Marquette University
e-Publications@Marquette

Master's Theses (2009 -)   Dissertations, Theses, and Professional Projects

Development of a Velocity Metric for Rigid-Body Planar Motion

Luis Ernesto Criales Escobar
Marquette University

DEVELOPMENT OF A VELOCITY METRIC FOR RIGID-BODY PLANAR MOTION

by

Luis E. Criales Escobar

A Thesis submitted to the Faculty of the Graduate School,

Marquette University,

in Partial Fulfillment of the Requirements for

the Degree of Master of Science

Milwaukee, Wisconsin

December 2009

ABSTRACT

DEVELOPMENT OF A VELOCITY METRIC FOR
RIGID-BODY PLANAR MOTION

Luis E. Criales Escobar

Marquette University

Two motions of motion quality have been developed for planar motion. They are the "best motion measure" and the "velocity metric". The "best motion measure identifies the best motion for a given displacement. The "velocity metric" quantifies the discrepancy between two planar motions for the same rigid body.

The best motion measure compares the motion of each particle on the body to an "ideal", but usually unobtainable, motion. This ideal motion moves each particle from its current position to its desired position on a straight-line path. Although the ideal motion is not a valid rigid body motion, this does not preclude its use as a reference standard in evaluating valid rigid body motions. It is shown that the best motion measure can be reduced to the product of two components: the average distance from a point in the plane to the body, and a term based on configuration parameters.

The optimal instantaneous planar motion for general rigid bodies in translation and rotation is characterized. This optimal motion is defined as the geodesic motion of a frame located at the geometric center of the body. In other words, the geometric center of the body will move in a straight-line path to its desired position, while the body rotates about the axis perpendicular to the plane. The optimal angular velocity is a function of the discrepancy between the current configuration and the desired configuration.

A velocity measure used to evaluate how closely one motion is to another is also defined. This second measure is shown to meet the mathematical requirements of a metric. For a single particle, the velocity metric is the norm of the difference in translational velocity. It is shown that for most cases, this metric can be characterized as the average distance from a point in space to the body. This point is the instantaneous center of the motion associated with the discrepancy between the two motions.

Because the measures described above depend on body geometry, a method to analytically evaluate any polygonal body based on polygonal decomposition into simpler bodies is provided. Results for example planar positioning problems are presented.

ACKNOWLEDGMENTS

Luis E. Criales Escobar

The development and completion of this thesis would not have been possible without the support of many individuals. First, I would like to thank my adviser, Dr. Joe Schimmels, for all his guidance and support throughout the past two years. His dedication and enthusiasm regarding research and education are unparalleled. I would also like to thank Dr. Phil Voglewede and Dr. Shuguang Huang for teaching me about rigid-body dynamics and robotics, which have been essential for the completion of this work.

I thank the Lord for giving the opportunity to experience life in a completely different way as a Marquette student for the past 2 years. It has been an adventure with both sweet and sour moments, but I like to believe that they have all contributed in making me a better person. I have met a lot of people and developed friendships that I am sure will last for the rest of my life. I would like to thank all my friends who have allowed me to keep all these things in perspective. I have always felt like they are next to me giving me their best words of advice.

Last but not least, I would like to dedicate this work to my parents, who have always given me the best advice for any and all situations. I cannot begin to imagine how hard it must be to send their first-born 3000 miles away from home and will be forever grateful for the trust and faith they have always placed in me.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

θ -   Angular discrepancy between two pose configurations of a rigid body.

q -   Current location of particle q in the plane.

q' -   Target location of particle q in the plane.

$\mathbf{r}_{ab}$ -   Location of particle b relative to a.

ω -   Angular velocity of a body.

$R(\theta)$ -   Rotation matrix for a given discrepancy between frames.

O -   Center of rotation for a given configuration problem.

o -   Instantaneous center of rotation for a given motion.

$\mathbf{I}$ -   Identity matrix.

$\mathbf{v}_q$ -   Translational velocity vector of particle q.

$\Delta\mathbf{v}_q$ -   Difference in translational velocity vector at q.

$\mathbf{v}_{i_q}$ -   Ideal translational velocity vector for particle q.

$\mathbf{v}_{a_{Cq}}$ -   Translational velocity vector for particle q due to the geodesic motion of the frame located at C.

v_{d_q} -   Best motion measure for particle q.

v_D -   Best motion measure for a body.

τ -   Time to complete motion.

$[\omega_\times]$ -   Skew-symmetric matrix representing $\omega\hat{\mathbf{k}}\times$.

k -   Scalar indicating relative magnitude of rotation with respect to translation.

$\bar{r}$ -   Average distance from a location in the plane to a body.

$\mathbf{t}$ -   Planar twist describing translational and angular velocity of a motion.

v_{m_q} -   Velocity metric for particle q.

v_M -   Velocity metric for a body.

$\mathbf{P}_{Cq}$ -   Transformation matrix from twist at C to translational velocity at q.

$d(e_1, e_2)$ -   Measure of "Distance" between two elements of a set.

$\mathbf{H}_{CC'}$ -   Homogeneous transformation from frame C to C'.

$||\mathbf{x}||$ - L^2 norm of vector $\mathbf{x}$.

$|x|$ -   Absolute value of scalar x.

1. INTRODUCTION

The ability to achieve accurate positioning is a vital characteristic of a robot. It is currently difficult to characterize the accuracy of different configurations due to the need to consider rotational and translational error simultaneously. A measurable quality is often needed to properly evaluate the distance between an obtained rigid body position and its target configuration. This problem arises frequently in general mechanism design, robot calibration, and motion planning.

The ability to adequately characterize the "distance" between configurations is not the only relevant problem in these subject areas. There is also the need to adequately compare all motions that will produce a change in this distance. In other words, there is a need to identify the best available motion that will take the body as close as possible to its desired configuration. Therefore, it makes intuitive sense to develop a metric which characterizes the ability of a velocity to be error-reducing. A fairly significant amount of work has been directed towards the analysis of distance metrics, but very little emphasis has been placed on metrics that characterize the ability of a velocity to reduce the distance between two rigid body configurations.

