The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Perceiving Guaranteed Continuously Collision-free Robot Trajectories in an Unknown and Unpredictable Environment

Rayomand Vatcha and Jing Xiao

Abstract— **This paper addresses continuous collision-checking of a high-DOF robot trajectory in a completely unknown and unpredictable environment (i.e., obstacles are unknown and their motions are also unknown). In [1], the authors introduced how to discover, if a robot at configuration $\mathbf{q}$ at a future time t is *guaranteed* collision-free or not using the novel concept of the *dynamic envelope* and *atomic obstacles* based on sensing in such an unknown and unpredictable environment. In this paper, we further show that if a point $(\mathbf{q}, t)$ in the robot's configuration-time space (CT-space) is discovered collision-free, a neighborhood (CT-region) of $(\mathbf{q}, t)$ is also guaranteed collision-free. Based on that, given a continuous robot trajectory, we present a method to compute a set of discrete CT-points such that, if these points are discovered to be guaranteed collision-free, their associated collision-free neighborhood CT-regions contains the continuous trajectory, i.e., the trajectory is guaranteed continuously collision-free.**

I. Introduction

The ultimate goal of intelligent robotics is to enable a robot of high degrees of freedom (DOF) to work autonomously in an unknown and unpredictable environment. For the robot to move safely in such an environment, it must be able to discover guaranteed continuously collision-free trajectories in real-time. However, the most common practice in the robotics literature for collision-checking is to discretize a continuous path or trajectory into a sequence of discrete configurations or configuration-time points and just check if the robot is collision-free or not at these discrete points. A collision can be missed between two consecutive discrete points, especially if the size of an obstacle is not known beforehand. A finer discretization will increase computation cost but not be able to eliminate the problem.

Most of the work addressing continuous collision checking are for known environments, using the knowledge of obstacle motion. One approach formulates the trajectories of the robot and approximated obstacles in the environment as functions of time and finds the time instants when collision occurs analytically [2]. However, if the trajectory functions are nonlinear, solving for collision time instants can be difficult. The adaptive bisection algorithm [3] is based on the intuition that if the sum of the distances traveled by two objects is less than the minimum distance between them before traveling, then they cannot collide during their motions.

This research was supported by the National Science Foundation under Grant IIS-0742610.

R. Vatcha is a PhD student in Computer Science Department,University of North Carolina at Charlotte, USA. rvatcha@uncc.edu

J. Xiao is with the faculty of Computer Science Department, University of North Carolina at Charlotte, USA. xiao@uncc.edu

There are also approaches in the literature that focus on generating or approximating the continuous swept volume by a robot along a path or trajectory [4], [5], which can then be used to perform collision tests against obstacles. One approach [6] models the motion between two discrete configurations of an articulated robot in order to avoid generating the swept volume of individual links. Graphics hardware is then used to perform fast collision queries for approximated swept volumes in a virtual prototyping environment. Another approach [7] is focused on growing the physical robot's volume at discrete configurations along a path to form a continuous region for collision tests. These approaches are focused on the robot rather than the obstacles, which are mostly assumed static.

To our best knowledge, there is no method in the literature that has addressed continuous collision checking in a completely unknown and unpredictable environment, where obstacle geometries are not known and whether obstacles move or not and their motions cannot be predicted. We address the novel problem of continuous collision checking in such an environment in this paper.

The rest of the paper is organized as follows. Section II introduces basic assumption and notations used in the paper. Section III provides a review of how to use the concept of dynamic envelopes [1] to perceive collision-free configuration-time points (CT-points) for a robot in an unknown and unpredictable environment. Section IV describes a collision-free CT region associated with a collision-free CT point discovered via a dynamic envelope. Section V presents a method to detect in real-time if a robot trajectory is guaranteed continuously collision-free or not through checking a set of discrete configuration-time (CT) points. Section VI provides an implemented example and some experimental results. Section VII concludes the paper.

II. Assumption and Notations

Here we assume a high-DOF robot consisting of multiple polyhedral links. This is a reasonable assumption since a real robot's link is commonly modeled by a polygonal mesh or can be approximated by a polyhedral bounding box.

The following notations describe such a robot model in the Cartesian space (i.e., the physical space).

- l: the set of points constituting a polyhedral link (rigid body) of the robot.
- l_x: the set of vertices of l.
- R: the set of all points of all links of the robot, i.e., $R = \bigcup l$.

