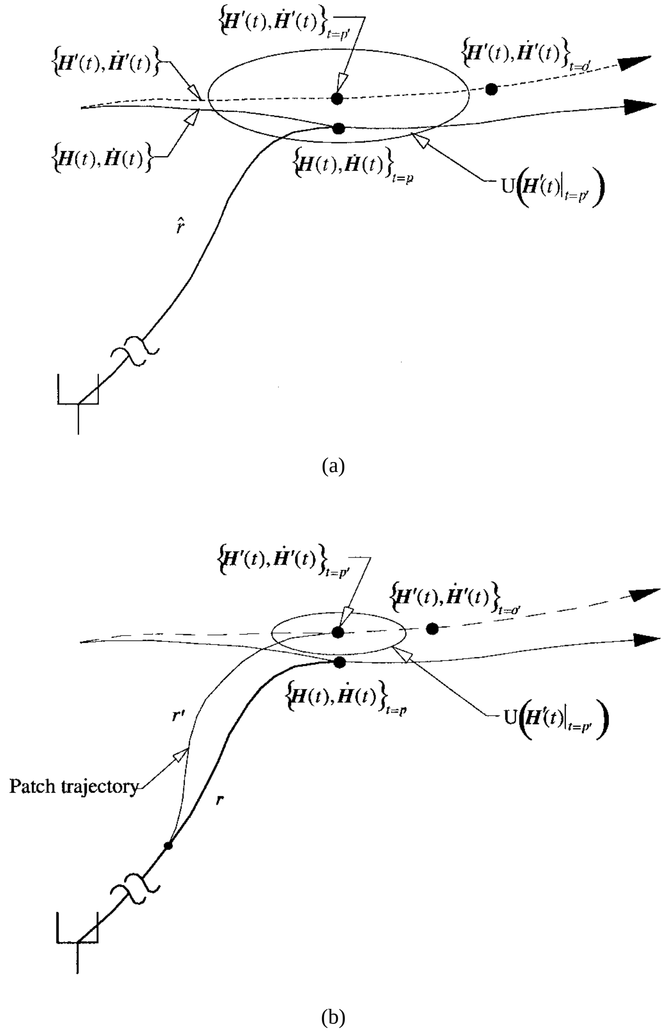

$\{\boldsymbol{H}'(t),\dot{\boldsymbol{H}}'(t)\}_{t=p'}$
$\{\boldsymbol{H}'(t),\dot{\boldsymbol{H}}'(t)\}$
$\{\boldsymbol{H}'(t),\dot{\boldsymbol{H}}'(t)\}_{t=o'}$
$\{\boldsymbol{H}(t),\dot{\boldsymbol{H}}(t)\}$
$\{\boldsymbol{H}(t),\dot{\boldsymbol{H}}(t)\}_{t=p}$
$\mathrm{U}(\boldsymbol{H}'(t)|_{t=p'})$
$\hat{r}$
(a)
$\{\boldsymbol{H}'(t),\dot{\boldsymbol{H}}'(t)\}_{t=p'}$
$\{\boldsymbol{H}'(t),\dot{\boldsymbol{H}}'(t)\}_{t=o'}$
$\{\boldsymbol{H}(t),\dot{\boldsymbol{H}}(t)\}_{t=p}$
$\mathrm{U}(\boldsymbol{H}'(t)|_{t=p'})$
r'
Patch trajectory
r
(b)

Fig. 7. Spatial evaluation of the location of the currently aimed rendezvous state.

If it is, then, the current robot motion under execution would take the end-effector to a spatial location which is sufficiently close to $\{\boldsymbol{H}'(t), \dot{\boldsymbol{H}}(t)\}|_{t=p'}$ [Fig. 7(a)]. However, the algorithm must now proceed to step 3) to check whether the spatial location of the end-effector lies within the tolerance limits surrounding the corresponding new predicted target location, which corresponds to $o' = \hat{r}$.

Otherwise, plan a robot-trajectory patch to $\{\boldsymbol{H}'(t), \dot{\boldsymbol{H}}(t)\}|_{t=p'}$ and $o' = r'$ [Fig. 7(b)].[5]

3) Determine whether the newly considered rendezvous state lies within its tolerance limits (Fig. 8)

$$p' \in T'(o'). \tag{21}$$

If it does, further replanning is not necessary:

a) robot is sent to the original rendezvous state if no replanning was carried out, or

b) robot is sent to the revised rendezvous state if a new robot patch was determined, and the cycle of prediction and comparison is repeated by returning to step 1).

Otherwise, replanning is necessary to ensure interception, as will be discussed below.