It is important to understand why distance metrics are used and what some of the shortcomings are that prevent or limit their use in practical applications. The same is true of velocity metrics. This chapter reviews previously developed distance and velocity metrics and provides an outline for this thesis.

1.1 Distance Metrics

The need for distance metric functions for rigid body motion has been recognized for more than 100 years [1]. A metric space must comply with three mathematical requirements: positive definiteness, symmetry, and the triangle inequality. A formal definition of the metric space and the three conditions that must be satisfied is given in Chapter 4. However, it is important to note that these three conditions are not overly constraining. A consequence of this is that an indefinite number of distance metrics may be defined, but it is important to select one that is physically intuitive. A physically meaningful metric is important since it allows practical application. To comply with physical consistency, kinematic distance metrics are preferred to be *frame bi-invariant* and *scale invariant* [2]. If the metric results do not vary depending on where the frame of reference used to describe the body is located in space, or where the body-based frame of reference is located on the body itself, then the metric is said to be *frame bi-invariant*. In other words, a metric which is frame bi-invariant is both world-frame invariant and body-frame invariant. Furthermore, if the results obtained by applying the metric to a configuration problem are independent of the units selected to represent length, then the metric is said to be *scale invariant*.

Let us illustrate a physically inconsistent metric using a simple example. Consider a metric function D_1 defined as $D_1 = d + \theta$, where d is the translational distance for a single frame between both configurations and θ is the magnitude of the orientational error between configurations. Such a metric is not scale-invariant because d and θ have different units, implying that the results will be influenced by the selection of units used to describe length and orientation. Furthermore, this metric is not frame bi-invariant since the value of d will change depending on

where the body frame is located.

Another example of a metric is $D_2 = \sqrt{d^2 + (l\theta)^2}$. In this case, a scaling factor l is introduced to account for the unit discrepancy between translation and rotation. However, the choice of l is arbitrary. Therefore, this metric is not scale-invariant.

Metrics that are independent of a body's specific geometry considered the bodies to be of infinite size. Most of the previous work in the development of kinematic distance metrics was done without much regard for practical applications. Therefore, to consider bodies of infinite size was not a limitation at the time since metrics should be independent of body shape and size. Regardless, it is important to understand how kinematic distance metrics have evolved in the last century. To more adequately grasp this evolution, distance metrics can be grouped into three categories based on the characteristics of the approach used by the mathematicians and kinematicians who developed them. These three categories are the following: projective mapping metrics, Lie-group algebra based metrics, and geometry-specific Euclidean metrics.

1.1.1   Projective Mapping Metrics

Several distance metrics based on projective mappings to obtain dimensionless coordinates for position have been developed. One of the first known attempts to address the need for a positioning metric was made by Stéphanos [1] in 1883. Stéphanos associated an elliptic metric with the special orthogonal group, $SO(3)$. $SO(3)$ is also known as a the rotation group, and is best defined as the collection of all spatial rigid body motions that result in an arbitrary point being fixed in space. Based on Stéphanos' work, several other researchers aimed to map

displacements into special projective spaces. In particular, distance metrics have been obtained by mapping displacements from the rotational ($SO(3)$), planar ($P(3)$), and Euclidean ($E(3)$) groups into real projective spaces.

Blaschke [3] showed that the set of all motions that move a body's particles in parallel planes could be mapped as a three-dimensional projective space. Furthermore, he associated this space with a quasi-elliptic metric. The group of motions that move all of a body's particles in parallel planes is also known as $P(3)$, the Planar group. As a side note, Blaschke himself coined the term "kinematic mapping" to describe his approach. Around the same time, Study [4] tried to map arbitrary spatial displacements into a real projective space. Study referred to the set of all spatial displacements as $E(3)$, the Euclidean group, and the real projective space into which he mapped the displacements is known as the Soma space. Around 70 years later, Ravani and Roth [5] reconfigured Study's real projective space into a dual projective space. They also identified an elliptic metric that corresponds to this dual projective space, whereas is not clear whether Study associated a metric to the Soma space. More recently, Eberharter and Ravani [6] made use of Study's quadric to define a new metric for rigid body displacements based on an optimized local mapping of the quadric. Eberharter and Ravani used stereographic projections to optimize the local mappings. A stereographic projection is a mapping that projects a sphere into a hyperplane for every point on the sphere except the projection point. The optimization of the local mappings is done with respect to a given set of key positions, manually selected, as opposed to defining and developing a global metric. By using this procedure [6], curve-design algorithms can be used for motion design, with emphasis in smooth functions.

1.1.2   Lie-Group Algebra Based Metrics

Other researchers [7] have addressed the development of rigid body distance metrics using Lie groups. It has been shown that the special Euclidian group, $SE(3)$, is a Lie group. The special Euclidean group is the set of all matrices which correspond to a homogenous transformation describing the configuration of a body (position and orientation). In other words, any given spatial displacement is an element of $SE(3)$. Kuffner [8] tackled the relative weighting of translation and rotation components of $SE(3)$. He showed that unit quaternions [9] were preferable to represent the rotational component of $SE(3)$ elements. Larochelle and Murray [10] presented two metrics based on two techniques for approximating elements of $SE(3)$ with elements of $SO(4)$. In fact, they generalized their approach so that elements of $SE(n)$ can be approximated to elements of $SO(n+1)$. The proposed metrics are based on singular value and polar decompositions (SVD and PD respectively). Both metrics are projection based, using hyperdimensional rotations to approximate displacements. However, their metrics are not scale-invariant. A characteristic length was chosen to normalize the translational/rotational components.

1.1.3   Geometry-Specific Euclidean Metrics

Park showed that there is no *general* frame bi-invariant metric for spatial rigid body motion [2]. He also identified the mathematical requirements for world-frame invariance and the mathematical conditions for body-frame invariance as described previously. More recently, Lin and Burdick [11] noted that body-frame invariance is an overly conservative constraint that does not accurately reflect the true nature of a finite rigid body for physical phenomena such as kinetic energy. By allowing a rigid-body frame transformation, it is possible to obtain a physically meaningful form of frame invariance. In other words, the same rigid

body displacement is being considered, albeit described separately in different body-frame coordinates [11]. Several kinematicians have proposed distance metrics based on a predetermined finite body geometry. The application of such metrics is limited to problems in which the *same* rigid body is evaluated for multiple configurations.