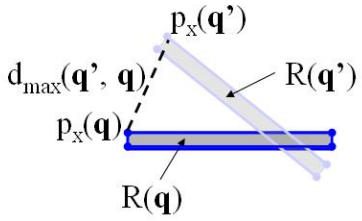


Fig. 1. Illustration of $d_{max}(\mathbf{q}', \mathbf{q})$ of a rod robot

- R_x: the set of all bounding vertices of all links, i.e., $R_x = \bigcup l_x$.
- $\mathbf{p}(\mathbf{q})$: the position of a point $p \in R$ in the Cartesian space when the robot is at configuration $\mathbf{q}$.
- $\mathbf{p}_x(\mathbf{q})$: the position of a point $p_x \in R_x$ in the Cartesian space, when the robot is at configuration $\mathbf{q}$.
- $R(\mathbf{q})$: the region occupied by the robot R at configuration $\mathbf{q}$.
- $d_{max}(\mathbf{q}', \mathbf{q})$: the distance defined below:

$$d_{max}(\mathbf{q}', \mathbf{q}) = \max_{\forall p_x \in R_x} \|\mathbf{p}_x(\mathbf{q}') - \mathbf{p}_x(\mathbf{q})\|. \quad (1)$$

Figure 1 illustrates $d_{max}(\mathbf{q}', \mathbf{q})$ of a rod robot (which can also be considered a rectangular link of a high-DOF robot).

We also use the following different temporal notations in the description of a robot's operation:

- τ: time of sensing.
- t: time of action.

III. Dynamic Envelope: A Review

For a robot to move in an unknown and unpredictable environment safely, it is necessary that the robot is able to sense on-line if it can be guaranteed collision-free at some time and place. In [1], an approach is introduced to discover from sensing whether a high-DOF robot (such as a manipulator) will be guaranteed collision-free at a configuration $\mathbf{q}$ at some future time t, i.e., whether the CT-point $\chi = (\mathbf{q}, t)$ is guaranteed collision-free. To do that, it uses the novel concept *dynamic envelope* complemented by *atomic obstacles*.

Atomic obstacles are low-level sensory data representing actual obstacles collectively in similar and simple geometry without distinguishing them. In other words, all actual (unknown) obstacles as a whole can be sensed as many atomic obstacles (with default geometry) and their respective locations at any sensing moment. Although the obstacles and their motions (or lack of motion) are unknown, an upper bound can be put on the changing rate of the environment in terms of a maximum possible speed v_{max} of each atomic obstacle. Of course, an atomic obstacle may have varied actual speeds in $[0, v_{max}]$. To be safe, v_{max} can be quite over-estimated, but the approach for discovering guaranteed collision-free CT-points is shown [1] to be robust for over-estimated v_{max}.

Now, we describe the concept of dynamic envelope.

Definition 1: For a CT-point $\chi = (\mathbf{q}, t)$, a *dynamic envelope* $E(\chi, \tau_i)$, as a function of sensing time $\tau_i \leq t$, is a closed surface enclosing $R(\mathbf{q})$ in the physical space so that the distance between any point on $E(\chi, \tau_i)$ and the surface of $R(\mathbf{q})$ is $v_{max}(t - \tau_i)$.

The following are major properties of a dynamic envelope $E(\chi, \tau_i)$. They capture non-worst case scenarios regarding atomic obstacle motions, without assuming any particular kinds of obstacle motion.

1) A dynamic envelope shrinks monotonically over sensing time with speed v_{max}, i.e., $E(\chi, \tau_i) \supset E(\chi, \tau_{i+m})$, where $m > 0$, $\tau_i < \tau_{i+m} \leq t$.
2) An atomic obstacle not on or inside $E(\chi, \tau_i)$ will never be on or inside $E(\chi, \tau_{i+m})$.
3) An atomic obstacle either on or inside $E(\chi, \tau_i)$ can be outside $E(\chi, \tau_{i+m})$, for certain τ_{i+m}, if not moving towards $R(\mathbf{q})$ in maximum speed v_{max}.

Suppose some atomic obstacles are on or inside the dynamic envelope initially. As the dynamic envelope shrinks over time, it is possible that at some sensing moment $\tau_j < t$, the dynamic envelope $E(\chi, \tau_j)$ is free of atomic obstacles. When that happens, the CT-point $\chi = (\mathbf{q}, t)$ is discovered guaranteed collision-free at τ_j (based on the above properties), and we call that the dynamic envelope *expires* at τ_j. Moreover, all the continuous CT-points in the interval $[(\mathbf{q}, \tau_j), (\mathbf{q}, t)]$ are also guaranteed collision-free.