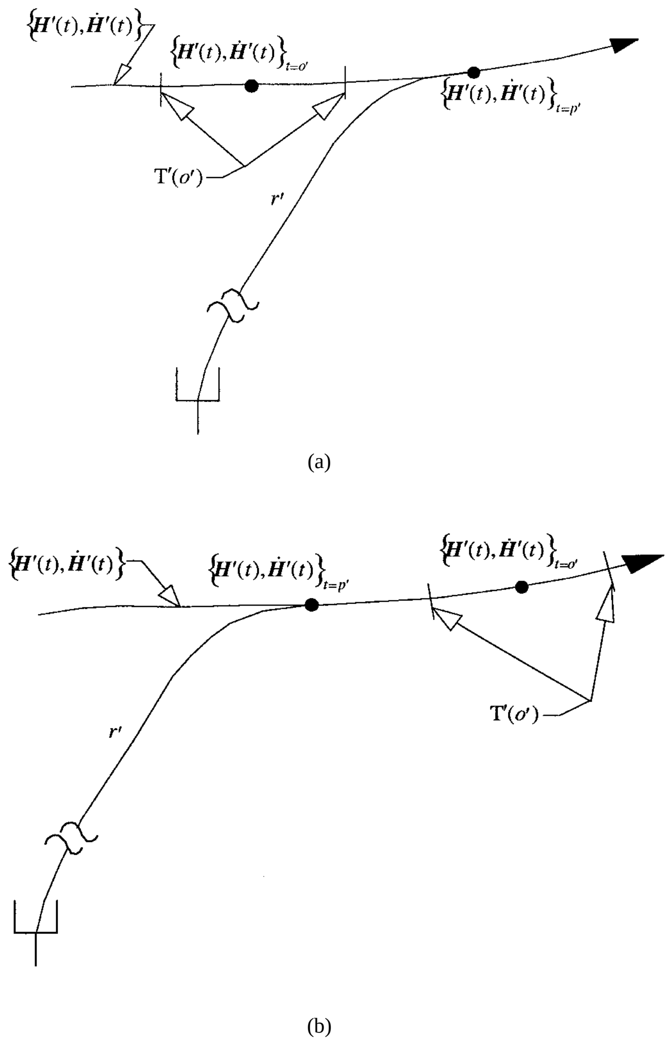

$\{\boldsymbol{H}'(t),\dot{\boldsymbol{H}}'(t)\}$
$\{\boldsymbol{H}'(t),\dot{\boldsymbol{H}}'(t)\}_{t=o'}$
$\{\boldsymbol{H}'(t),\dot{\boldsymbol{H}}'(t)\}_{t=p'}$
$\mathrm{T}'(o')$
r'
(a)
$\{\boldsymbol{H}'(t),\dot{\boldsymbol{H}}'(t)\}$
$\{\boldsymbol{H}'(t),\dot{\boldsymbol{H}}'(t)\}_{t=p'}$
$\{\boldsymbol{H}'(t),\dot{\boldsymbol{H}}'(t)\}_{t=o'}$
$\mathrm{T}'(o')$
r'
(b)

Fig. 8. (a) Interception is guaranteed if the revised rendezvous point is ahead of the front tolerance limit. (b) Interception is not guaranteed if the revised rendezvous point is behind the rear tolerance limit.

b) Ensuring Interception: Once it has been determined that further replanning is necessary, namely, the tolerance-limit constraint is not satisfied, the problem of ensuring interception must be considered. To achieve this objective, a conservative strategy is proposed herein. The strategy is based on the hypothesis that it is preferable for the robot to be early, rather than late. Two cases are considered.

1) If for the revised rendezvous point, $r' < h'$, i.e., the robot will arrive at the rendezvous state before the target does, then, this point ensures interception [Fig. 8(a)].

2) Otherwise, $r' > h'$ and the robot must catch up to the target. In this case, it must also be noted however that $\boldsymbol{H}'(t)|_{t=p'}$ is behind the *rear* tolerance limit [Fig. 8(b)], since (21) is not satisfied [see step 3)]. Thus, in order to ensure interception, a newer rendezvous point must be selected. Herein, this point is selected as $p'' = y'$ (14).

[5] Robot-trajectory patch planning will be discussed in Section V.

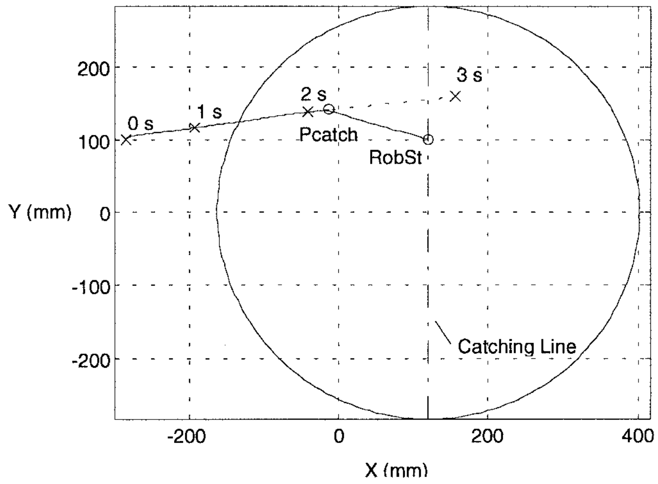


Fig. 9. Simulation of the object-catching process with the new planning strategy (top view).

V. Robot-Trajectory Planning

The optimal rendezvous-point planning problem formulated and solved in the previous section relies on our ability to: 1) carry out long-term target-trajectory predictions and 2) to plan near-time-optimal robot trajectories to potential interception points. The former issue was addressed in Section III, the latter subproblem is addressed below.