Kazerounian and Rastegar [12] introduced a measure based on the Euclidean distance for all particles on the body, while Martinez and Duffy [13] proved that the measure satisfied the mathematical requirements of a metric. This metric for a specified body $\mathcal{B}$ is

$$D_{\text{avg}} = \frac{1}{V} \int_{q \in \mathcal{B}} d_q \, dV, \tag{1.1}$$

where V is the body volume, and $d_q = |\mathbf{p}_{qq'}| = \sqrt{\delta_x^2 + \delta_y^2 + \delta_z^2}$ is the distance between two configurations for a single particle q on the body. It is clear that the metric shown in (1.1) is frame-invariant, since no particular fixed body reference frame is chosen and all body particles are considered. However, the exceptional computational capacity requirement associated with performing this calculation for the entire body volume is seen as a deterrent in its possible application. A less computatationally intensive alternative suggested by Martinez and Duffy is given by the metric based on calculating the maximum distance for any particle on the body from proper positioning:

$$D_{\text{max}} = \max_{q \in \mathcal{B}} d_q. \tag{1.2}$$

Another variant proposed by Chirikjian and Zhou [14] is based on the squared distance between the two configurations for each particle. By squaring the distance, the square root is eliminated from the calculation in (1.1) and the

integral is simplified significantly. However, this metric has its own limitations. The first limitation is that it is not physically consistent with what is expected of a distance metric. The units are length squared as opposed to length. The second limitation has to do with the way in which different particles are weighted by the metric. Particles which are closer to their proper positioning contribute less to the metric than particles that are father away. This is a consequence of squaring the values. Therefore, not all particles on the body are weighted equally.

1.2   Velocity Metrics

The main difference between developing a rigid body distance metric and a rigid body velocity metric lies on the fact that an optimal or target velocity has not yet been defined. Some previous work has been done trying to characterize the best motion. One such area of prior work is trajectory planning [15]. However, this work focuses on motion dynamics and the forces required to generate a motion, as opposed to establishing the best motion from a geometric perspective. Another area addresses motion planning [16], although this work is directed toward finding a path from one position to another when obstacle avoidance is a vital issue. Due to the lack of an accepted distance metric, however, little work has been done in establishing a measure of a good instantaneous motion when many motions are available.

1.2.1   Riemannian Metrics

In differential geometry and topology, a manifold is a mathematical space that is modeled based on the Euclidean space and can be considered as an n-th dimensional expansion of the Euclidean space. A manifold can be described as a collection of "charts", also referred to as an "atlas". The terminology is derived from the fact that the idea of a manifold is similar to being able to describe the

world using a series of maps, that when placed together provide a complete representation of the Earth. One particular example of a manifold is a differentiable manifold, a manifold similar enough to the Euclidean space that allows the user to use calculus [17]. Furthermore, it is possible to attach a tangent space to every point of a differentiable manifold. A tangent space is a real vector space which contains all vectors that pass through that particular point. A tangent space is a generalization of the concept of a bound vector in the Euclidean space. In this way, a notion of "distance" can be established within the manifold. A common measure of this distance is the inner product. If the inner product of every tangent space of a differentiable manifold is smooth across all "charts", the manifold is said to be Riemannian [18]. Furthermore, the inner product is known as a Riemannian metric (by satisfying positive definiteness, symmetry, and the triangle inequality).

In a similar manner to the way in which Lie-group theory has been used to develop distance metrics, several kinematicians have used this theory in their velocity-based research. By definition, Lie groups are differentiable manifolds so Riemannian metrics are of special interest as well. Zefran, Kumar and Croke [7] applied a functional based on velocity and its derivatives that measures the smoothness of a trajectory, hoping to find a trajectory that would minimize this functional. In fact, their proposed measure for trajectory smoothness is a Riemannian metric on $SE(3)$. However, this metric depends on the choice of the body-fixed reference frame, making it not body-frame invariant.

Other previous work with Lie groups has considered the distance along the path taken by a frame attached to a body, which can be expressed as

$$L = \int_a^b < \dot{\mathbf{X}}, \dot{\mathbf{X}} >^{1/2} dt, \qquad (1.3)$$

where $\dot{\mathbf{X}}$ is the instantaneous motion of the frame. If the world-frame invariant metric described by Park is chosen [2], $< \dot{\mathbf{X}}, \dot{\mathbf{X}} > = c^2 \, \omega^T \omega + \, v^T v$, for which ω is the angular velocity, v is the translational velocity, and c is a selected scalar with units of length. A "characteristic length" is chosen for c since there is no general natural length scale. The choice for c is based on engineering judgement and practical application. This form of velocity metric has been used to evaluate motions relative to a desired motion. However, the results depend on both the choice of characteristic length c and the choice of body frame location.

1.2.2   Geodesic Motion and Application

From a kinematic perspective, an important quality of a good motion is that the path taken corresponds to the minimum distance [2, 7]. The minimum distance curve is referred to as a *geodesic*. For planar motion, the motion associated with a geodesic curve given by (1.3) corresponds to a straight line path of the body frame from its current location to a target location, with rotation about the axis perpendicular to the plane of motion.

Recall that means of assessing the quality of a relative configuration of a finite rigid body were presented in Section 1.1.3. The metric given by (1.1) involved calculating the average distance of all body particles from their intended locations, while (1.2) involved only determining the maximum distance of all body particles from their intended locations. If the maximally displaced particle $D_{\max}$ is used as a reference to evaluate motion quality, many possible motions are viewed to be equally acceptable. As a consequence, a binary measure is obtained, given by

$$\mathbf{v}_q^T \mathbf{r}_{qq'} < 0 \tag{1.4}$$

where q is the particle of maximum distance, v_q is the translational velocity at q, and $\mathbf{r}_{qq'}$ is the current location of q relative to its desired location. Because the

distance metric applies to a single particle, half of all motions are distance reducing, while the other half are distance increasing. Even though this velocity measure is both scale and frame invariant, it ensures only that the particle of maximum displacement is instantaneously reduced by the motion.

