A Closest Point Algorithm for Parametric Surfaces with Global Uniform Asymptotic Stability

Volkan Patoglu and R. Brent Gillespie

Abstract— **We present an algorithm that determines the point on a convex parametric surface patch that is closest to a given (possibly moving) point. Any initial point belonging to the surface patch converges to the (possibly moving) closest point without ever leaving the patch. Thus the algorithm renders the patch invariant and is globally uniformly asymptotically stable. The algorithm is based on a control problem formulation and solution via a switching controller and common control Lyapunov function. Analytic limits of performance are available, delineating values for control gains needed to out-run motion (and shape) and preserve convergence under discretization. Together with a top-level Voronoi diagram-based switching algorithm, the closest point algorithm treats parametric models formed by tiling together convex surface patches. Simulation results are used to demonstrate invariance of the surface patch, global convergence, limits of performance, relationships between low-level and top-level switching, and a comparison to competing Newton-iteration based methods.**

I. Introduction

A fast and reliable collision detection algorithm is essential for computer simulation of dynamical systems, including systems of rigid bodies and deformable bodies. Collision detectors are also important components of software tools for computer-aided design and computer-aided manufacturing. Many collision detectors are based on closest point algorithms that determine the pair of closest points on two disjoint bodies. A closely related problem is the determination of the furthest points on two intersecting bodies. Also, the determination of the point on a body closest to a penetrating point is used in penalty-based haptic rendering algorithms. The penetrating point might be the image of a stylus tip in the hand of an operator acting through a haptic interface. The vector connecting the closest point and the image of the stylus tip determines the magnitude and direction of the reaction force to be rendered. Likewise, closest point algorithms that can determine the penetration depth and direction (defined suitably) between two intersecting bodies can be used to render reaction forces appropriate to the intersection of the image of a fingertip and a virtual object through a thimble-based haptic interface.

In our previous work [1] [2], we developed a closest point tracking method with local convergence properties that operates on pairs of parametric surface patches. That algorithm can be called a *direct* method, in that it operates directly on parametric surface representations such as NURBS surfaces rather than on polyhedra such as those that result from the tessellation of such surfaces. For a full literature survey on direct and indirect closest point algorithms for parametric surfaces, see [2]. In the present paper, we also concentrate on a *direct* method. Our reasons for pursuing direct rather than indirect methods have to do with certain properties that can be developed and proven much more easily for the direct methods. These properties will not be available for the indirect methods (the polyhedral methods accompanied by tessellation algorithms), primarily because tessellation masks the construction of the parametric models. We further believe that these properties will be essential for any method that is to be extended for use in an advanced collision detector. We use the term *advanced* to describe a collision detector that can treat intersecting, non-convex, and possibly deformable bodies formed by tiling together multiple convex surface patches. (Note that a body itself may be non-convex, even when all component surface patches are convex, depending on the orientation and composition of the patches.) We anticipate that direct methods (again, with those certain properties) will be especially advantageous for the case of deforming bodies since parametric models offer compact geometric representations even under deformation. Polygonal models, on the other hand, require calculation of new tessellations as the bodies deform. Parametric models also support solid mechanics-based continuous deformation models which, in certain cases, may prove computationally more efficient than finite element approaches. Although very high resolution tessellations can be achieved at interactive speeds, smoothness and continuity independent of a particular rendering are intrinsic properties of direct methods.

In this paper we present a closest point algorithm with certain properties not possessed by our previous algorithms. The algorithm applies to a convex parametric surface patch, a mathematical object that we will define carefully in the body of the paper. The certain provable properties are *convergence*: all initial points on the patch converge to the closest point, and *invariance*: all paths starting in the patch never leave the patch. These two properties taken together yield an algorithm that is *globally* asymptotically convergent.

In the algorithm's nominal form it determines the point on a convex surface patch that is closest to a given (possibly moving) point. In forms that are simple extensions, it determines the pair of closest points on two disjoint convex surface patches. In forms that we claim are further simple extensions, but will not fully lay out here for lack of space, it determines the furthest pair between two intersecting convex surface patches, and thus becomes what may be called an *extremal* algorithm.

Although convergence may seem like a desirable property (and indeed a property possessed by many available algorithms) invariance and global convergence at first glance may seem unwarranted. We argue, however, that global convergence is essential if a given closest point algorithm is to be successfully extended to serve as the basis of an advanced collision detector. For example, if a convex body is made of

Volkan Patoglu and R. Brent Gillespie (corresponding author) are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA. (e-mail: vpatoglu@umich.edu and brentg@umich.edu)

tiled together convex surfaces patches, the point on that body's surface that is closest to a given point in its interior will make a path with discontinuous jumps as the given point moves across a medial axis [3] of the body. With discontinuous jumps, it is much more difficult to guarantee that the initial condition will lie within a local region of attraction of the analytical closest point.

Another very desirable feature in any closest point algorithm is the availability of analytical limits of performance, or the availability of adjustments that can be made to preserve the properties in the face of demands made upon the algorithm. What adjustments can be made to preserve the properties under relative motion of the bodies, especially fast relative motion? How are those adjustments dependent on the sharpness of body shapes? What tradeoffs exist between the algorithm speed (in terms of large time step used in a discretization) and convergence rates?

Many closest point algorithms are based on a Newton Iteration [4] [5] [6] [7]. A well-known deficit in Newton's iteration, however, is a limited region of attraction. Even for convex problems, Newton's iteration cannot yield global convergence. Its region of attraction is local and comprises a not necessarily connected set. In our work, we build algorithms based on the explicit use of feedback control. Newton's Flow can be interpreted as a special case of feedback control (using a particular control law) which, as can be shown using control analysis, inherits only *local* convergence. In our work presented here, we rely on control laws that can be proven globally convergent. Interestingly, these control laws are also simpler to implement. As a natural product of our adoption of control theoretic design tools, our algorithm is equipped with gains that can be tuned to preserve the desirable properties under various demands. One can even imagine tuning gains adaptively to yield maximum computational efficiency when the body shape and motion allows it, without risking loss of global convergence. This is possible since the boundaries on the gain values where the properties break down can be easily evaluated.

In the following, we first carefully define a *convex* surface patch and outline the design of our closest point algorithm in Section II. In Section III we present the control law that renders the closest point globally asymptotically convergent, even when there exists relative motion between the point and surface patch. The controller is presented in the form of a theorem and the proof (based on a common Lyapunov function) is outlined in detail. The basic closest point algorithm involves only a single patch and point. Extensions to two patches are related in Remarks to the Theorem. In Section IV we present four simulations which are designed to feature each of the critical properties of our algorithm: invariance, global region of attraction, selection of gains given relative motion and discretization, and the de-coupled nature of the top-level Voronoi-region based switching and low-level switching between the surface patch interior and its bounding curves. Finally, in Section V, we revisit the relationship between our controls-based approach to collision detector design and the Newton iteration based approaches espoused in the past by other researchers.

II. Preliminaries

A. Convex Surface Patch

In this paper we treat bodies described by a collection of tiled together surface patches. We restrict the surface patches to be convex and for now, we also require the bodies to be convex [1]. That is, the convex surface patches shall be oriented and joined together at their boundaries in such a way that a line joining any two points in the interior of the compact body will be wholly contained in that body. While defining convexity for a body is straightforward even when it is a tiled body, defining convexity for a surface patch requires special consideration. Like the definition for a convex space curve [8], we have defined a convex surface patch as *any* patch cut from a compact convex body. Convexity of the surface patch depends on convexity of the body from which it is cut, and not on the curvature of the bounding curves that lie in the surface. Thus the projection of a surface patch onto a plane may well produce a planar area which is not convex. Likewise, smoothness of the surface patch depends on smoothness of the body from which it was cut (we require at least $\mathcal{C}^2$ smoothness). Note that the body from which a patch is cut is to be distinguished from the body formed by the tiled together patches. The convex tiled body is the intersection of the bodies that play host to the patch cutting operations. To accommodate a requirement for our convergence proof that appears below, we further define a convex surface patch as 'nice' when the angle between any two surface normals (pointing outward) subtend less than 180 degrees. This requirement is non-restrictive as any convex surface patch that is not nice can easily be divided into at most two convex surface patches that are nice.