Figure 2 shows an example, where $\chi = ((3, 3), 3)$, $v_{max} =$ 1 unit/s. The robot is a planar rod robot with fixed orientation and can translate along the plane. In this example, the whole workspace is assumed visible so that all obstacles are viewed as atomic obstacles. However, the work to be described in this paper does not require that assumption. Rather, as long as the concerned region for a trajectory is visible, we can determine if the trajectory is continuously collision-free or not.

IV. Collision-free CT-region Discovered along with a Collision-free CT Point

We have the following theorem.

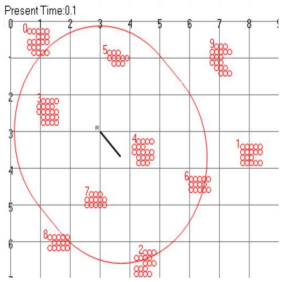
Theorem 1: When a CT-point $\chi = (\mathbf{q}, t)$ is discovered collision-free at sensing time τ_i, i.e., the dynamic envelope $E(\chi, \tau_i)$ is free of obstacles, a continuous neighborhood $F(\chi, \tau_i)$ of χ is also discovered collision-free, such that, for any CT-point $\chi' = (\mathbf{q}', t') \in F(\chi, \tau_i)$, its configuration $\mathbf{q}'$ satisfies:

$$d_{max}(\mathbf{q}', \mathbf{q}) \leq v_{max}(t - \tau_i), \quad (2)$$

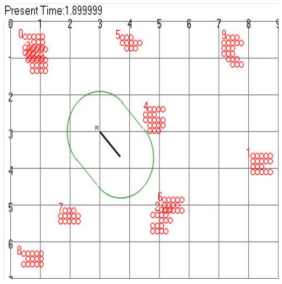
and its time t' satisfies

$$\tau_i \leq t' \leq t - \frac{d_{max}(\mathbf{q}', \mathbf{q})}{v_{max}} < t. \quad (3)$$

Proof: When inequality (2) is satisfied, $R(\mathbf{q}')$ is contained by the dynamic envelope $E(\chi, \tau_i)$, which is free of obstacles. If $\chi' = (\mathbf{q}', t')$ is collision-free, it means that the dynamic envelope of χ' is contained in the dynamic envelope of χ, i.e., $E(\chi', \tau_i) \subset E(\chi, \tau_i)$, and $\tau_i \leq t'$. Since $d_{max}(\mathbf{q}', \mathbf{q})$ is



(a) $E(\chi, 0.1)$ contains atomic obstacles



(b) $E(\chi, 1.89)$ shrinks and does not contain an atomic obstacle

Fig. 2. A rod robot in an environment of unknown obstacles. The unknown obstacles are modeled by atomic obstacles as red circles. $\chi = ((3,3),3)$ is detected guaranteed collision-free at $\tau_i = 1.89$s. These figures are reprint from [1].

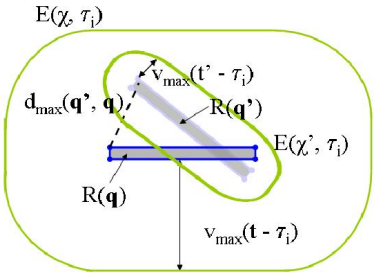


Fig. 3. Illustration of inequality (4)

the maximum distance between two corresponding points of $R(\mathbf{q}')$ and $R(\mathbf{q})$, we have

$$d_{max}(\mathbf{q}', \mathbf{q}) + v_{max}(t' - \tau_i) \leq v_{max}(t - \tau_i), \qquad (4)$$

which can be simplified to

$$t' \leq t - \frac{d_{max}(\mathbf{q}', \mathbf{q})}{v_{max}}.$$

Thus, the theorem is proven. ∎

Figure 3 illustrates the proof for a rod robot.