There exists a large body of literature in the field of time-optimal robot-trajectory planning (e.g., [18]–[21]). While these and various other approaches not discussed herein usually consider robot dynamics and yield optimal paths, the computation times required for finding optimal PTP solutions are prohibitive for current use in on-line mode.

Task-space quintic polynomials have been recently used for robot trajectories in real-time applications. Thus, they were adopted for our APPE system, as well. One must note, however, that the "optimal-solution" feature of the proposed APPE (versus the nonoptimality of tracking-based methods and of other APPE systems reported in the literature) is based on its ability to select the earliest rendezvous point on the target's predicted trajectory, denoted by $\hat{p} = r^*$ herein. The use of a different robot-trajectory planning method would simply result in the determination and use of a different $r(t)$ curve in Fig. 4. Namely, the success of the APPE system proposed in this paper is independent of the specific technique selected due to its modularity (Fig. 1), although the location of the optimal rendezvous point would vary from one robot-trajectory planning technique to another,

Quintic polynomials are used herein for trajectory planning, since they permit a relatively simple calculation of a trajectory based on the specification of initial and final position, velocity, and acceleration constraints. The use of this trajectory-planning method also allowed us to compare our methods with other approaches which use quintic polynomials, as well [3], [7]. The general form of the polynomial is

$$q(t) = q_0 + q_1 t + q_2 t^2 + q_3 t^3 + q_4 t^4 + q_5 t^5. \qquad (22)$$

Since there are seven independent variables (six coefficients and time), the variables are uniquely determined when the motion time plus the initial and final states are specified [3]. In our case, however, the motion time of the robot is considered as the optimization variable that must be determined.

The quintic polynomial is inherently one dimensional, therefore, in general, three separate quintic trajectories would need to be planned, one for each of the three Cartesian coordinate axes. However, in the case of motion planning for object interception, we can make the assumption that most of the robot travel will occur in the direction of the first-planned rendezvous point. Thus, it is more convenient to perform the quintic-polynomial-based trajectory planning not in the world coordinates, but in a specially defined coordinate system. This system is defined so that one of the principal axes is lined up with the vector from the robot starting point to the first-planned rendezvous point. When the robot trajectory has to deviate from this path, another two quintic polynomials need to be used, one for each of the orthogonal principal axes. If the initial motion direction is taken as X_Q (Fig. 6), the two secondary directions (along Y_Q and Z_Q) are then defined so that they form a right-handed orthogonal coordinate system F_Q.

To find the time-optimal quintic motion trajectory, the motion time is minimized, subject to the robot's performance capability constraints. For simplification purposes, the robot's performance is described herein by the worst case velocity and acceleration limits. Thus, a minimum travel time will be achieved when either velocity or acceleration limits are reached at some point during the trajectory execution. It is also assumed that most of the travel will occur along the X_Q direction, so that a near-optimal solution will be achieved when the travel time is minimized for the quintic trajectory along that direction.

This problem can be formulated as follows.

Given the initial and final conditions in terms of the required robot position, velocity, and acceleration in X_Q direction, and given the maximum velocity and acceleration limits of the robot, V_{lim} and A_{lim}, find the minimum travel time, such that

$$t_{\min} = \max(t_{\text{vel}}, t_{\text{acc}}) \qquad (23)$$

where

$$t_{\text{vel}} = f_v(V_{\text{lim}}), \quad \text{and} \quad t_{\text{acc}} = f_a(A_{\text{lim}}) \qquad (24)$$

and $f_v(\cdot)$ and $f_a(\cdot)$ are functions which return such motion time that the given velocity or acceleration limit is achieved during the execution of the trajectory.

The problem of finding t_{vel} and t_{acc} requires a numerical solution when either of the initial (due to patch planning) or final (due to rendezvous with a moving object) conditions are nonstationary (nonzero velocity or acceleration). The algorithm which accomplishes this task was implemented by using a standard iterative numerical search procedure.

VI. System Implementation

A. System Simulations

Prior to implementing the motion-planning strategies described above, computer simulations were conducted for verifying the operation of the proposed APPE system. As an illustrative case, the two-dimensional object-catching problem was formulated. To provide the input data for the example considered herein, the object's motion and the robot's starting-point positions were arbitrarily set to those provided in [7]. Therein, it is assumed *a priori* that the object's interception will occur on a predefined "catching line." Thus, the robot's end-effector was initially positioned 160 mm above the object's motion plane, directly over the catching line.

The simulated experiment reported herein was performed with the planning strategies presented in Sections IV and V. The robot is allowed to intercept the object at any point along the object's trajectory. In Fig. 9, "RobSt" marks the robot end-effector's starting point, and "Pcatch" marks the point of interception. Numbers above the target trajectory mark seconds of target travel time, and the large circle indicates the robot's workspace.