B. Feedback Based Algorithm

In this paper, and without loss of generality, we shall use four curves intersecting in four distinct vertices to define the boundary of a surface patch. For example, Figure 1 shows a convex surface patch S composed of its interior $\breve{S}$ and the four curves c_i $(i = 1 \ldots 4)$ that bound it. Further, a parametric surface patch may be conveniently parameterized using parameters u and v whose domains are restricted to [0,1]. Thus the boundaries are the $u = 0, v = 0, u = 1, v = 1$ curves and the whole patch S is described as the image of the vector mapping $\mathbf{f}(u, v) : ([0, 1] \times [0, 1]) \to \Re^3$. Given a point $\mathbf{Q}$ lying outside a convex surface patch S, there exists a *unique* point $\mathbf{P}^*$ of the patch that is closer to $\mathbf{Q}$ than any other point of the patch [9]. We will call $\mathbf{P}^*$ the closest point and (u^*, v^*) the parameters of the closest point.

The central problem we address in this paper is how to determine the closest point $\mathbf{P}^*$ given surface patch S and point $\mathbf{Q}$ and how to maintain $\mathbf{P}^*$ given relative motion between S and $\mathbf{Q}$. To determine $\mathbf{P}^*$, we use an algorithm that causes an initialization point $\mathbf{P_0}$ lying anywhere in the patch to converge to the closest point. The algorithm drives to zero the tangent-plane projections of the vector $\mathbf{\Delta R}$ from the best current guess $\mathbf{P}$ (referred to as the witness point) to $\mathbf{Q}$. Maintenance of the

[1] Later, after a means of tracking the *furthest* points on two intersecting convex surface patches has been developed, the restriction of convexity on the tiled-together bodies can be lifted. These topics will be addressed in future papers.

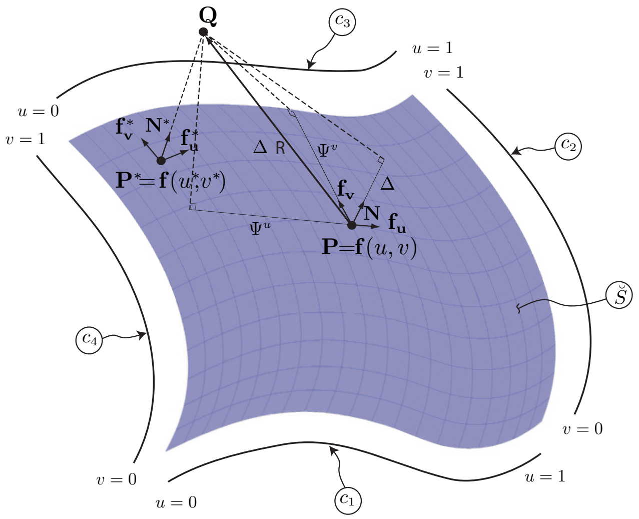


Fig. 1. A surface patch S, parameterized by $(u, v) \in ([0, 1] \times [0, 1])$, is composed of its interior $\breve{S}$ and its bounding curves c_i $(i = 1, \ldots , 4)$. Point $\mathbf{Q}$ (the free moving point outside the patch), $\mathbf{P}^*$ (the corresponding closest point on the patch) and $\mathbf{P}$ (the current best estimate of the closest point) are indicated in the figure. Vectors $\mathbf{f_u}$, $\mathbf{f_v}$ (the unit surface tangents in the u, v directions) and $\mathbf{N}$ (the unit surface normal) define the surface frame at any point on the patch. Vector $\mathbf{\Delta R}$ (the vector from $\mathbf{P}$ to $\mathbf{Q}$) is expressed in the surface frame where Ψ^u, Ψ^u and Δ are the resulting measure numbers. As the best estimate $\mathbf{P}$ approaches the closest point $\mathbf{P}^*$, the measure numbers in the tangent directions Ψ^u, Ψ^u, referred as the projection errors, converge to zero, and the measure number Δ converges to E, the minimum distance between S and $\mathbf{Q}$.

closest point $\mathbf{P}^*$ comes for free, since convergence of $\mathbf{P}$ to $\mathbf{P}^*$ ensures continual tracking as the closest point $\mathbf{P}^*$ changes location on the surface patch S under the effects of relative motion between $\mathbf{Q}$ and S and the shape of S. Note that, when the difference vector $\mathbf{\Delta R}$ is projected onto the surface tangents $\mathbf{f_u}$ and $\mathbf{f_v}$, these projections are called the *projection errors* and labelled Ψ^u and Ψ^v, respectively.

Our algorithm is based on the formulation of a nonlinear control problem and its solution takes the form of a feedback stabilizing controller. The "plant" whose output the controller drives is an integrator wrapped around the differential kinematics of the error vector $\mathbf{\Delta R}$. The outputs of the plant are the parameters u and v that locate $\mathbf{P}$. The objective of the controller is to manipulate u and v until the projections Ψ_u and Ψ_v of ΔR onto the surface tangents $\mathbf{f_u}$ and $\mathbf{f_v}$ at $\mathbf{P}$ are driven to zero. The "simulation" or numerical integration of the differential kinematics then produces the convergent algorithm. Feedback is used to stabilize the integration. Note that similar feedback stabilization techniques have been used to solve for the inverse kinematics of robot manipulators [10] and to solve for the motion of constrained multibody systems [11].

In complete analogy to the solution of a manipulator's inverse kinematics by a feedback stabilized simulation of its differential kinematics (see [10], Fig. 3.12), Figure 2 shows the feedback stabilized integration of the differential error kinematics. The term *error kinematics* refers to the dependence of the projection errors Ψ^u and Ψ^v on the parameters u and v, on the location of $\mathbf{Q}$, and on the directions of the surface tangents $\mathbf{f_u}$ and $\mathbf{f_v}$ at $\mathbf{P}$. Let the vector x contain the parameters u and v and let $\Psi(x)$ in the feedback loop contain the error kinematics Ψ^u and Ψ^v. Differentiating the error kinematics produces an expression which may be encapsulated in $\dot{\Psi} = M(x)\dot{x} + b(x)$, whose inverse appears (solved for $\dot{x}$, renamed w) in the forward loop in Figure 2. Note that, by convexity of the surface patch S, the projection errors Ψ are zero only if the witness point $\mathbf{P}$ is the closest point $\mathbf{P}^*$. For any other point $\mathbf{P}$, the projection vector Ψ has a direction and a non zero magnitude that can be used to drive the parameters u and v to u^* and v^*.

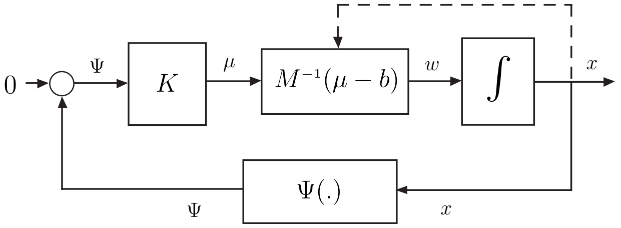


Fig. 2. This figure demonstrates the feedback stabilized integration of the differential error kinematics of the closest point problem. The projection errors $\Psi(x)$ are regulated to zero by the proportional control law K that drives the *inverse* differential error kinematics, whose convergent integration results in the desired surface parameters.

In [2], we have shown that the control law $w = M^{-1}(-K\Psi - b)$ renders the closest point solution *locally* asymptotically stable and an estimate on the basin of attraction of this controller is given by the set of points around $\mathbf{P}^*$ where the M matrix is positive definite. While solving for a manipulator's inverse kinematics with a feedback stabilized simulation of its differential kinematics, one may consider different control laws than the one resulting from the direct inversion of its differential kinematics. In [10], Lyapunov type arguments are used to demonstrate that utilizing the Jacobian transpose in the control law produces performance comparable to that using the Jacobian inverse. In the next section, we will prove with a control Lyapunov function that asymptotic convergence to $\mathbf{P}^*$ is preserved even when the term $w = M^{-1}(\mu - b)$ in the controller is replaced by a positive constant. The convergence properties of this simplified control law follow from the fact that any positive definite matrix can be bounded by its eigenvalues and the effect of motion can be counteracted by the feedback term given sufficiently large controller gains.