Corollary 1: If $\chi' \in F(\chi, \tau_i)$ for some $\tau_i < t$, then $\chi' \in F(\chi, \tau_{i+m}), \tau_{i+m} \in [\tau_i, t'], m = 0, 1, 2, ...$

Proof: From inequality (3), subtracting t', we have

$$0 \leq (t - t') - \frac{d_{max}(\mathbf{q}', \mathbf{q})}{v_{max}}$$

which, multiplying v_{max}, leads to:

$$d_{max}(\mathbf{q}', \mathbf{q}) \leq v_{max}(t - t')$$

From the above, because $\tau_{i+m} \leq t'$, we have

$$d_{max}(\mathbf{q}', \mathbf{q}) \leq v_{max}(t - \tau_{i+m}) \qquad (5)$$

and also

$$\tau_{i+m} \leq t' \leq t - \frac{d_{max}(\mathbf{q}', \mathbf{q})}{v_{max}} < t. \qquad (6)$$

With (5) and (6), based on Theorem 1, the corollary holds. ∎

We next characterize the geometry of the CT-region $F(\chi, \tau_i)$ below, based on Theorem 1 and Corollary 1.

- $F(\chi, \tau_i)$ is on the time interval $[\tau_i, t]$.
- The region of $F(\chi, \tau_i)$ on the time-slice τ_i contains all the configurations that satisfy inequality (2), whose size and shape depend on (a) the robot kinematics, and (b) the size of the dynamic envelope $E(\chi, \tau_i)$.
- At the time-slice t, $F(\chi, \tau_i)$ contains the single CT-point $\chi = (\mathbf{q}, t)$.
- Based on Theorem 1, for $\tau_i \leq t' \leq t(\chi, \mathbf{q}')$, where

$$t(\chi, \mathbf{q}') = t - \frac{d_{max}(\mathbf{q}', \mathbf{q})}{v_{max}} \qquad (7)$$

all CT-points $\chi' = (\mathbf{q}', t')$ are in $F(\chi, \tau_i)$. Equation (7) shows that the smaller $d_{max}(\mathbf{q}', \mathbf{q})$ is, the longer is the hyper-line from $(\mathbf{q}', \tau_i)$ to $(\mathbf{q}', t(\chi, \mathbf{q}'))$ in $F(\chi, \tau_i)$.
- $F(\chi, \tau_{i+m}) = F(\chi, \tau_i) - F_p$, where $F_p = \{(\mathbf{q}', t') | (\mathbf{q}', t') \in F(\chi, \tau_i), t' < \tau_{i+m}\}$.

As an example, Figure 4 illustrates the geometry of the CT-region $F(\chi, \tau_i)$ of the same rod robot as in Figure 2 with only two translational degrees of freedom. The region of $F(\chi, \tau_i)$ on the time slice τ_i is a circular disc.

V. Perceiving Guaranteed Continuously Collision-Free Trajectory

For an n-DOF robot, a continuous trajectory segment Γ from CT-point $\chi_s = (\mathbf{q}_s, t_s)$ to CT-point $\chi_e = (\mathbf{q}_e, t_e)$ can be formulated as:

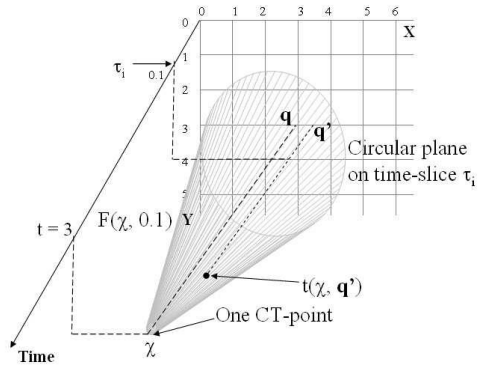
$$\Gamma = \mathbf{q}(t) = [q_1(t), ..., q_n(t)]^T, \quad t_s \leq t \leq t_e, \qquad (8)$$

where $q_1(t), ..., q_n(t)$ are continuous functions of time t for respective joint variables.

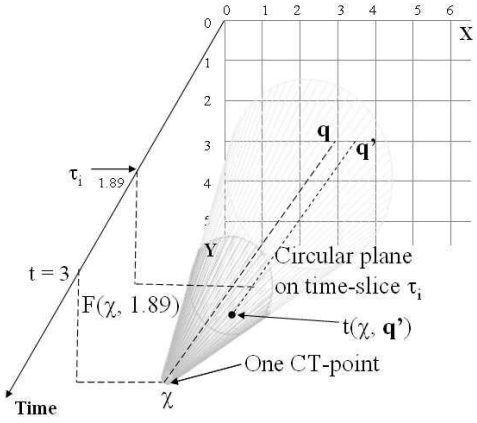
In order to discover if Γ is continuously collision-free or not, our idea is to find a set of sparse CT-points $Q(\Gamma) = \chi_j, j = 1, ..., k$, such that if each $\chi_j = (\mathbf{q_j}, t_j) \in Q(\Gamma)$ is discovered collision-free at sensing time $\tau_j \leq t_s$, then the (continuous) CT-points on Γ are contained in $\bigcup F(\chi_j, \tau_j)$, i.e., the entire trajectory segment Γ is discovered continuously collision-free at time instant $\max(\tau_j) \leq t_s$. We now explain the details of how this works below.