The Y velocity component of the object is relatively small, reaching a constant velocity of 21 mm/s after initial acceleration from rest. The robot begins to move as soon as the object enters the robot's workspace. The robot's acceleration limit was set to 2 m/s^2 and the velocity limit at 1 m/s. The object's motion was simulated by specifying an acceleration profile. Target-trajectory predictions were made using the values of the object's position, velocity, and acceleration at a particular point in time and extrapolating these based on the constant-acceleration motion model. The object's simulated motion data was sampled at 0.025-s intervals. The trajectory replanning time was set to 0.05 s.

As can be seen from Fig. 9, the robot is capable of intercepting the target before it reaches the catching line (located at $X = 120$ mm). This, and other simulations not shown here, have demonstrated the capability of our strategy for successful time-optimal object interception without the restrictions imposed by constraining the potential rendezvous-point positions.

B. Experimental APPE Prototype Configuration

Our present experimental system consists of a 6-degree-of-freedom GMFanuc S-100 industrial robot with the standard Fanuc Controller, an NC X–Y table (used for object motion), and a computer-vision system. In order to improve overall system performance, computations were performed on two separate 80 486 PC's, one devoted to tracking the object and the other to planning the interception and communicating with the robot controller. A 38 400-baud RS232C serial link was used to transfer data between the two PC's.

1) Tracking System: The tracking system consisted of a Hitachi 30 Hz CCD camera, a Matrox 640B frame grabber, and an 80 486 33-MHz PC. Since our emphasis is on visual servoing, and not on computer vision, simplifications to the object tracking problem were made. Objects being tracked were marked using planar red circular markers. An active red-color filter, developed in our laboratory, thresholds the analog camera signals, such that only one feature, the circle's centroid, is tracked [22]. The CCD camera with a 25-mm lens was placed 1.8 m above the surface of the X–Y table at an angle of ~10° from the table's normal. This setup yields a 600 × 400 mm field of view with ~0.93 mm/pixel resolution. The robot camera system was calibrated using the monoview noncoplanar point technique proposed in [23]. The error in x and y directions was less than 0.5%. At present, the entire process (i.e., grabbing an image, finding the object's centroid, and updating the KF) takes ~65 ms. Once updated, the KF, along with the global time at which the images were obtained, is stored in a buffer, and the motion-planning system is notified that new object data is available. At the planner's request, this data is sent to the second PC. The PC-to-PC serial communication is interrupt driven; this allows the tracker to continuously track the object without waiting for the other PC to read the data.

2) Motion-Interception System: An 80 486DX4 100 MHz PC was used to plan robot motion. This PC obtains the KF state parameters, along with the global time at which the images were obtained from the tracking system. Multiple-step-ahead prediction is used to predict the object's future trajectory, which, in turn, is used to determine optimal robot–object rendezvous points. The planner then generates a quintic trajectory, such that the robot's end-effector (a semispherical cage) will land on the object. The PC communication with the Fanuc robot controller is achieved via an RS232C serial link at 9600 baud rate, the fastest rate that the robot controller will allow. The robot–PC serial communication is interrupt driven, with the PC sending robot trajectory points at the robot's request (~220-ms increments). As more data becomes available from the tracker, the robot trajectory is replanned, as necessary, to ensure a successful catch.