C. Voronoi Switching to Locate the Active Patch

If the modeling environment is represented with a *single* patch and the algorithm is initialized *sufficiently close* to $\mathbf{P}^*$, then the proposed controller can guarantee convergence within the patch even though $\mathbf{P}^*$ changes location on the patch under the effect of relative motion. However, consideration of a single parametric surface patch by itself is not quite sufficient, since within a parametric modeling environment, objects are generally modeled using collections of tiled-together surface patches. When the object model consists of tiled-together surface patches, detection of the *active* patch on which $\mathbf{P}^*$ lies becomes an important concern. At a given instant of time, the closest point solution may lie on any of the surface patches and the active patch is subject to change due to relative motion.

To update the active patch with respect to the relative motion of the bodies, we propose a *feature* based switching algorithm.

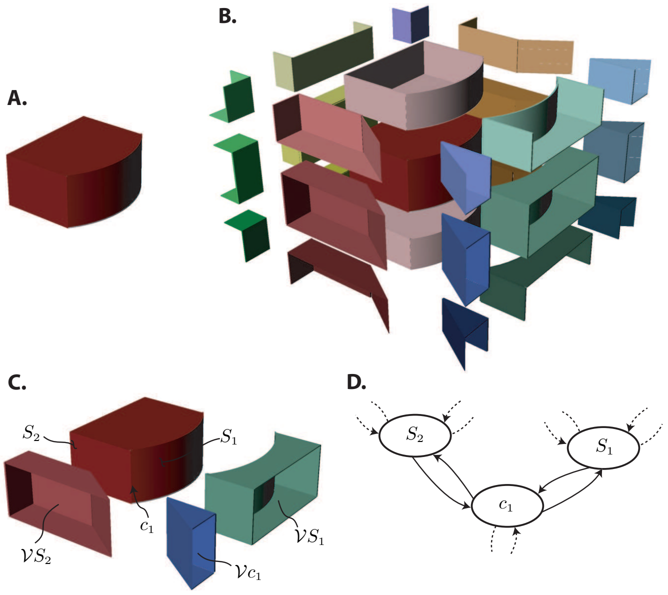


Fig. 3. (A,B) illustrate an object consisting of tiled-together surface patches and all of its Voronoi regions. (C) presents only a portion of the Voronoi regions and labels them appropriately. (D) shows the automaton associated with (C) that governs the discrete dynamics of the Voronoi based switching algorithm.

Feature based switching algorithms are well established in the literature for the collision detection of polygonal models [12] [13]. The Voronoi diagram of an object partitions the space around it into distinct regions. Feature based switching algorithms rely on the fact that if an external point $\mathbf{Q}$ is in the Voronoi region of a feature, it is closer to this corresponding feature than any other feature.

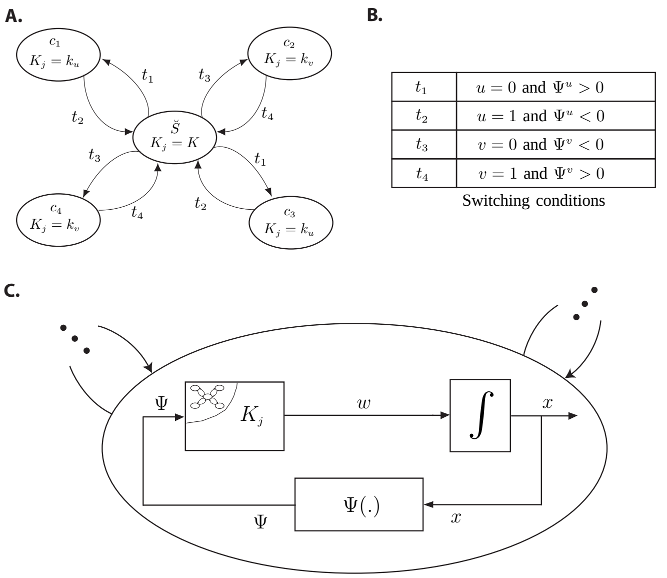
Voronoi diagrams also exist for parametric objects formed using tiled-together surface patches [3] [14]. For example, Figure 3 (A) illustrates such an object consisting of several surface patches whereas (B) shows all of its Voronoi regions. Figure 3 (C) presents only a portion of these Voronoi regions with appropriate labels and (D) illustrates the automaton associated with (C) that governs the discrete dynamics of the Voronoi based switching algorithm. Note that determination of Voronoi regions for curved objects can be computationally expensive; however, the Voronoi switching algorithm requires only a numerical pre-computation of these regions before the simulation is started. Consequently, this step does not affect the real time performance of the algorithm.

The proposed Voronoi based switching algorithm is virtually the same as the Lin-Canny closest feature algorithm [12]. The algorithm triggers updates to the closest feature when the point $\mathbf{Q}$ crosses between the Voronoi regions of the object. The discrete dynamics of this switching algorithm can be modeled by an automaton constructed according to the connectedness of the object's features. For example, Figure 3 (D) shows a portion of such an automaton. Incorporation of the Voronoi based switching algorithm with the feedback controller results in a hybrid control system that can handle object models built from tiled-together patches. In our previous work [2], details of the Voronoi based switching algorithm and its incorporation with the feedback controller are discussed in detail and will not be further elaborated here.

D. Boundary Switching for Global Convergence

Incorporation of the feedback controller with the Voronoi based switching algorithm extends our results presented earlier for the feedback controller to multiple patches, however initialization sufficiently close to $\mathbf{P}^*$ is still required for the asymptotic convergence of this hybrid algorithm. The requirement to initialize sufficiently close to $\mathbf{P}^*$ is quite restrictive and it is desirable to design an algorithm that can be initialized anywhere within the active patch. In particular, special attention must be paid to the constraints imposed by the boundaries to achieve global convergence within a surface patch.

Having two unconstrained degrees of freedom, the controller as presented is feasible at any point of a surface without boundaries; however, when applied to a surface patch, there is no guarantee that the updated witness points will stay within the patch boundaries. To guarantee that the parameters u and v locating the witness point stay within the defined range (which is constrained to $([0,1] \times [0,1])$), we propose to saturate the parameters at the boundaries to keep the witness points on the boundary curves. The saturated version of the control algorithm is utilized whenever the witness point is on a boundary curve and the main control algorithm attempts to drive it outside the boundary. Saturation is implemented by simply determining the component of control signal that attempts to drive the witness point outside the boundary and setting it to zero.



t_1	$u = 0$ and $\Psi^u > 0$
t_2	$u = 1$ and $\Psi^u < 0$
t_3	$v = 0$ and $\Psi^v < 0$
t_4	$v = 1$ and $\Psi^v > 0$

Switching conditions

Fig. 4. This three-part figure illustrates the incorporation of several controller layers to obtain the overall hybrid control algorithm. (A) represents the automaton that decides on the controller gains for the lower level feedback loop. This automaton is composed of the modes corresponding to the surface patch interior $\breve{S}$ and the bounding curves c_i $(i = 1, \ldots, 4)$ and tracks the mode changes due to motion of the witness point $\mathbf{P}$ on the closed patch. (B) shows the switching conditions for autonomous mode changes of the boundary switching. (C) indicates the overall controller architecture driving $\mathbf{P}$ to $\mathbf{P}^*$. The lower level feedback controller maintains the current estimate of the closest point by continually driving the projection errors to zero while the boundary switching selects controller gains that guarantee convergence for any initialization within the patch. The Voronoi switching forms the highest level of the controller and keeps track of the active patch when the objects in the modeling environment consist of tiled-together surface patches.