A. Associate Γ to the CT-region $F(\chi_1, \tau_1)$ of a single CT-point $\chi_1 = (\mathbf{q}_1, t_1)$

Let us first consider the condition for the CT-region $F(\chi_1, \tau_1)$ of a single CT-point $\chi_1 = (\mathbf{q}_1, t_1)$ to include all



(a) $F(\chi, 0.1)$ based on the dynamic envelope $E(\chi, 0.1)$ shown in Figure 2(a)



(b) $F(\chi, 1.89)$ based on the dynamic envelope $E(\chi, 1.89)$ shown in Figure 2(b). $F(\chi, 1.89) \subset F(\chi, 0.1)$

Fig. 4. The geometry of CT-region $F(\chi, \tau_i)$ for the 2D rod robot of Figure 2.

(continuous) CT-points satisfying Γ. From the inequality (3) in Theorem 1, the following condition can be easily derived:

$$d_{max}(\mathbf{q}(t), \mathbf{q}_1) \leq v_{max}(t_1 - t), \forall t \in [t_s, t_e] \qquad (9)$$

From equation (1), $d_{max}(\mathbf{q}(t), \mathbf{q}_1)$ can be further computed as

$$d_{max}(\mathbf{q}(t), \mathbf{q}_1) = \max_{\forall p_x \in R_x} \|\mathbf{p}_x(\mathbf{q}(t)) - \mathbf{p}_x(\mathbf{q}_1)\|$$

where $\mathbf{p}_x(\mathbf{q}(t))$ can be obtained from $\mathbf{q}(t)$ by forward kinematics.

Next we want to choose the CT-point $\chi_1 = (\mathbf{q_1}, t_1)$ to satisfy the condition (9), which shows that the complicated non-linear function $d_{max}(\mathbf{q}(t), \mathbf{q}_1)$ has to be bounded by the straight line $f(t) = v_{max}(t_1 - t)$ during interval $[t_s, t_e]$. Figure 5 illustrated the condition (9) visually. If we choose $\mathbf{q_1}$ to be the end configuration of Γ, i.e., $\mathbf{q}_1 = \mathbf{q}_e$, then, at least for a time interval immediately before t_e, $d_{max}(\mathbf{q}(t), \mathbf{q}_1)$ monotonically decreases as a function of t until $d_{max}(\mathbf{q}(t), \mathbf{q}_1) = 0$ when t_e is reached. We now need to select a t_1 to make $d_{max}(\mathbf{q}(t), \mathbf{q}_1)$ below the straight-line segment $f(t) = v_{max}(t_1 - t)$. To increase the area below $f(t)$

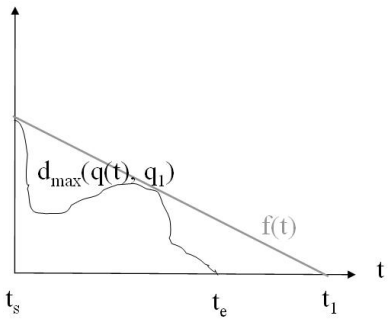


Fig. 5. Illustration of the condition (9)

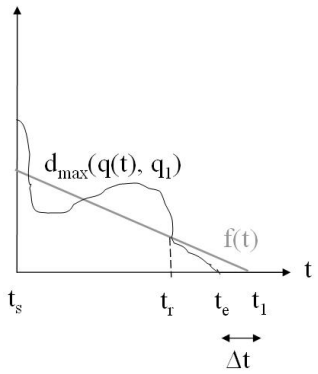


Fig. 6. A situation after $f(t)$ is shifted Δt to end at t_1

in order to satisfy condition (9), we have to select $t_1 \geq t_e$ i.e., shifting $f(t)$ along the t axis in the positive direction. Figure 5 illustrates a situation after $f(t)$ is shifted to end at t_1.