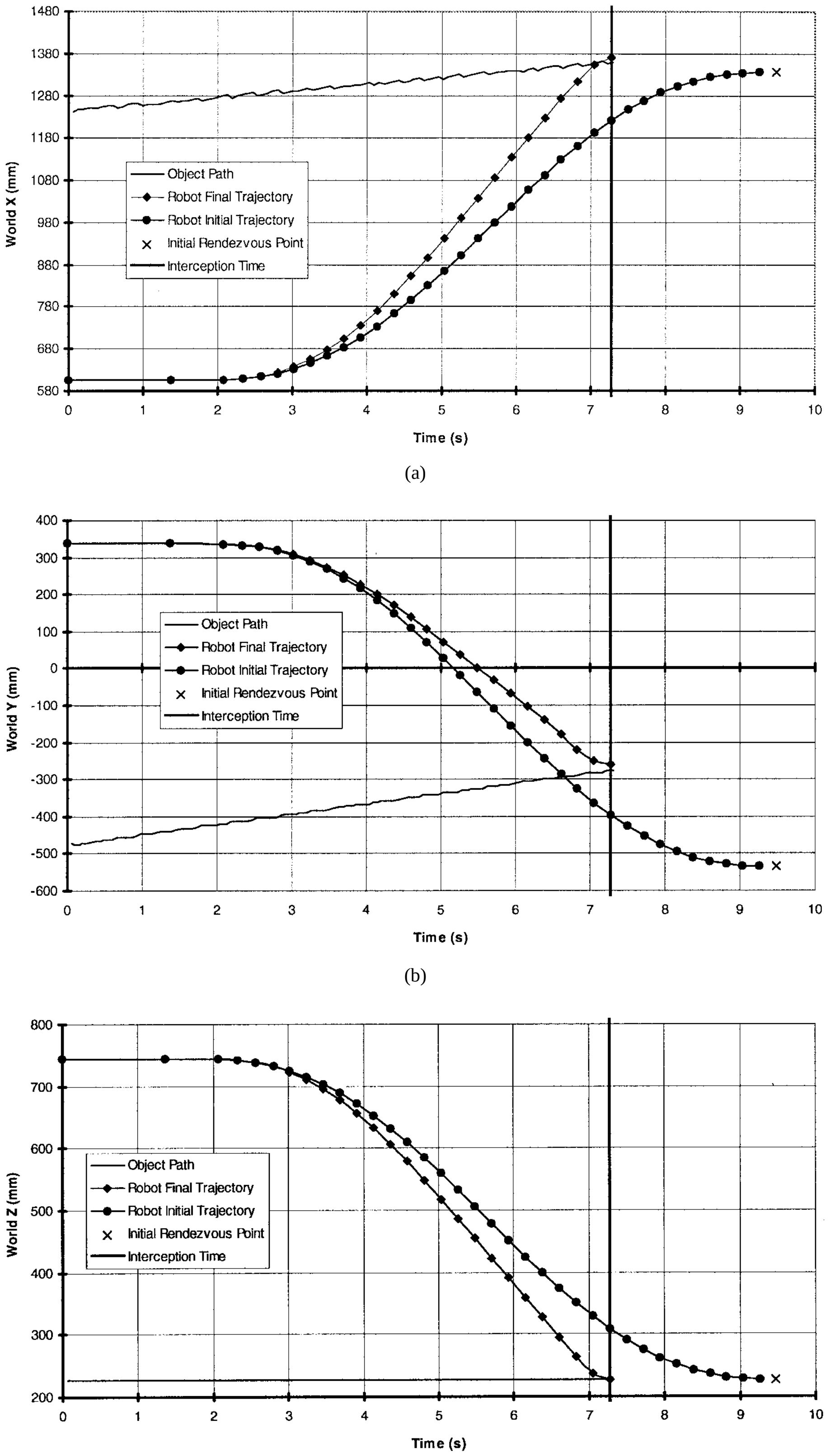
C. Experimental Results and Discussion

The current experimental system is able to reliably intercept objects, with an average error of less than 10 mm in each direction, traveling on random trajectories with velocities of up to 45 mm/s and accelerations greater than 200 mm/s^2. However, it should be remembered that the proposed APPE was designed to bring the robot end-effector to a pregrasping location near the object. Thus, errors of 10-mm magnitude constitute very successful results.

Fig. 10 shows the results of an experimental trial in which an object is moving at a constant velocity of 32 mm/s on a linear trajectory, much like a part moving on a typical industrial conveyor. Fig. 10(a)–(c) shows each coordinate direction, $X, Y,$ and Z, versus time. In each figure, the object's observed actual trajectory, along with the robot's initial trajectory (i.e., the trajectory that it would have been implemented without replanning) and final trajectory are plotted. The final trajectory implemented by the robot comprised 39 replanned patches.

Fig. 11 shows experimental results for an object moving on a circular path of radius $R = 280$ mm at 30 mm/s. The final trajectory implemented by the robot comprised 44 replanned patches.

Fig. 12 shows results for an object which is initially at rest and then subsequently accelerates to a velocity of 45 mm/s after the robot has already started moving toward it, thus forcing the robot to alter its trajectory significantly.