As a consequence of the saturated control law at the patch boundaries, another switching layer is added to the overall controller at a lower level than the Voronoi based switching algorithm. The discrete dynamics of this switching at the boundaries is completely decoupled from the dynamics of Voronoi based switching algorithm and is dictated instead by the current parameters of the witness point. The boundary switching is best represented using an automaton with five distinct modes and such an automaton is illustrated in Figure 4 (A). Each mode in the automaton corresponds to a different set of gains for the feedback controller. The mode switches within the automaton take place depending on the current parameters of the witness point. A summary of the switching rules are shown in Figure 4 (B).

The global controller for a surface patch has two interacting layers: the automaton which selects the proper set of controller gains K depending on the current state of the witness point and the feedback control law which makes use of these gains to update the witness point. On top of these two control layers, the Voronoi based switching algorithm is put in place to keep track of the active patch due to the existence of multiple patches. An abstraction of the overall controller is presented in Figure 4 (C).

In the next section, it will be shown that, for any initialization within the active surface patch, uniform asymptotic convergence of the witness points to the closest point is guaranteed by the proposed switching controller.

III. A Globally Convergent Closest Point Algorithm

In this section, we consider the problem of finding the minimum distance between a point and a convex surface patch when both of the bodies to which they are attached are allowed to undergo rigid body motion with respect to one another. Below, we state and prove a theorem that guarantees global uniform asymptotic convergence of solutions when the controller gains are chosen to be sufficiently high to compensate for the motion of the bodies. We provide a practically implementable lower bound for these controller gains and show global convergence of solutions when the algorithm is initialized using any point within the patch.

The algorithm relies on a controller to generate $(\dot{u}, \dot{v})$ whose integration produces (u, v) that converges to (u^*, v^*). The switching nature of the controller guarantees that given any initialization (u_0, v_0) in the patch, the controller drives (u, v) to the minimum distance solution (u^*, v^*) without ever leaving the patch. That is, the switching controller renders the patch positively invariant.

Before stating the theorem, we will introduce some non-restrictive assumptions on the types of motion allowed. First, we will require the motion be continuous to assure Lipschitz continuity of the closest distance between the convex bodies [15]. An upper bound on the speed of the relative motion is also required. Given the angular velocity vector $^{\mathbf{N}}\boldsymbol{\omega}^{\mathbf{A}}$ of A (the body to which the surface patch is fixed) in a world reference frame N , we will define the upper bound $\mathcal{B}$ as the Euclidian norm of the vector $(^{\mathbf{N}}\boldsymbol{\omega}^{\mathbf{A}} \times \mathbf{Q})$. This bound is not restrictive and is only required so that a fast enough controller can be designed to compensate for the perturbing effects of motion.

Let the first fundamental matrix for the surface patch S be denoted by $\mathcal{I} = \begin{bmatrix} \mathcal{E} & \mathcal{F} \\ \mathcal{F} & \mathcal{G} \end{bmatrix}$ and let α be its largest eigenvalue. Also define ζ as $\left\| \frac{\boldsymbol{\Delta}\mathbf{R}}{\|\boldsymbol{\Delta}\mathbf{R}\|} - \mathbf{N}^* \right\|$. Note that ζ and $(\Psi^{u2} + \Psi^{v2}) = \|\boldsymbol{\Delta}\mathbf{R} - \Delta\mathbf{N}\|$ approach zero with the same rate as the witness point $\mathbf{P}$ converges to $\mathbf{P}^*$.

Theorem 1: If the image of the mapping $\mathbf{f}(u, v) : ([0, 1] \times [0, 1]) \rightarrow \Re^3$ defines a 'nice' rigid convex parametric surface patch S, the point $\mathbf{Q}$ is in the external Voronoi region of S, $\mathbf{Q}$ and $\mathbf{f}$ are in continuous motion with respect to one another, and given controller gains satisfying $K \geq \frac{\mathcal{B}\,\zeta\,\|\boldsymbol{\Delta}\mathbf{R}\|}{\alpha\,(\Psi^{u2}+\Psi^{v2})}$, $k_u \geq \frac{\mathcal{B}\,\zeta\,\|\boldsymbol{\Delta}\mathbf{R}\|}{\mathcal{G}\,\Psi^{u2}}$ and $k_v \geq \frac{\mathcal{B}\,\zeta\,\|\boldsymbol{\Delta}\mathbf{R}\|}{\mathcal{E}\,\Psi^{v2}}$, then the switching controller

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{cases} -k_v \begin{bmatrix} 0 \\ \Psi^v \end{bmatrix}, & \begin{array}{l} \text{if } u = 0 \text{ and } \Psi^u > 0 \\ \text{or } u = 1 \text{ and } \Psi^u < 0 \end{array} \\ -k_u \begin{bmatrix} \Psi^u \\ 0 \end{bmatrix}, & \begin{array}{l} \text{if } v = 0 \text{ and } \Psi^v > 0 \\ \text{or } v = 1 \text{ and } \Psi^v < 0 \end{array} \\ -K \begin{bmatrix} \Psi^u \\ \Psi^v \end{bmatrix}, & \text{otherwise} \end{cases} \tag{1}$$

renders the minimum distance point $\mathbf{P}^*$ uniformly asymptotically stable over the whole surface patch S.

The proof is omitted here but it is based on a *common control Lyapunov function* which is defined as the difference between the Euclidian norm of the the vector $\boldsymbol{\Delta}\mathbf{R}$ and the minimum distance E between the point and the surface patch,

$$V = \|\boldsymbol{\Delta}\mathbf{R}\| - E. \tag{2}$$

Being a common control Lyapunov function, V is positive definite and decresent. To prove uniform asymptotic stability of the algorithm, the negative definiteness of the time derivative of the control Lyapunov function is to be shown.

Since the boundaries of the surface patch constrain the allowable motion directions of a candidate point whenever it is on a boundary, the feasible control inputs are also constrained at the boundaries. The difference between feasible control directions on the boundaries and on the interior of the surface patch results in different control modes, within each of which proper control laws are to be designed.

In the proof, it is shown that rendering the time derivative of the control Lyapunov function negative definite in each controller mode is possible with a switching controller. Feasibility of the control laws in each control mode are also analyzed and proper switching conditions are supplied. Although the controller is of a switching nature, the existence of a common Lyapunov function guarantees that the stability is uniform over the set of all switching signals [16].

Since the proposed controller renders the time derivative of the common control Lyapunov function negative definite at any point on the surface patch except the minimum distance solution, we can claim that the common control Lyapunov function is actually a common Lyapunov function and the control algorithm is uniformly asymptotically stable [16]. Note that the common Lyapunov function guarantees uniform asymptotic stability, which is to say, stability holds under arbitrary switching, so analysis of the switching sequence is not required within the proof. Moreover, since the whole surface patch is rendered positively invariant under the proposed control law, we can also claim that the region of attraction of the proposed controller is the whole surface patch (including the boundaries), and the algorithm is *globally* uniformly asymptotically stable (GUAS).

Remark 1: Global uniform exponential stability (GUES) can be achieved by the same controller setting the controller gains as $k_v = \frac{\|\Delta R\|^2}{\Psi^{v2}}$, $k_u = \frac{\|\Delta R\|^2}{\Psi^{u2}}$ and $K = \frac{\|\Delta R\|^2}{(\Psi^{u2}+\Psi^{v2})}$. However, the control input w required to achieve exponential stability grows unbounded as the candidate point approaches the closest point. Therefore, exponential convergence can be obtained only to very close vicinity of the solution (a ball of diameter ϵ around the closest point). This result is practically satisfactory since convergence rate is of utmost importance when the candidate point is away from the solution.

Remark 2: When there is no relative motion between the bodies, the lower bound on the controller gains K, k_u and k_v simplifies to zero.