The subsequent question is how much t_1 should be greater than t_e (or $f(t)$ should be shifted). Apparently the greater the t_1, the more likely condition (9) will be satisfied. However, a greater t_1 can mean a delayed discovery of trajectory Γ being collision-free, if it is indeed collision-free. This is because, for the same configuration $\mathbf{q}_1$, it may take longer time to discover if the CT-point $\chi_1 = (\mathbf{q}_1, t_1)$ is collision-free or not than the CT-point $\chi_1' = (\mathbf{q}_1, t_1')$, if $t_1' < t_1$. It can be shown that $\tau_1 \geq \tau_1' + 0.5(t_1 - t_1')$, i.e., the sensing moment when χ_1 is discovered collision-free will be more than $0.5(t_1 - t_1')$ later than the sensing moment when χ_1' is discovered collision-free. Moreover, if the CT-point χ_1 is discovered collision-free at a time later than t_s, i.e., $\tau_1 > t_s$, then the CT-point is no longer useful for discovering trajectory Γ to be collision-free *beforehand*. It is too late.

Thus, we have to limit t_1 to be only slightly later than t_e to avoid much delay, but then this small shift Δt of the line $f(t)$ may not result in condition (9) to be satisfied (see Figure 6). This is why we want to consider using not just one CT-point, but a set of CT-points $Q(\Gamma) = \chi_j, j = 1, ..., k$, to discover if trajectory Γ is continuously collision-free or not.

B. Associate Γ to $\bigcup F(\chi_j, \tau_j)$ of a set of CT-points $\{\chi_j\}$

Following the previous section V.A, for some time interval $[t_r, t_e]$ ending at t_e, $d_{max}(\mathbf{q}(t), \mathbf{q}_j)$ is below $v_{max}(t_1 - t)$.

Now we need to find out the value of t_r. There are the following possible results:

- Case 1: $t_r = t_s$, implying that $d_{max}(\mathbf{q}(t), \mathbf{q}_1)$ is below $f(t)$ for the entire time interval $[t_s, t_e]$;
- Case 2: t_r is the greatest root of equation $d_{max}(\mathbf{q}(t), \mathbf{q}_1) = v_{max}(t_1 - t)$.

If case 1, the single CT-point $(\mathbf{q}_1, t_1)$ is sufficient for discovering if the trajectory Γ is continuously collision-free or not (as described in section V.A).

Case 2 means that only the part of Γ for the time interval $[t_r, t_e]$, where $t_s < t_r < t_e$, can be contained by $F(\chi_1, \tau_1)$ of the CT-point $\chi_1 = (\mathbf{q}_1, t_1)$. That further means that the dynamic envelope of χ_1 can be used to only discover if that part of Γ is continuously collision-free or not. Therefore, we need to find additional CT-points for checking if the remaining part of Γ, for the time interval $[t_s, t_r]$, is continuously collision-free. We use Algorithm 1 to find such a set $Q(\Gamma)$ of CT-points.

Algorithm 1 $Q(\Gamma)$ generator

1: Input Γ for the time interval $[t_s, t_e]$
2: $t_r = t_e$
3: $j = 0$; $Q(\Gamma) = \emptyset$
4: **repeat**
5: $\quad j = j + 1$
6: $\quad$ Apply the process of section V.A to find $\chi_j = (\mathbf{q}_j, t_j)$ for Γ in $[t_s, t_r]$
7: $\quad$ Add χ_j to $Q(\Gamma)$
8: $\quad$ Find new t_r from equation:

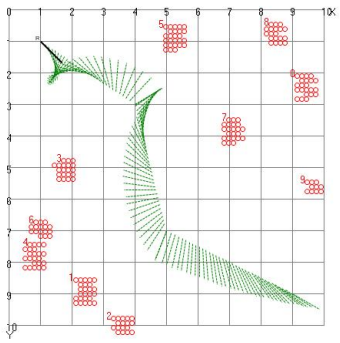
$$d_{max}(\mathbf{q}(t), \mathbf{q}_j) - v_{max}(t_j - t) = 0 \qquad (10)$$

9: **until** $t_r = t_s$ (i.e., Case 1)
10: **return** $Q(\Gamma)$

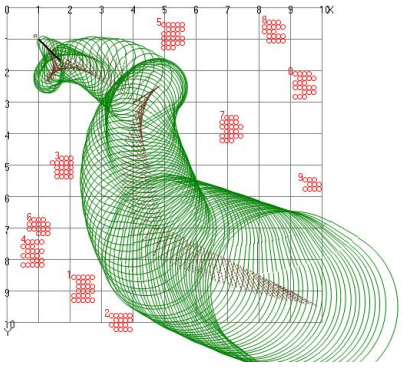
Note that the value of Δt (which is the amount of shift of $f(t)$ along positive time axis – see section V.A) affects the number of CT-points in $Q(\Gamma)$. Recall that we want Δt to be small to avoid time delay in discovering if Γ is collision-free or not. However, if Δt is too small, there can be too many CT-points in $Q(\Gamma)$, and thus the cost of collision checking (of the CT-points via their dynamic envelopes) increases. So instead of using a fixed Δt, our strategy is to adapt the value of Δt to balance the need of fast discovery of collision-free CT points and that of fast collision-checking.