Previous work has also addressed further restriction of the set of motions that reduce rigid body distance, by looking at different particles on the body [19, 20, 21].
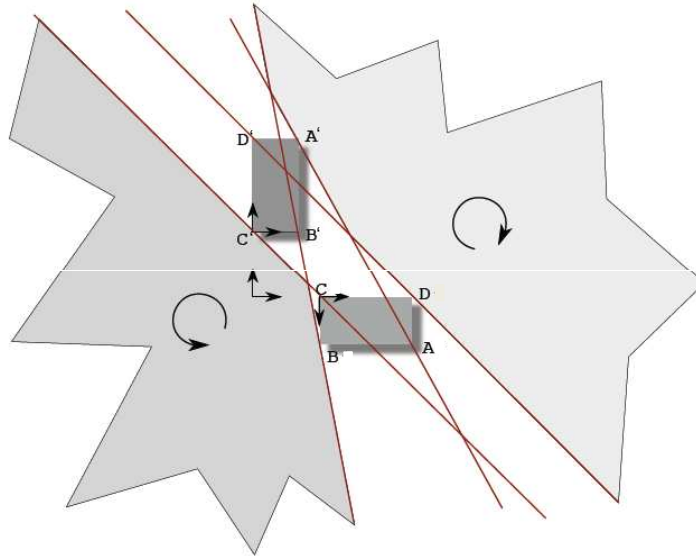
Figure 1.1: Rotation Centers Yielding Error Reduction.

It has been shown that the particle of maximal displacement is always one of the body vertices for polyhedral bodies. Therefore, applying the distance reduction conditions to the vertices ensures that the metric is always reduced. For the body of Fig. 1.1, these conditions are expressed as:

$\mathbf{v}_i^T \mathbf{r}_{ii'} < 0 \quad \forall i \in \{A, B, C, D\}$. The set of planar motions (the locus of all rotation centers) that reduce the distance from proper positioning for all of the vertices can

be seen in Fig. 1.1. However, a set of binary measures of motion quality does not comply with the mathematical requirements of a metric. Therefore, a means of identifying the best velocity and a means of evaluating the quality of motions relative to a best velocity are unavailable. This thesis will focus on finding appropriate answers to both of these issues.

1.3 Overview

This thesis will provide two important contributions. The first result consists of identifying the best available rigid-body motion for a given planar displacement by applying a measure based on velocity for all particles on a rigid body. This measure will be known as the *best motion measure.* The second contribution is a velocity metric that can be used to evaluate any two rigid-body planar motions by again evaluating the particle velocity of all particles on the rigid body. This metric will be known as the *velocity metric.*

To find the best available rigid-body motion, a velocity measure is defined based on reducing the average Euclidean distance for all particles that constitute the rigid body for planar displacements. In Chapter 2, the strategy used to calculate the difference between a desired motion and an available rigid body motion is identified. The procedure to calculate the measure is presented and analytical solutions for simple geometries are presented. Optimal instantaneous velocities for rigid bodies in planar motion are characterized based on the best motion measure. Examples for simple geometries are also provided for better understanding in this chapter.

In Chapter 3, the velocity metric for planar rigid-body motion is developed based on the discrepancy between two motions. First, the strategy used to

evaluate the motion discrepancy is described. Then, the procedure to calculate the discrepancy between the two motions is presented. It is shown that the measure is intuitive, physically consistent (scale and frame invariant), and meets the mathematical requirements of a metric. *Both* the best motion measure and the velocity metric depend on body geometry.

Analytical solutions for simple geometries are described in Chapter 2. Chapter 4 presents an algorithm used to evaluate any polygonal body with polygonal or circular holes. This algorithm consists of: 1) polygonal body decomposition into right-triangles, and 2) the summation of measure values for each element of the decomposition to calculate the measure for the whole body. An example of the application of this algorithm is presented as well.

Finally, results are summarized and plans for future work in this area are described in Chapter 5.

2. BEST MOTION MEASURE FOR RIGID-BODY PLANAR MOTION

In this chapter, a rigid body velocity measure is developed that is based on the distance metric that considers the average Euclidean distance for all particles on the body. There are two main reasons to develop this velocity measure. The first is to better understand the nature of a velocity metric developed based on average particle distance, and the second is to determine what constitutes the best motion. The idea behind this measure is to be able to evaluate any motion relative to an "ideal" velocity. For this measure, the "ideal" velocity does not correspond to a valid rigid-body motion. In fact, the "ideal" velocity is associated with the displacement metric: the straight-line distance between two locations of the same particle. Recall that the final goal is to develop a velocity metric based solely on valid rigid body motions.

The first step in developing the velocity measure is to accurately describe possible body configurations without loss of generality. Then, the "ideal" velocity for each particle is identified and mathematically characterized. Once this "ideal" velocity has been determined, the velocity measure is defined as the discrepancy between this "ideal" motion and an available rigid body motion.

It is shown that the geometry of the body influences the best motion measure. The body geometry-dependent portion of the best motion measure is characterized by the average distance of the body particles from a frame executing a geodesic motion. A graphical representation of the variation of best motion measure with respect to the location of the frame undergoing the geodesic motion

is provided for three simple geometries: a rectangle, a right-triangle, and a circle. The best available rigid-body planar motion is based on the calculation of the optimal geodesic frame location and the optimal angular velocity of the body.