Remark 3: The presented controller treats all possible motion between the two bodies as perturbations and makes use of sufficiently high gains to suppress them. However, whenever possible, it may be desirable to take advantage of motion to achieve even faster convergence rates at a cost of a slightly more complex control law. This idea results in an enhanced version of the controller given as

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{cases} \frac{1}{G} \begin{bmatrix} 0 \\ -k_v \Psi^v + b_v \end{bmatrix}, & \text{if } u = 0 \text{ and } \Psi^u > 0 \\ & \text{or } u = 1 \text{ and } \Psi^u < 0 \\ \frac{1}{F} \begin{bmatrix} -k_u \Psi^u + b_u \\ 0 \end{bmatrix}, & \text{if } v = 0 \text{ and } \Psi^v > 0 \\ & \text{or } v = 1 \text{ and } \Psi^v < 0 \\ I^{-1} \begin{bmatrix} -K\Psi^u + b_1 \\ -K\Psi^v + b_2 \end{bmatrix}, & \text{otherwise} \end{cases} \tag{3}$$

where

$$\begin{aligned} b_v = & \frac{\Psi^u}{\Psi^v} \frac{1}{2} \left[1 - sign\left(\Psi^u ({}^N\boldsymbol{\omega}^A \times \mathbf{Q}) \cdot \mathbf{f_u}\right)\right] ({}^N\boldsymbol{\omega}^A \times \mathbf{Q}) \cdot \mathbf{f_u} \\ & + \frac{1}{2} \left[1 - sign\left(\Psi^v ({}^N\boldsymbol{\omega}^A \times \mathbf{Q}) \cdot \mathbf{f_v}\right)\right] ({}^N\boldsymbol{\omega}^A \times \mathbf{Q}) \cdot \mathbf{f_v}, \end{aligned}$$

$$\begin{aligned} b_u = & \frac{\Psi^v}{\Psi^u} \frac{1}{2} \left[1 - sign\left(\Psi^v ({}^N\boldsymbol{\omega}^A \times \mathbf{Q}) \cdot \mathbf{f_v}\right)\right] ({}^N\boldsymbol{\omega}^A \times \mathbf{Q}) \cdot \mathbf{f_v} \\ & + \frac{1}{2} \left[1 - sign\left(\Psi^u ({}^N\boldsymbol{\omega}^A \times \mathbf{Q}) \cdot \mathbf{f_u}\right)\right] ({}^N\boldsymbol{\omega}^A \times \mathbf{Q}) \cdot \mathbf{f_u}, \end{aligned}$$

and

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \left[1 - sign\left(\Psi^u ({}^N\boldsymbol{\omega}^A \times \mathbf{Q}) \cdot \mathbf{f_u}\right)\right] ({}^N\boldsymbol{\omega}^A \times \mathbf{Q}) \cdot \mathbf{f_u} \\ \frac{1}{2} \left[1 - sign\left(\Psi^v ({}^N\boldsymbol{\omega}^A \times \mathbf{Q}) \cdot \mathbf{f_v}\right)\right] ({}^N\boldsymbol{\omega}^A \times \mathbf{Q}) \cdot \mathbf{f_v} \end{bmatrix}$$

Note that this control law takes advantage of the motion whenever motion helps convergence and cancels it as much as possible whenever motion acts as a disturbance.

Remark 4: Since the controller and its associated dynamics are implemented in discrete time, the impact of discretization on the stability properties should be considered. Discretization introduces upper limits on the controller gains that depend on the integration method and step size chosen. There exist standard techniques whereby the convergence rate (determined by controller gains) and discretization step size can be traded off against one another while maintaining stability. In [17] and [18] standard discrete time controller design techniques are utilized to calculate an upper bound on controller gains given an explicit integration method and fixed integration step size. With these techniques, it becomes possible to preserve the stability of the algorithm after discretization.

Remark 5: Although the above theorem is given for the minimum distance solution between a convex surface and a point, it is relatively straightforward to extend the controller to operate on two convex surface patches (denoted $\mathbf{f}(u, v)$ and $\mathbf{h}(r, s)$ respectively) to find the minimum distance between them. If the surface patches are convex and the minimum distance point lies within the external Voronoi region of the corresponding patch, one can follow a similar Lyapunov analysis using the same proposed control Lyapunov function to prove that, as long as both candidate points lie inside the patch and with proper choice of gains, a similar controller renders the minimum distance solution globally uniformly asymptotically stable. Extension to the enhanced version of the controller is also straightforward.

IV. Simulation Results

We have developed computer simulations to demonstrate the important features of our algorithm. Our simulations are implemented in MATLAB and sample results are presented below. The first simulation highlights the importance of low-level boundary switching and shows the invariance of the surface patch under the switching controller. The second simulation demonstrates the global convergence of the our controller and characterizes the region of attraction of Newton iteration based methods. Importance of controller gain selection under relative motion and discretization is the theme of the third simulation whereas the top level Voronoi switching and its decoupled nature is shown in the fourth simulation. Common to the first three simulations is the convex surface patch whose definition is given in Table I.

TABLE I

Definition of the Convex NURBS Patch Used in Simulations of Figures 5,6, and 7

order	[3 3]				
control points (x, y, z, w)	(0,0,0,1)	(3,0,2,1)	(5,0,3,1)	(8,0,3,1)	(10,0,0,1)
	(1,3,3,1)	(3,3,5,1)	(5,3,6,1)	(8,3,5,1)	(9,3,3,1)
	(2,5,5,1)	(3,5,7,1)	(5,5,8,1)	(8,5,7,1)	(8,5,5,1)
	(1,8,3,1)	(3,8,5,1)	(5,8,6,1)	(8,8,5,1)	(9,8,3,1)
	(0,10,0,1)	(3,10,2,1)	(5,10,3,1)	(8,10,2,1)	(10,10,0,1)
knots u	$[0\ 0\ 0\ \frac{1}{3}\ \frac{2}{3}\ 1\ 1\ 1]$				
knots v	$[0\ 0\ 0\ \frac{1}{3}\ \frac{2}{3}\ 1\ 1\ 1]$				

A. *Invariance of a Surface Patch*

The four-part Figure 5 shows a simulation that illustrates the convergence behavior of an initialization P_0 on the surface patch S. (A) shows the points that locate the witness points in each of the simulation snapshots. (B) demonstrates the mode changes of the low level controller automaton as the convergence takes place. At startup, the first witness point

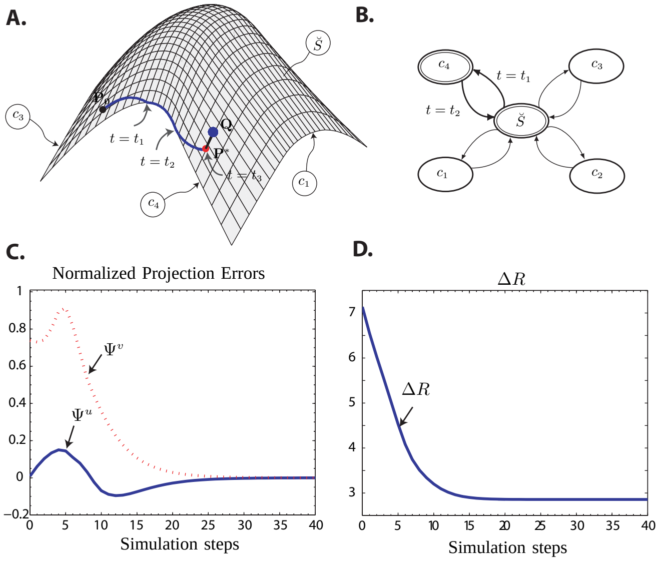


Fig. 5. This four-part figure shows the convergence of an initialization $\mathbf{P_0}$ through several mode changes demonstrating invariance of the surface patch under the proposed controller. (A) shows the path traced on the patch by the witness points whereas (B) illustrates the mode changes the controller goes through during convergence. The witness point, initialized within the surface patch $\breve{S}$, hits the surface boundary c_4 at $t = t_1$, traces along c_4, and leaves c_4 at $t = t_2$ converging to the solution at $t = t_3$. Correspondingly, in (B) the mode switches of the controller take place at $t = t_1$ and $t = t_2$ when the switching rules are satisfied. (C) demonstrates evolution if the normalized projection errors Ψ^u and Ψ^v while the algorithm converges to the closest point solution as shown in (D).