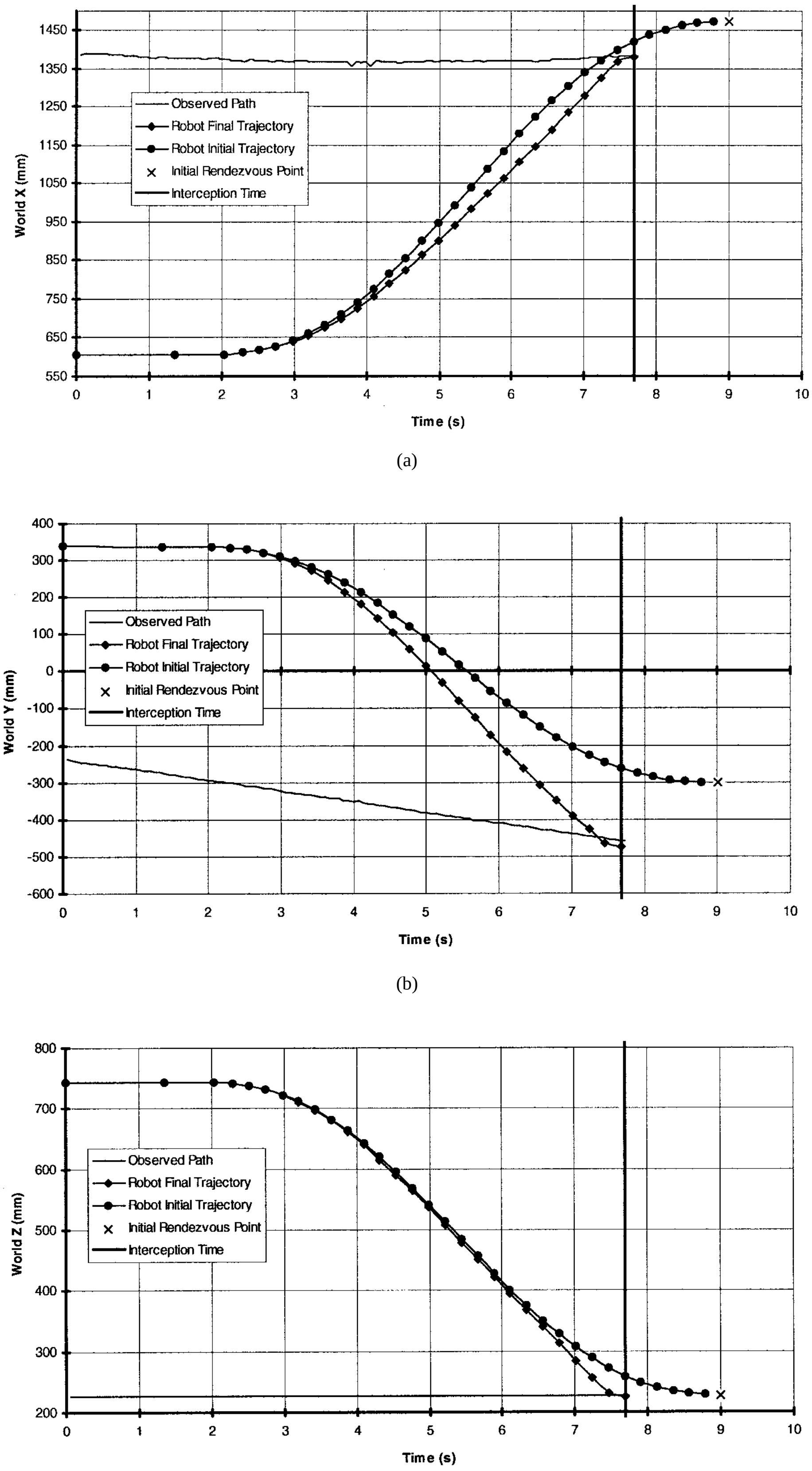


(a)

(b)

(c)

Fig. 10. Intercepting an object moving at 32 mm/s on a linear trajectory.

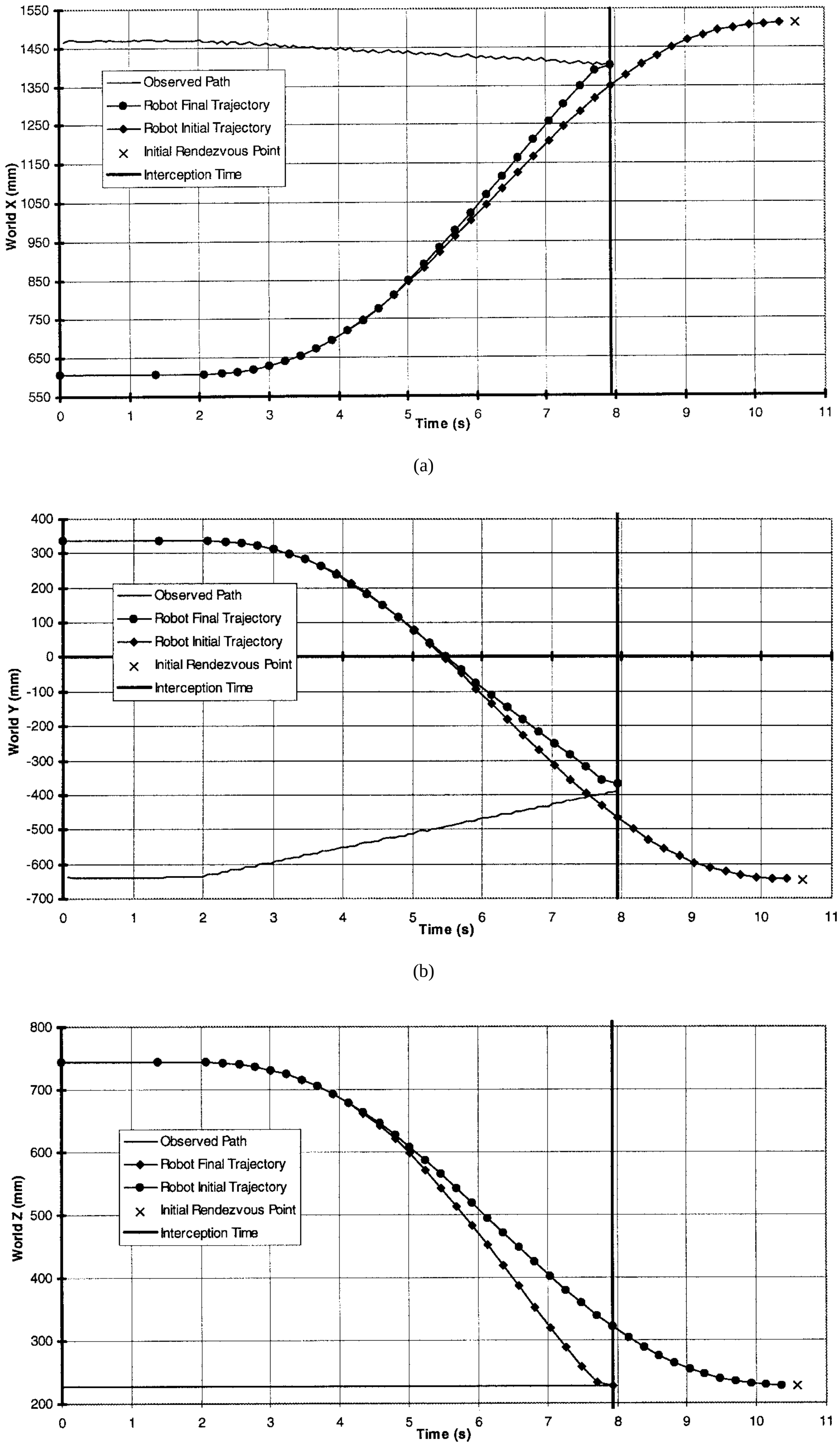


(a)