2.1 Characterization of Body Configurations

For the measure based on the Euclidean distance between configurations for all particles that constitute the rigid body, given by (1.1), the selection of coordinate frames is arbitrary. This is possible since the metric given by (1.1) is scale- and frame-invariant. However, the way in which the reference frames are selected will greatly influence the complexity of the calculations that must be performed, so the choice of reference frames is relevant.
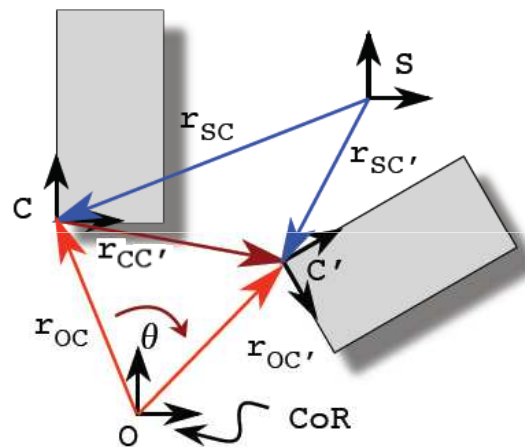
Figure 2.1: World-Frame Located at the Center of Rotation (CoR).

Consider the simple example illustrated in Figure 2.1. The figure illustrates a body in its current configuration and its target configuration. Consider a body-frame at particle C such that the body-frame's axes are aligned with the

body edges to simplify the calculations. C is the current location of the body

frame and C' is the desired frame location. The displacement of the body from its

target configuration can be described with a homogeneous transformation for the

frame at C as follows:

$$\mathbf{H}_{CC'} = \begin{bmatrix} \mathbf{R}(\theta) & \mathbf{r}_{CC'} \\ 0 & 1 \end{bmatrix}, \tag{2.1}$$

where $\mathbf{R}(\theta) \in \mathbb{R}^2$ is the rotation matrix that describes the rotational displacement

of the body from its target configuration for the planar case, and $\mathbf{r}_{CC'}$ is the

position vector indication the target location of body frame C' relative to the

current location of the frame at C. In fact, $\mathbf{r}_{CC'}$ can be calculated as:

$$\mathbf{r}_{CC'} = \mathbf{r}_{SC'} - \mathbf{r}_{SC}, \tag{2.2}$$

where $\mathbf{r}_{SC}$ is the position of particle C relative to an arbitrary world-frame S. As

mentioned previously, aligning the world-frame with the body-reference frame will

greatly simplify all calculations. In fact, the calculations are greatly simplified if

the world reference frame is located at O, the center of rotation (CoR) for the

given pose discrepancy. It is important to note that the CoR is located at infinity

if there is no discrepancy in orientation between the body's current and desired

configurations. However, for pure translation, the best motion is simply the one

that takes the body in the straight-line path to their desired configuration. This

problem is trivial and will be addressed in more detail later. For the sake of

consistency, it is hereby assumed that $\theta \neq 0$ and that O is located at some finite

distance from C somewhere in the plane.

Using O as the location of the world frame, (2.2) is reduced to:

$$\begin{aligned}
\mathbf{r}_{CC'} &= \mathbf{r}_{OC'} - \mathbf{r}_{OC} \\
&= (\mathbf{R}(\theta) - \mathbf{I})\mathbf{r}_{OC},
\end{aligned} \tag{2.3}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{r}_{OC}$ is the position of particle C with respect to the world frame located at O.

The location $\mathbf{r}_{CO}$ of the CoR relative to frame C for the displacement $\mathbf{H}_{CC'}$ needed to determine $\mathbf{r}_{OC}$ is obtained by inverting the relationship between $\mathbf{r}_{CC'}$ and $\mathbf{r}_{OC}$ contained in (2.3). The location is

$$\begin{aligned}
\mathbf{r}_{CO} &= -\mathbf{r}_{OC} \\
&= -(\mathbf{R}(\theta) - \mathbf{I})^{-1}\mathbf{r}_{CC'} \\
&= -\frac{1}{2 - 2cos\theta}(\mathbf{R}(\theta)^T - \mathbf{I})\mathbf{r}_{CC'}.
\end{aligned} \tag{2.4}$$

2.2 Definition of the "Ideal" Velocity for a Particle

Now that a suitable world reference frame has been identified, it is possible to define and then describe the "ideal" velocity for a particle on the body. Recall that a path, as given by (1.3), is best if it is the minimum distance path (or geodesic) for the specified body frame. The *geodesic* for a given frame corresponds to a straight line path of the body frame from one location to the other at constant translational velocity, with concurrent rotation at constant angular velocity about the Euler axis passing through the frame's origin. Figure 2.2a. illustrates the motion that would be induced on particle A if the body were to
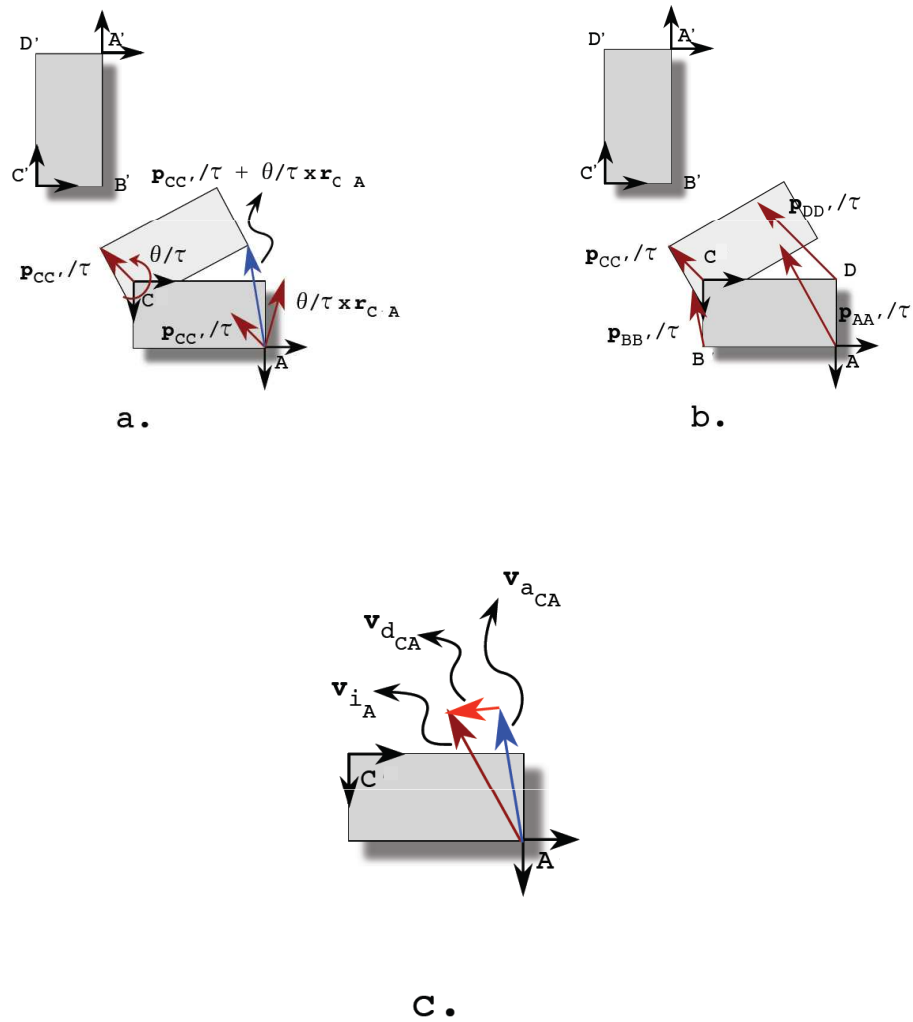
Figure 2.2: Particle Velocities for Geodesic Motions.