(which is the initialization point $\mathbf{P_0}$) belongs to the interior of the surface patch $\breve{S}$; consequently, the automaton governing the controller gains is in the corresponding mode. At time $t = t_1$, the witness point $\mathbf{P}$ hits the bounding curve c_4 and the controller command would otherwise attempt to drive $\mathbf{P}$ outside this boundary of the patch. However, these conditions satisfy the rules for boundary switching and an immediate controller mode change takes place, invoking a saturated version of the control law. The new controller mode (with the saturated control law) stays valid as long as the unrestricted controller attempts to drive $\mathbf{P}$ outside the boundaries of S. At $t = t_2$ the use of the unrestricted controller becomes feasible and the controller goes through another mode change removing the restriction on the witness points to move along c_4. After this mode change, the witness points are free to move in two degrees of freedom and converge to $\mathbf{P}^*$ at time $t = t_3$.

(C) demonstrates the change in the normalized projection errors Ψ^u, Ψ^v and (D) illustrates the evolution of $\|\mathbf{\Delta R}\|$ as the algorithm converges to the closest point solution $\mathbf{P}^*$. As shown in (C) and (D), even on a convex patch, the projection errors do not monotonically decrease while the witness point converges to $\mathbf{P}^*$. This is due to the fact that the projection errors are evaluated at the witness point $\mathbf{P}$ and not at the closest point $\mathbf{P}^*$. An initialization sufficiently close to $\mathbf{P}^*$ is required for the monotonic behavior of the projection errors. An important feature of our algorithm is its ability to handle any initialization whereas a Newton iteration based method relies on the monotonic decrease of the projection errors and therefore requires sufficiently close initializations.

This simulation demonstrates the positive invariance of the surface patch and the importance of the boundary switching to achieve convergence for any initialization within the patch. The boundary switching is required since without the saturation of the control law at the boundaries, invariance of the patch cannot be guaranteed. Without the boundary switching, invariance does not exist because the level curves of the Lyapunov function for the unrestricted controller go out of the patch boundaries. The important feature of the boundary switching is to restrict the witness point to move along the boundary, but doing so without sacrificing the asymptotic convergence of the algorithm.

B. Region of Attraction for Various Controllers

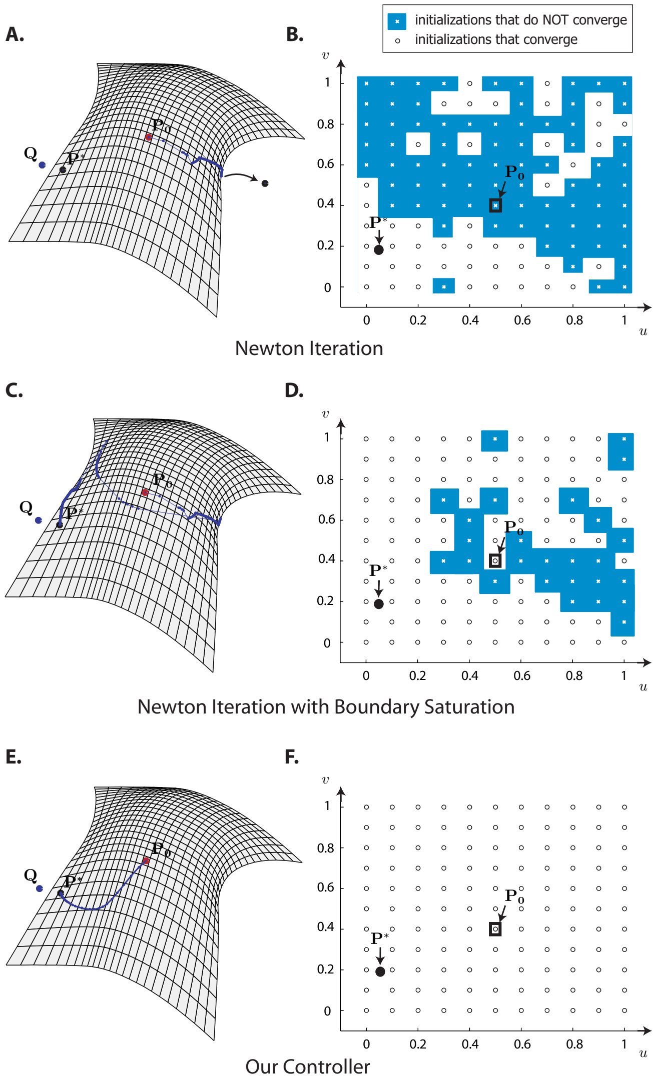


Fig. 6. In this Figure, convergence characteristics and the region of attraction for three different control laws are demonstrated. The figures on the left illustrate the path that witness points follow when initialized at $\mathbf{P_0}$. Point $\mathbf{Q}$ is the external point, $\mathbf{P}^*$ is the closest point solution. In these simulations, no relative motion between the bodies was used. The figures on the right show the region of attraction of the corresponding controller in the parameter space of the patch. The dark regions with white crosses × correspond to initializations that do not converge whereas the circles ○ on the white background illustrate the basin of attraction. (A) and (B) belong to Newton based iteration with no special boundary control. (C) and (D) are for Newton based iteration with saturation at the boundaries. (E) and (F) demonstrate our proposed controller. Global convergence of our controller can be compared to limited regions of attraction for Newton based methods.

The objective of this simulation is to reveal the global basin of attraction of our controller and to compare it with the limited region of attraction for Newton iteration based methods. Figure 6 shows the convergence characteristics and the region of attraction for three different control laws. The figures on the left illustrate the paths that witness points follow when all three simulations are initialized at the same point $\mathbf{P_0}$. To be able to compare our algorithm to Newton based methods, no relative motion between the bodies was used. Point $\mathbf{Q}$ represents the external point whereas $\mathbf{P}^*$ labels the closest point solution. In the figures on the right, the circles $\circ$ on the white background label the parameters for initialization points that converge to $\mathbf{P}^*$ whereas the dark regions with white crosses $\times$ correspond to parameters for initialization points that do not converge. Consequently, the regions with white background illustrate the basin of attraction of the corresponding controller in the parameter space of the surface patch.

(A) and (B) demonstrate a controller based on Newton iteration. During this simulation, no special control law is used at the patch boundaries. (A) shows that for this particular initialization $\mathbf{P_0}$, the Newton based controller starts in the wrong direction and fails to converge to $\mathbf{P}^*$ by going out of the boundaries of the patch. (B) demonstrates the region of attraction of this controller which is a relatively small, unconnected set.

(C) and (D) are again for a controller based on Newton iteration; however, this time saturation is utilized at the patch boundaries. (C) shows that for the particular initialization $\mathbf{P_0}$, the controller acts like the previous controller until the witness point hits the patch boundary and the parameters of the witness point saturate. At this instant, the control law with saturation becomes valid and the witness points move along the boundary until the control law no longer attempts to move the witness points outside the patch boundaries. As soon as the unrestricted control law is re-activated, it takes the witness point inside the boundary. However, after a few iterations, this new witness point inside the patch (corresponding to a new initialization) hits another boundary resulting in another controller mode change. Once again the witness points are restricted to the boundary and they leave the boundary when the unrestricted controller becomes feasible. Luckily, this time the new witness point within the patch results in convergence to $\mathbf{P}^*$ without attempting to leave the boundary. (D) illustrates the region of attraction for this controller with saturation at the boundaries. The region of attraction is larger than the case without saturation but is not global.

As demonstrated in (D), the region of attraction of the Newton based iteration with boundary saturation contains the region of attraction of the Newton based iteration (shown in (B)) and extends it significantly. However, the new initialization points recovered by the boundary saturation are the points that hit the boundary of the patch (probably a few times) and finally land in the region of attraction of Newton based iteration. This behavior of the algorithm is not desirable for two reasons. First of all, the convergence of the saturated controller at the boundaries is not guaranteed and the use of this control law may actually result in a sequence of witness points diverging from the solution. Secondly, to arrive at a witness point within the region of attraction of the unrestricted Newton's iteration, the algorithm may go through many mode changes causing the witness points to jump around by hitting the boundaries, and such a behavior is not time efficient and is computationally demanding.