(b)

(c)

Fig. 11. Intercepting an object moving on a circular path of 280-mm radius at 30 mm/s.



(a)

(b)

(c)

Fig. 12. Intercepting an initially stationary object.

The trajectory implemented by the robot is the result of 44 replanned patches.

VII. Conclusions

The novel APPE system proposed in this paper complements other viable APPE systems previously suggested in the literature for the interception of moving objects. The key to the success of our APPE system is the effectiveness of its target-trajectory prediction and robot-trajectory planning strategies. As noted earlier, the object's motion unpredictability would diminish this effectiveness, whereas other APPE systems which advocate the use of fixed interception planes would be more successful.

References

[1] R. Sharma, J.-Y. Herve, and P. Cucka, "Dynamic robot manipulation using visual tracking," in *Proc. IEEE Int. Conf. Robotics and Automation*, Nice, France, May 1992, pp. 1844–1849.

[2] P. K. Allen, A. Timcenko, B. Yoshimi, and P Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 152–165, Apr. 1993.

[3] R. L. Andersson, *A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control*. Cambridge, MA: MIT Press, 1988.

[4] M. Lei and B. K. Ghosh, "Visually guided robotic tracking and grasping of a moving object," in *Proc. IEEE Conf. Decision and Control*, San Antonio, TX, Dec. 1993, pp. 1604–1608.

[5] T. H. Park and B. H. Lee, "An approach to robot motion analysis and planning for conveyor tracking," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 378–384, Mar./Apr. 1992.

[6] B. Hove and J. J. E. Slotine, "Experiments in robotic catching," in *Proc. Amer. Control Conf.*, Boston, MA, 1991, pp. 380–385.

[7] G. C. Buttazzo, B. Allotta, and F. P. Fanizza, "Mousebuster: A robot for real-time catching," *IEEE Trans. Contr. Syst. Technol.*, vol. 14, pp. 49–56, Feb. 1994.

[8] M. D. Mikesell and R. J. Cipra, "Development of a real time intelligent robotic tracking system," in Proc. *ASME 23rd Mechanism Conf.*, Sept. 1994, vol. 72, pp. 213–222.

[9] A. J. Koivo and N. Houshangi, "Real-time vision feedback for serving robotic manipulator with self-tuning controller," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, pp. 134–142, Jan./Feb. 1991.

[10] A. Kosaka and G. Nakazawa, "Vision-based motion tracking of rigid objects using prediction of uncertainties," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1995, pp. 2637–2644.

[11] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 14–35, Feb. 1993.

[12] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82D, pp. 35–45, 1960.

[13] A. C. Harvey, *Forecasting, Structural Time Series Models and the Kalman Filter*. New York: Cambridge Univ. Press, 1989.

[14] A. Gelb, J. F. Kasper, R. A. Nash, C. F. Price, and A. A. Sutherlannd, *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.

[15] Y. Bar-Shalom and T. Fortman, *Tracking and Data Association*. Boston, MA: Academic, 1988.

[16] E. A. Croft, "On-line planning for robotic interception of moving objects," Ph.D. dissertation, Dep. Mech. Eng., Univ. Toronto, Toronto, Ont., Canada, 1995.

[17] A. Kosaka and A. C. Kak, "Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties," *CVGIP: Image Understanding*, vol. 56, no. 3, pp. 271–329, Nov. 1992.

[18] H. P. Geering, L. Guzzella, S. A. R. Hepner, and C. H. Onder, "Time-optimal motions of robots in assembly tasks," *IEEE Trans. Automat. Contr.*, vol. AC-31, pp. 512–518, June 1986.

[19] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. Robotics Res.*, vol. 4, no. 3, pp. 3–17, 1985.

[20] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Automat. Contr.*, vol. AC-30, pp. 531–541, June 1985.

[21] E. A. Croft, B. Benhabib, and R. G. Fenton, "Near-time-optimal robot motion planning for on-line applications," *J. Robot. Syst.*, vol. 12, no. 8, pp. 553–567, Aug. 1995.

[22] R. Safaee-Rad, K. C. Smith, B. Benhabib, and I. Tchouhanov, "3D location estimation of circular features for machine vision," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 624–640, Oct. 1992.

[23] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the shelf TV cameras and lenses," *IEEE Trans. Robot. Automat.*, vol. RA-3, no. 4, pp. 323–344, Aug. 1987.