move along the geodesic associated with the frame located at particle C after a finite period of time.

The ideal motion for each particle of a rigid body will be determined by the ideal motion of each particle independently. In other words, the "ideal" velocity for a set of particles that compose a rigid body is determined by analyzing the shortest path toward proper positioning for each particle. The "ideal" velocities for particles A, B, C, and D are illustrated in Fig. 2.2b. By assuming that the

motion for each particle is achieved in the same time interval, the best motion for each particle will then have a magnitude and a direction. The time interval can be chosen arbitrarily, since only the *relative* magnitudes are important. Therefore, let this finite time interval be τ.

It is easily observed that the "ideal" motion described above is only achievable simultaneously for all particles when the configuration problem is one of pure translation ($\theta = 0$). In general, the "ideal" motion is not a valid rigid body motion, since particles of a rigid body are not allowed to move independently. However, the fact that the best motion is generally not an available rigid body motion does not preclude its use as a convenient reference standard.

The actual velocity of a specific particle in a valid rigid body motion can be compared to the "ideal" motion for that same particle. For example, Figure 2.2c illustrates the velocity of particle A for the frame C geodesic, $\mathbf{v}_{a_{CA}}$, relative to the ideal particle motion of A, $\mathbf{v}_{i_A}$. The discrepancy between the two velocities at A is:

$$\mathbf{v}_{d_{CA}} = \mathbf{v}_{a_{CA}} - \mathbf{v}_{i_A} \tag{2.5}$$

The velocity measure is determined by evaluating the average magnitude of this discrepancy for all particles that constitute the rigid body, following the same idea used for the distance metric in (1.1). The magnitude of the discrepancy for particle A is given by the Euclidean norm:

$$v_{d_{CA}} = \left(\mathbf{v}_{d_{CA}}^T \mathbf{v}_{d_{CA}}\right)^{1/2} \tag{2.6}$$

Then, for all particles q in the body:

$$v_{D_q C} = \int_{q \in \mathcal{B}} (\mathbf{v}_{d_{C_q}}^T \mathbf{v}_{d_{C_q}})^{1/2} \, dq \qquad (2.7)$$

The measure given by (2.7) is calculated in the next section.

2.3 Calculation of the Best Motion Measure

The two elements of (2.5) must be expanded separately in order to evaluate the measure. The ideal velocity of particle A is given by the distance between A and A', divided by the time interval τ. Also, (2.3) can be used to simplify the distance between A and A':

$$
\begin{aligned}
\mathbf{v}_{i_A} &= \mathbf{p}_{AA'}/\tau = (\mathbf{r}_{OA'} - \mathbf{r}_{OA})/\tau \\
&= (\mathbf{R}(\theta) - \mathbf{I}) \; \mathbf{r}_{OA}/\tau.
\end{aligned}
\qquad (2.8)
$$

The velocity of A for a valid rigid body motion can be described using the definition of geodesic for a specific frame. For the planar case, any motion of the body can be described by the geodesic of a specific frame. This frame does not necessarily have to be on the body. The proof that any planar motion can be described by the geodesic of a properly selected frame can be found in Appendix A. Using this concept, the velocity of A for the geodesic for frame C is given by

$$
\begin{aligned}
\mathbf{v}_{a_{CA}} &= \mathbf{v}_{i_C} + \mathbf{v}_{CA} \\
&= \mathbf{v}_{i_C} + \theta/\tau \hat{\mathbf{k}} \times \mathbf{r}_{CA} = \mathbf{v}_{i_C} + \omega \hat{\mathbf{k}} \times \mathbf{r}_{CA} \\
&= (\mathbf{R}(\theta) - \mathbf{I}) \mathbf{r}_{OC}/\tau + [\omega_\times]\mathbf{r}_{CA} \,,
\end{aligned}
\qquad (2.9)
$$

where $\omega = \theta/\tau$, the angular velocity of the body, and $[\omega_\times]$ is the 2×2 skew-symmetric cross-product matrix associated with ω.