Finally, (E) and (F) demonstrate our proposed controller. Global convergence of our controller is apparent from the figure on the right since all initializations within the surface patch converge to the solution. Note that this is also guaranteed by the theorem given in section III. Moreover, (E) demonstrates that unlike the Newton iteration based controller, the proposed controller converges directly to the solution without bouncing around at the boundaries. This behavior of the controller is always guaranteed by design. No matter if the witness point is on the boundary or within the patch, the control law always results in an update that is closer to $\mathbf{Q}$ than all previous witness points. Consequently, as also demonstrated in Figure 5, the witness points always take a direct path towards $\mathbf{P}^*$ with no unnecessary boundary switches.

C. *Convergence under Relative Motion*

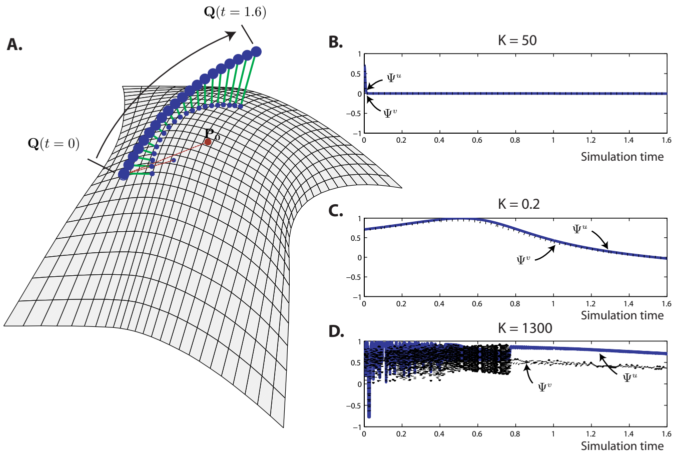


Fig. 7. This four-part figure demonstrates the importance of the choice of the feedback gain for convergence of the algorithm. (A) illustrates the convergence and the tracking behavior of the algorithm when initialized at $\mathbf{P_0}$ and the external point $\mathbf{Q}$ undergoes relative motion with respect to the fixed patch. For this simulation the controller gain is set to $K = 50$. The lines in (A) connect the witness point $\mathbf{P}$ to $\mathbf{Q}$ at every simulation snapshot. Convergence of the initialization error is apparent from the first three snapshots, whereas successful tracking can be observed thereafter. (B,C) and (D) present the evaluation of the normalized projection errors Ψ^u and Ψ^v for three different controller gains. For a low gain of $K = 0.02$, as in (C), the convergence is very slow whereas for a very high gain of $K = 1300$ (given in (D)) undesirable chatter appears due to discretization. For $K = 50$ (shown in (B)), the projection errors asymptotically converge to zero resulting in the tracking behavior presented in (A).

This simulation is performed to demonstrate the importance of the choice of the feedback gain for the convergence of the algorithm. As dictated by theory in section III, there exists upper and lower bounds on the feedback controller gain. The lower bound on the feedback gain exists so that disturbing effects of motion can be suppressed by the controller. This lower bound is primarily governed by the relative motion between the bodies but is also dependent on the shape. The upper bound on the controller gain is due to discretization of the control law and is dictated by the sampling rate and integration method selected.

Figure 7 (A) illustrates the convergence and the tracking behavior of the algorithm under the relative motion between the bodies. The algorithm is initialized at $\mathbf{P_0}$ with the controller gain set to $K = 50$. The external point $\mathbf{Q}$ undergoes relative motion with respect to the fixed patch. At every simulation snapshot, the lines connecting the witness point $\mathbf{P}$ to $\mathbf{Q}$ are shown. In the first three snapshots the controller is compensating for the initialization error and converges to the instantaneous $\mathbf{P}^*$, which thereafter changes location on the patch under the effect of the relative motion. Once the initialization error is compensated for, the controller successfully tracks the motion of $\mathbf{P}^*$ keeping witness points sufficiently close to the solution.

(B,C) and (D) present the normalized projection errors Ψ^u and Ψ^v of the same simulation for three different controller gains. In (B), for a gain of $K = 50$, the projection errors asymptotically converge to zero resulting in the tracking behavior presented in (A). For this particular simulation, it is possible to increase the feedback gain up to $K = 1250$ to achieve even faster convergence rates. At $K = 1250$ the algorithm becomes unstable due to discretization. Behavior of the projection errors for $K = 1300$ is given in (C). Finally, although there is relative motion between the bodies, since the angular velocity of the patch ${}^N\boldsymbol{\omega}^A$ is zero, the lower limit on K is also zero. (D) shows the projection errors for a low gain of $K = 0.02$, in which case the convergence still takes place but is quite slow.

D. *Voronoi Switching*

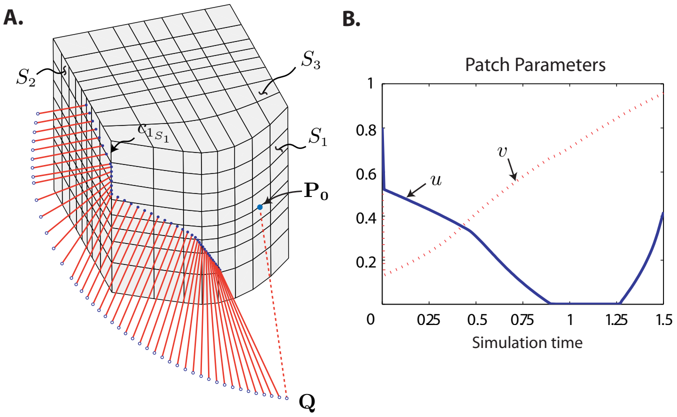


Fig. 8. This two-part figure illustrates the upper level Voronoi based switching algorithm for an object consisting of tiled-together surface patches. (A) demonstrates the convergence and the tracking behavior of the algorithm when initialized at $\mathbf{P_0}$ and the external point $\mathbf{Q}$ undergoes relative motion with respect to the fixed object. The lines in this figure connect the witness point $\mathbf{P}$ to $\mathbf{Q}$ at every simulation snapshot. (B) presents the evaluation of the patch parameters u and v. The Voronoi regions of the object used in the simulation and the automaton associated with the Voronoi switching are given in Figure 3 (C) and (D).

This final simulation demonstrates the upper level Voronoi based switching algorithm for an object consisting of tiled-together surface patches. Figure 8 (A) shows a convex object made of five planar patches and a convex curved patch. The same object and its Voronoi regions are illustrated in Figure 3. The simulation is initialized at point $\mathbf{P_0}$ in the curved patch S_1 and the external point $\mathbf{Q}$ is allowed to trace a pre-specified curved path around the fixed object. The lines in the figure connect the witness point $\mathbf{P}$ to $\mathbf{Q}$ at every simulation snapshot. Figure 8 (B) presents the evaluation of the surface patch parameters u and v as the closest point tracking takes place. Compensation for the initialization error can be observed from the trajectory of the surface parameters at the very beginning of the simulation ($t < 0.01$). At $t = 0.89$ when u decreases to zero, $\mathbf{Q}$ hits the boundary of the Voronoi region of S_1, labelled $\mathcal{V}S_1$, and crosses to the Voronoi region of the bounding curve $c_{1_{S_1}}$. At the same instant, a switching is triggered by the Voronoi based switching algorithm, that sets $c_{1_{S_1}}$ as the active feature. As long as the bounding curve $c_{1_{S_1}}$ stays active, tracking is restricted to this feature; therefore, only the parameter v is updated while u is kept at zero. At $t = 1.26$, when $\mathbf{Q}$ crosses into $\mathcal{V}S_2$, another switching takes place and S_2 becomes the active patch. Tracking continues on S_2 until the simulation is terminated. Note that mode changes triggered by the Voronoi based switching algorithm are due to motion of the external point $\mathbf{Q}$ and are independent of the motion of the witness point $\mathbf{P}$. Illustrations of the features that become active in this simulation and the corresponding Voronoi regions are given in Figure 3 (C). Also, the automaton associated with the Voronoi switching algorithm is shown in Figure 3 (D).

V. Discussion and Conclusions

We have contributed a closest point algorithm with certain attractive properties that follow from its formulation as a dynamic control problem and its solution by synthesis of a feedback controller. These properties include *global* uniform asymptotic convergence, invariance, and the availability of analytical limits of performance.