Note also that solving the non-linear equation (10) to find roots may require numerical techniques. There are derivative-free[1] fast numerical methods [8], [9], [10] which guarantee to find the roots of a non-linear equation $g(x) = 0$ in an interval $[a, b]$, if $g(a)g(b) < 0$. If $g(a)g(b) > 0$, we can sample within this interval for potential roots, which requires evaluating a non-linear function. If the function is a high order polynomial, there are fast numerical methods for evaluation [11]. In the case of a robot, the L.H.S. of equation (10) can be converted to a polynomial (by variable substitution in transcendental equations).

[1]No need to differentiate L.H.S. of (10)



(a) Γ shown by the sequence of CT-points in $Q(\Gamma)$



(b) Dynamic envelopes of the CT-points in $Q(\Gamma)$

Fig. 7. Piece-wise continuous trajectory Γ consisting of three segments

VI. An Implemented Example

Our approach was implemented in a simulation environment for a mobile rod robot having three degrees of freedom, i.e. $\mathbf{q} = [x, y, \theta]^T$, where the origin of the robot frame was at one end of the rod. The rod robot (which can be considered a link of a high-DOF robot) moves in an unknown environment with atomic obstacles shown as red circles (as in Figure 2). A continuous trajectory segment for the rod robot was generated with cubical polynomials [12]. Step 8 of Algorithm 1 was trivially implemented using the intermediate value theorem and Brent's method [8] for root finding. Figure 7(a) shows the configurations of a piece-wise continuous trajectory Γ consisting of three continuous segments. The discrete configurations shown along Γ are from the set $Q(\Gamma)$ generated by Algorithm 1. Γ starts from time $t_s = 2s$ and ends at time $t_e = 4.4402s$. Figure 7(b) shows a snapshot of dynamic envelopes of the CT-points in $Q(\Gamma)$ when the continuous Γ is discovered collision-free. In this environment, $v_{max} = 1$ unit/s.

Table I shows the results for the same trajectory Γ but in three environments of different levels of obstacle dynamics, as indicated by different values of v_{max} (which is the upper-bound of obstacle speeds). It shows the smallest time interval and the largest time interval between two consecutive CT-points $(\mathbf{q}_j, t_j)$ and $(\mathbf{q}_{j+1}, t_{j+1})$ in $Q(\Gamma)$, the number of CT-points in $Q(\Gamma)$, and Δt in each case. In all cases, the time to generate $Q(\Gamma)$ was within 1s. Note that we set the minimum

TABLE I

RESULTS OF $Q(\Gamma)$ FOR THE SAME CONTINUOUS MOTION-SEGMENT Γ IN DIFFERENT ENVIRONMENTS

v_{max} (unit/s)	Smallest Interval (s)	Largest interval (s)	$\|Q(\Gamma)\|$	Δt (s)
1	.0154	.0570	99	0.1
1.5	.0250	.0858	60	0.1
2	.0158	.1023	41	0.1

In all cases, the time to generate $Q(\Gamma)$ was within 1s.

value for Δt to be 0.1s, and there was no need to increase Δt further in this example for all cases. Note also that as v_{max} increases, the number of CT-points in $Q(\Gamma)$ decreases. This can be explained from inequality (2): the increase of v_{max} enlarges the dynamic envelope $E(\chi, \tau_i)$ of any CT-point $\chi = (\mathbf{q}, t)$ and therefore enlarges the associated CT-region $F(\chi, \tau_i)$ for any sensing moment τ_i.