Damir Hujic (S'95–A'95) received the B.A.Sc. degree (with honors) and the M.A.Sc. degree, both in mechanical engineering, from the University of Toronto, Toronto, Ont., Canada, in 1993 and 1996, respectively.

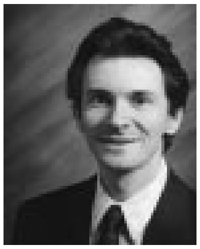
Following receipt of the B.A.Sc. degree, he worked at the Ontario Hydro Research Division, developing nondestructive techniques for inspecting CANDU nuclear reactors. He is currently with Celestica Inc., North York, Ont., Canada, working in the surface mount technology manufacturing industry. Past research topics include ultrasonic NDE, computer vision, image processing using wavelet transforms, and robot controls.

Mr. Hujic is an associate member of the Society of Automotive Engineers.

Elizabeth A. Croft (M'95) received the B.A.Sc. degree in 1988 from the University of British Columbia, Vancouver, B.C., Canada, the M.A.Sc. degree in 1992 from the University of Waterloo, Waterloo, Ont., Canada, and the Ph.D. degree in 1995 from the University of Toronto, Toronto, Ont., Canada, all in mechanical engineering.

During her post-graduate work, she held National Sciences and Engineering Research Council of Canada Scholarships and the Margaret McWilliams Pre-Doctoral Fellowship. Her dissertation work was in the area of robotic interception of moving objects. From 1988 to 1990, she worked in the consulting industry in the area of motor vehicle dynamics. In 1995, she joined the Industrial Automation Laboratory, Department of Mechanical Engineering, University of British Columbia, as an Assistant Professor and BC Packers Junior Chair of Industrial Automation and a member of the Centre for Integrated Computer Systems Research. Her research interests include industrial automation, robotics, soft computing, and sensor integration.

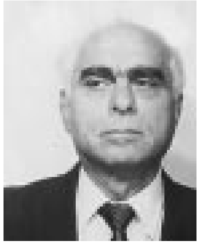
Prof. Croft is a Registered Professional Engineer in the Province of British Columbia, Canada, a senior member of the Society of Manufacturing Engineers and a member of the American Society of Mechanical Engineers.



Gene Zak received the B.A.Sc. degree (with honors) from the Division of Engineering Science (computer-aided manufacturing option) in 1986 and the M.A.Sc. degree in mechanical engineering in 1990 from the University of Toronto, Toronto, Ont., Canada, where he is currently working towards the Ph.D. degree in mechanical engineering.

Following receipt of the M.A.Sc. degree, he was with Spar Aerospace, where he worked on the International Space Station Project. His current research area is rapid prototyping. Past research topics include robot calibration, robot kinematic modeling, and machine vision.

Mr. Zak is a student member of the American Society of Mechanical Engineers and the Rapid Prototyping Association/Society of Manufacturing Engineers.



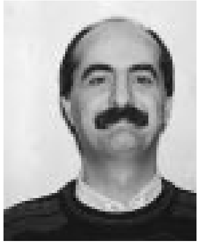
Robert G. Fenton received the Ph.D. degree in mechanical engineering from the University of New South Wales, Sydney, Australia.

He is currently a Professor in the Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ont., Canada. His areas of research interest cover kinematics, dynamics, stress analysis, robotics, and automation. He has published more than 250 papers in journals and conference proceedings and has coauthored a book.

James K. Mills (S'81–M'82) received the B.Sc. degree in electrical engineering, with a background in physics, from the University of Manitoba, Winnipeg, Man., Canada, and the Master's degree in electrical engineering and the Ph.D. degree in mechanical engineering from the University of Toronto, Toronto, Ont., Canada, in 1980, 1982, and 1988, respectively.

He is currently a Full Professor of Mechanical Engineering at the University of Toronto. Following receipt of the Ph.D. degree, he was a Senior Systems Engineer with DSMA Ltd., working on various projects, including captive trajectory systems. He has authored or coauthored more than 110 publications in the areas of dynamics and control of robotic systems. He has also presented several invited papers at international conferences and has given many invited presentations at institutes and universities around the world. His current research interests include the development of multirobot assembly systems, performance-related aspects of control system design, and learning control systems, as well as the application of control theory to manufacturing processes. He was a Visiting Scientist at the Centre for Artificial Intelligence and Robotics, Bangalore, India.

Dr. Mills has twice served on the IEEE International Conference on Robotics and Automation Review Committee and on several other conference committees.



Beno Benhabib (M'93) received the Ph.D. degree in mechanical engineering from the University of Toronto, Toronto, Ont., Canada, in 1985.

He is currently a Professor in the Department of Mechanical and Industrial Engineering, University of Toronto. His research interests are in the general area of computer-integrated manufacturing. His published work covers various aspects of robot-motion planning, machine vision, robotic sensors, and supervisory control of manufacturing systems. He has been a consultant to various Ontario manufacturers in the areas of CAD/CAM, robotics, and automated quality control.

Dr. Benhabib is a senior member of the Society of Manufacturing Engineers, a member of the American Society of Mechanical Engineers, and a Registered Professional Engineer in the Province of Ontario, Canada.