Global uniform asymptotic stability of our algorithm follows from its derivation from a control Lyapunov function and such stability implies that any initialization within an active convex surface patch converges to the unique solution. The algorithm is in fact a switching algorithm, even at the low level involving only patches (not tiled bodies), and ours is a *common* Lyapunov function that need not switch as the controller does. Switching certain control gain terms to zero is necessary if the witness point moves onto a boundary curve as it navigates toward the closest point. We call this *boundary saturation*. This control gain switching renders the active patch invariant, meaning that the witness point cannot leave the patch during convergence.

These features pertain to the control-based algorithm that handles the closest point on a convex parametric surface patch. A top-level switching algorithm based on Voronoi diagram handles switching among convex surface patches, bounding curves, and vertices making up the convex body (collectively called features). The top-level algorithm significantly extends the properties outlined above, for it effectively increases the basin of attraction of the closest point beyond the active feature to the entire tiled body. We discussed how the Voronoi based switching and stabilized closest point algorithm with boundary saturation can be combined to form a hybrid dynamical system, wherein the membership of the witness point to a particular Voronoi region controls the switching among features on the body. Transitions between surface patches are handled in a manner that decreases minimum distance, as guaranteed by the the V-Clip [13] or the Lin-Canny [12] algorithm. Of course determination of the Voronoi diagram for parametric models can be computationally intensive affair, but this can be undertaken off-line.

Invariance is an important property for that part of the algorithm that handles closest point convergence on patches in order to de-couple the low-level and top-level switching. It is only appropriate to switch patches according to changes in the Voronoi region containing the point on the opposing body, not according to the convergence behavior of the witness point.

In our previous work [1] [2] (represented in Figure 2), we presented a closest point algorithm that possessed only *local* convergence. Whereas the present controller is based on a Lyapunov function defined in terms of the magnitude of the difference vector, the previous controller was based on the squared sum of the error vector projections Ψ^u and Ψ^v. The behavior of the present Lyapunov function is monotonic over the entire patch as the witness point converges. The behavior of the previous Lyapunov function is not monotonic over the entire patch, an example of which can be seen in Figure 5 (C). An estimation of the basin of attraction of the previous controller is a ball around the minimum distance solution where the Jacobian matrix M is positive definite.

As discussed in [2], there exists a close relationship between Newton iteration based methods and our previous control law. Without compensation for the motion of the bodies and under discretization using Euler's method, our previous feedback controller reduces to that published in [6]. Both methods possess only *local* convergence characteristics. The basin of attraction of the Newton-iteration algorithm was demonstrated by simulation in section IV-B to be a complicated and not necessarily connected set.

Analytical limits of performance for the algorithm are discussed in section III and demonstrated in section IV-C. The lower bound on the controller gains (defined in Theorem) is due to the relative motion between the bodies. The upper limit is due to discretization and its analysis is simple. Details on the use of discrete time controller design techniques to analyze convergence rates under various integrators are given in [19] [20] [21]. Once these bounds are in hand, the algorithm can be driven to its limits in speed. Note that the determination of stability-preserving gains K for algorithms based on Newton iteration is a much more complicated affair, since these are discrete and nonlinear methods.

When it comes to comparing the computational efficiency of our algorithm with that of the previously available methods, our argument relies on the simplicity of our feedback control law relative to the Newton iteration and gradient based methods. Our algorithm requires a mere simple controller gain K, whereas the Newton iteration based methods requires calculation and *inversion* of a Jacobian matrix M. Moreover, since derived in continuous time, the computational efficiency of our algorithm can be adjusted with a broader choice of numerical methods to be used for its discretization.

However, our chief motivation for pursuing a closest point algorithm that operates *directly* on parametric surfaces rather than on their tessellations is for benefits expected to accrue as its use is extended to non-convex, intersecting, and deformable bodies. The global property in particular will become important not just during initialization, but when a witness point undergoes discontinuous jumps between patches on a tiled body that is penetrated by a point or other body. We aim for an algorithm guaranteed not to break down except as quantifiable limits on performance are exceeded.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the National Science Foundation under award number IIS:PECASE #0093290, and some very fruitful discussions with Jessy Grizzle.

REFERENCES

[1] V. Patoglu and R. B. Gillespie, "Extremal distance maintenance for parametric curves and surfaces," in *Proc. 2002 IEEE International Conference on Robotics and Automation*, pp. 2817–2823, 2002.

[2] V. Patoglu and R. B. Gillespie, "Haptic rendering of parametric surfaces using a feedback stabilized extremal distance tracking algorithm," in *Proc. IEEE International Conference on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, vol. 3, pp. 391 – 399, 2004.

[3] R. Ramamurthy and R. T. Farouki, "Voronoi diagram and medial axis algorithm for planar domains with curved boundaries -I. theoretical foundations," *Journal of Computational and Applied Mathematics*, vol. 102, pp. 119–141, 1999.

[4] T. V. Thompson II, D. E. Johnson, and E. Cohen, "Direct haptic rendering of sculptured models," in *Proceedings Symposium on Interactive 3D Graphics*, pp. 167–176, 1997.

[5] T. V. Thompson II, D. D. Nelson, E. Cohen, and J. Hollerbach, "Maneuverable NURBS models within a haptic virtual environment," in *Proceedings of ASME International Mechanical Engineering Congress and Exposition*, vol. 61, pp. 37–44, 1997.

[6] D. D. Nelson, D. E. Johnson, and E. Cohen, "Haptic rendering of surface-to-surface sculpted model interaction," in *Proceedings ASME Dynamic Systems and Control Division*, vol. 67, pp. 101–108, 1999.

[7] Z. Zou and J. Xiao, "Tracking minimum distances between curved objects with parametric surfaces in real time," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2692 – 2698, 2003.

[8] P. Bhattacharya and A. Rosenfeld, "Convexity properties of space curves," *Pattern Recogn. Lett.*, vol. 24, no. 15, pp. 2509–2517, 2003.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*. http://www.stanford.edu/ boyd/cvxbook.html, 2003.

[10] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Springer, 2000.

[11] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computer Methods in Applied Mechanics and Engineering I*, pp. 1–16, 1972.

[12] M. C. Lin and J. F. Canny, "A fast algorithm for incremental distance calculation," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1008–1014, 1991.

[13] B. Mirtich, "V-Clip: Fast and robust polyhedral collision detection," *ACM Transactions on Graphics*, vol. 17, no. 3, pp. 177–208, 1998.

[14] H. Alt and O. Schwarzkopf, "The Voronoi diagram of curved objects," in *Proceedings of the Eleventh Annual Symposium on Computational Geometry*, pp. 89–97, 1995.

[15] S. M. Hong, *New algorithms for detecting collison between moving objects*. PhD thesis, The University of Michigan, 1989.

[16] D. Liberzon, *Switching in Systems and Control*. Birkhauser, 2003.

[17] S. T. Lin and J. N. Huang, "Stabilization of Baumgarte's method using the Runge-Kutta approach," *Journal of Mechanical Design*, vol. 124, pp. 633–641, 2002.

[18] J. C. Chiou and S. D. Wu, "Constraint violation stabilization using input-output feedback linearization in multibody dynamic analysis," *Journal of Guidance, Control and Dynamics*, vol. 21, no. 2, pp. 222–228, 1998.

[19] E. V. Solodovnik, G. J. Cokkinides, and A. P. S. Meliopoulos, "On stability of implicit numerical methods in nonlinear dynamical systems simulation," in *Proceedings of the Thirtieth Southeastern Symposium on System Theory*, vol. 3, pp. 27–31, 1998.

[20] S. T. Lin and J. N. Huang, "Parameter selection for Baumgarte's constraint stabilization method using predictor-corrector approach," *AIAA Journal of Guidance, Control and Dynamics*, vol. 23, no. 3, pp. 566–570, 2000.

[21] S. T. Lin and M. C. Hong, "Stabilization method for the numerical integration of controlled multibody mechanical systems: A hybrid integration approach," *JSME International Journal Series*, vol. 44, no. 1, pp. 79–88, 2001.