The velocity discrepancy at particle A between the ideal velocity and the available velocity for the geodesic of frame C can be rewritten using (2.8) and (2.9).

$$
\begin{aligned}
\mathbf{v}_{d_{CA}} &= \mathbf{v}_{i_A} - \mathbf{v}_{a_{CA}} \\
&= (\mathbf{R}(\theta) - \mathbf{I}) \ \mathbf{r}_{OA}/\tau - [(\mathbf{R}(\theta) - \mathbf{I}) \ \mathbf{r}_{OC}/\tau + [\omega_\times]\mathbf{r}_{CA}] \\
&= (\mathbf{R}(\theta) - \mathbf{I}) \ \mathbf{r}_{CA}/\tau - [\omega_\times]\mathbf{r}_{CA} \\
&= (\mathbf{D} - \mathbf{W}) \ \mathbf{r}_{CA}.
\end{aligned}
\tag{2.10}
$$

where $\mathbf{D} = (\mathbf{R}(\theta) - \mathbf{I})/\tau$ and $\mathbf{W} = [\omega_\times]$. The magnitude of the velocity discrepancy $v_{D_{CA}}$ for particle A, from (2.6), is

$$
\begin{aligned}
v_{d_{CA}} &= [\mathbf{v}_{d_{CA}}^T \ \mathbf{v}_{d_{CA}}]^{1/2} \\
&= [((\mathbf{D} - \mathbf{W}) \ \mathbf{r}_{CA})^T (\mathbf{D} - \mathbf{W}) \ \mathbf{r}_{CA}]^{1/2} \\
&= [\mathbf{r}_{CA}^T (\mathbf{D} - \mathbf{W})^T (\mathbf{D} - \mathbf{W}) \mathbf{r}_{CA}]^{1/2} \\
&= \frac{1}{\tau}\sqrt{\theta^2 + 2(1 - \cos\theta - \theta\sin\theta)} \sqrt{x_{CA}^2 + y_{CA}^2}.
\end{aligned}
\tag{2.11}
$$

The average velocity discrepancy for all particles on the body $q \in \mathcal{B}$ for the geodesic of any frame located at z can be similarly expressed as:

$$
\begin{aligned}
v_D &= \frac{\frac{1}{\tau}\sqrt{\theta^2 + 2(1 - \cos\theta - \theta\sin\theta)}}{A} \int_{q \in \mathcal{B}} \sqrt{x_{zq}^2 + y_{zq}^2} \ dA \\
&= \frac{\frac{1}{\tau}\sqrt{\theta^2 + 2(1 - \cos\theta - \theta\sin\theta)}}{A} \int_{q \in \mathcal{B}} \sqrt{\mathbf{r}_{zq}^T \mathbf{r}_{zq}} \ dA.
\end{aligned}
\tag{2.12}
$$

The calculation above evaluates only *complete* geodesic motions for an arbitrary

frame located at z. All such motions yield proper positioning in time $t = \tau$. In other words, not all planar motions are yet considered. Recall that when $t = \tau$, each complete geodesic motion causes collocation of the body reference frame with its proper position *and* proper orientation. It was shown that all motions yield a collocation of a body connected frame at some time.

Next, those motions that yield collocation of a body connected frame at time $t = \tau$ but allow variability in angular velocity are considered. The velocity of a particle q for all such motions associated with the *partial* geodesic of frame z (at a specified, but arbitrary, location in space) can be described by

$$\mathbf{v}_{a_{zq}} = (\mathbf{R}(\theta) - \mathbf{I}) \ \mathbf{r}_{Oz}/\tau + k \left(\omega \times \mathbf{r}_{zq}\right), \tag{2.13}$$

where k is a scalar indicating the relative magnitude of the rotational motion with respect to the translational motion of the frame. The motion is a complete geodesic of the frame at z if and only if $k = 1$.

Substituting (2.13) into (2.10) yields a means of evaluating all possible planar motions (scaled such that only the specified frame achieves its proper position at time $t = \tau$),

$$v_D = \frac{\frac{1}{\tau}\sqrt{(k\theta)^2 + 2(1 - \cos\theta - k\theta\sin\theta)}}{A} \int_{q\in\mathcal{B}} \sqrt{\mathbf{r}_{zq}^T\mathbf{r}_{zq}} \ dA. \tag{2.14}$$

The difficulty of calculating this measure of velocity quality is reduced to calculating $\bar{r}$, the average particle distance from a specified location in space. This is explicit in

$$v_D = \beta(\theta, k, \tau) \ \bar{r}(\mathcal{B}, \mathbf{r}_{Cz}), \tag{2.15}$$

where

$$\bar{r}(\mathcal{B}, \mathbf{r}_{Cz}) = \frac{1}{A} \int_{q \in \mathcal{B}} \sqrt{\mathbf{r}_{zq}^T \mathbf{r}_{zq}} \ dA, \tag{2.16}$$

in which $\mathcal{B}$ is a description of the body location relative to C and $\mathbf{r}_{Cz}$ is the location of the frame undergoing the complete or partial geodesic motion relative to frame C. Also,

$$\beta(\theta, k, \tau) = \frac{1}{\tau} \sqrt{(k\theta)^2 + 2(1 - \cos\theta - k\theta \sin\theta)}. \tag{2.17}$$

It is important to note that $\beta(\theta, k)$ is independent of body geometry and that $\bar{r}(\mathcal{B}, \mathbf{r}_{Cz})$ does not depend on the body displacement from proper positioning. It depends only on the body geometry and the relative location z of the particle undergoing a straight-line translation with rotation about this point relative to the point of reference C on the body. The instantaneous motion associated with the straight-line translation of specific frame location z is determined by the displacement from proper positioning. This motion can be described at the body frame C using (2.14) with C at the particle of interest q.

The calculation of (2.16) is challenging, but not overly difficult. The measure has reduced the problem from analyzing the average velocity discrepancy associated with a geodesic (or partial geodesic) motion from a straight-line path for each particle into the calculation of the average distance of body particles from a single point in the plane — the location of the frame undergoing the geodesic (or partial geodesic). The following sections provide a better insight on the measure by evaluating (2.15) for simple geometries and identifying the best motion for a given displacement.

2.4   Best Motion Measure Analysis

It has been shown that the the best motion measure is a function of the geometry of the body. In fact, the body geometry-dependent portion of the measure is characterized by the average distance of the body particles from the frame executing the geodesic motion, $\bar{r}$. Analytical expressions for the value of $\bar{r}$ at different locations in space have been obtained for simple geometries such as rectangles, right-triangles, and circles. The analytical form of $\bar{r}$ for each of them can be found in Appendix B, C, and D, respectively. A graphical representation of the variation in values of the best motion measure with respect to z (the location of the frame undergoing the geodesic) is provided for an example rectangle, a right-triangle, and a circle to provide a better understanding.