The attached video clip shows the 3-DOF rod robot simultaneously discovering and executing continuously collision-free trajectory segments in an unknown dynamic environment. Each trajectory segment Γ is shown by the sequence of configurations in $Q(\Gamma)$, along with the dynamic envelopes of the corresponding CT-points. Each $Q(\Gamma)$ is generated on-line by the approach introduced in this paper. Whether the CT-points in $Q(\Gamma)$ are collision-free or not are checked in real-time via their dynamic envelopes as described in [1]. When a CT-point is discovered collision-free, its dynamic envelope at that moment changes to green from red. Therefore, any trajectory segment covered by a sequence of green envelopes are discovered continuously collision-free, which the robot can move along. The red envelopes cover the trajectory segments that are uncertain, i.e., not yet discovered collision-free (and they may not be collision-free). Note that after the robot finishes executing a collision-free trajectory segment, if it cannot find a subsequent collision-free trajectory segment to follow, it is forced to stop until it can find one. When it is forced to stop, it may get hit by an obstacle. As seen in the movie for $v_{max} = 1.5$ unit/s, the robot gets hit by an obstacle (and momentarily change color to blue) during a forced stop.

VII. CONCLUSIONS

This paper addresses the novel problem of how to perceive continuously collision-free trajectories for a robot to operate in an unknown and unpredictable environment, where the obstacles and their motions (or lack of motions) are completely unknown. Given a continuous trajectory Γ in the configuration-time space (CT-space) of the robot, the idea is to determine a set $Q(\Gamma)$ of discrete configuration-time points (CT-points), such that when each of these CT-points, $(\mathbf{q}_j, t_j), j = 1, 2, ...$, is discovered collision-free based on the notion of dynamic envelope [1], a continuous neighborhood of $(\mathbf{q}_j, t_j)$ in the CT-space, i.e., a CT-region, is also discovered collision-free, and the union of these continuous CT-regions for all points in $Q(\Gamma)$ contains Γ, meaning that Γ is continuously collision-free. To be efficient, our method avoids minimum-distance computation, which is the most computationally expensive component of existing collision-checking algorithms.

The introduced method can be used by a real-time robot motion planner (such as RAMP[13]) to test if a trajectory candidate is continuously collision-free in an unknown 3-D environment in real time[2]. An implemented example involving a 3-DOF rod robot and the attached video demonstrated that. However, we plan to further test the efficiency of the method on a 7-DOF manipulator. We will also improve the root-finding algorithm for generating $Q(\Gamma)$, which bears most of the computation cost.

REFERENCES

[1] R. Vatcha and J. Xiao, "Perceived CT-space for motion planning in unknown and unpredictable environments," in *the Eighth International Workshop on the Algorithmic Foundations of Robotics (WAFR 2008)*, (Guanajuato, Mexico), December 2008.

[2] A. Schweikard, "Polynomial time collision detection for manipulator paths specified by joint motions," in *IEEE Transactions on robotics and automation*, pp. 865–870, 1991.

[3] F. Schwarzer, M. Saha, and J. claude Latombe, "Adaptive dynamic collision checking for single and multiple articulated robots in complex environments," *IEEE Transactions on Robotics*, vol. 21, pp. 338–353, 2005.

[4] S. Cameron, "Collision detection by four-dimensional intersection testing," *IEEE Transactions on Robotics and Automation*, vol. 6, pp. 291–302, 1990.

[5] A. Foisy and V. Hayward, "A safe swept volume method for collision detection," in *The Sixth Int. Symp. of Robotics Research*, (Pittsburgh (PE)), pp. 61–68, Oct 1993.

[6] S. Redon, M. C. Lin, D. Manocha, and Y. J. Kim, "Fast continuous collision detection for articulated models," *Journal of Computing and Information Science in Engineering*, vol. 5, no. 2, pp. 126–137, 2005.

[7] B. Baginski, "Efficient dynamic collision detection using expanded geometry models," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1714–1719, 1997.

[8] R. P. Brent, "An algorithm with guaranteed convergence for finding a zero of a function," *Computer Journal*, vol. 14, pp. 422–425, 1971.

[9] J. C. P. Bus and T. J. Dekker, "Two efficient algorithms with guaranteed convergence for finding a zero of a function," *ACM Trans. Math. Softw.*, vol. 1, no. 4, pp. 330–345, 1975.

[10] D. Le, "An efficient derivative-free method for solving nonlinear equations," *ACM Trans. Math. Softw.*, vol. 11, no. 3, pp. 250–262, 1985.

[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, August 2007.

[12] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[13] J. Vannoy and J. Xiao, "Real-time Adaptive Motion Planning (RAMP) of mobile manipulators in dynamic environments with unforeseen changes," in *IEEE Transactions on Robotics*, vol. 24(5), pp. 1199–1212, Oct. 2008.

[2]Even though only 2-D examples are used in this paper to be more illustrative, as one can see, no part in the method is restricted to 2-D space.