2.4.1   Variation of Best Motion Measure with Respect to z for a Rectangle

It is possible to graphically represent the analytically obtained values of v_D for simple geometries to evaluate the quality of different motions. In each of the examples considered, the CoR is located at (0,0) and the body must be rotated 90 degrees counter-clockwise about the CoR to achieve proper position/orientation. The relative rotational magnitude k is set to 1, so that the motions considered in each plot are the same. Summarizing, $\theta = \pi/2$ and $k = 1$. For further simplicity and without loss of generality, the elapsed time τ is constant and acts as a scaling factor. With θ, k, and τ set, β is known as well.

Consider the 4×2 rectangular body and configuration shown in Fig. 1.1. As described previously, the orientational displacement is given by $\theta = \pi/2$; the translational displacement of the frame located at C is $\mathbf{r}_{CC'} = [-4, 4]^T$. Using (2.15), the value of v_D is calculated for each motion, represented by the location of

the frame undergoing the geodesic motion. The value of $\bar{r}$ for different locations in space for a rectangle is provided in Appendix B.

Figure 2.3 illustrates how v_D changes with respect to the location of the frame undergoing the geodesic motion, z. As the shading changes from light to dark, the magnitude of v_D increases. The origin of the figure corresponds to the location of the Center of Rotation for the original displacement. In this example, the minimum value of $v_{D_{k=1}}$ is 1.366, and corresponds to z located at the geometric center of the rectangle. Recall that v_D indicates the average velocity discrepancy for all body particles for this particular motion relative to the ideal particle direct-path motion.
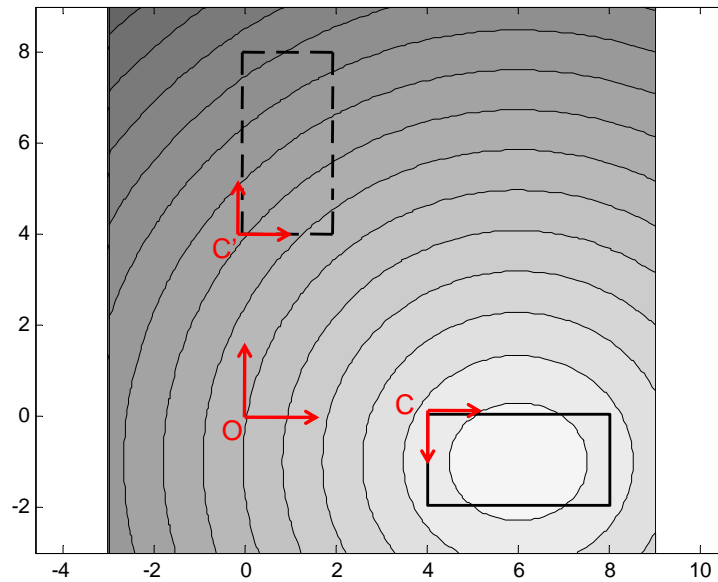
Figure 2.3: Velocity Measure v_D for a Rectangle with Respect to the Location of z, $k = 1$.

2.4.2   Variation of Best Motion Measure with Respect to z for a Right-Triangle

Consider a right-triangle of width 4 and height 2. The orientational displacement is given by $\theta = \pi/2$ and complete geodesic motions are considered. Therefore, $k = 1$. The translational displacement of the frame located at the right angle, C, is $\mathbf{r}_{CC'} = [-4, 4]^T$. Using (2.15), the value of v_D is calculated for each motion, represented by the location of the frame undergoing the geodesic motion. The analytical expression for $\bar{r}$ for different locations in space for a right-triangle is provided in Appendix C.

Figure 2.4 illustrates how v_D changes with respect to the location of the frame undergoing the geodesic motion, z. Just like the rectangle example, the shading changes from light to dark to indicate and increase in magnitude for v_D. The origin of the figure corresponds to the location of the Center of Rotation for the original displacement. In this example, the minimum value of v_D corresponds to z located at the geometric center of the right-triangle.

2.4.3   Variation of Best Motion Measure with Respect to z for a Circle

Consider a circle of radius 1. The orientational displacement is given by $\theta = \pi/2$ and complete geodesic motions are considered. Therefore, $k = 1$. The translational displacement of the frame located at the center of the circle, C, is $\mathbf{r}_{CC'} = [-4, 4]^T$. Using (2.15), the value of v_D is calculated for each motion, represented by the location of the frame undergoing the geodesic motion. The analytical expression for $\bar{r}$ for different locations in space for a circle is provided in Appendix D.

Figure 2.5 illustrates how v_D changes with respect to the location of the frame undergoing the geodesic motion, z. Just like the previous examples, the
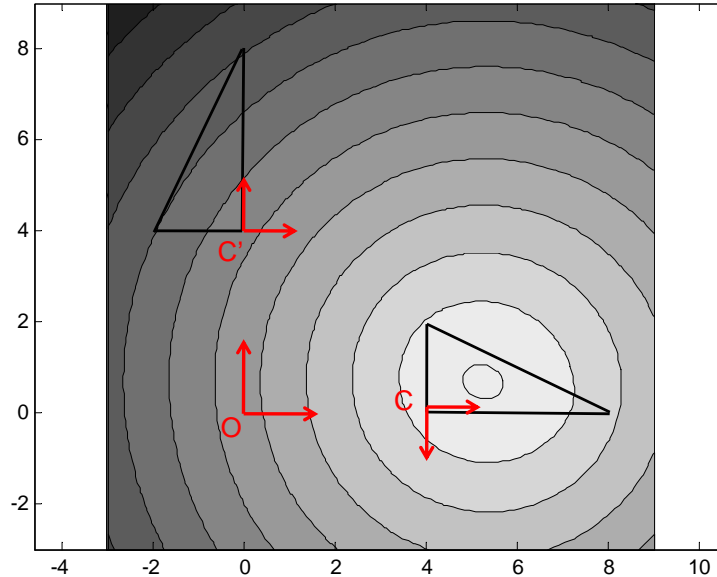
Figure 2.4: **Best Motion Measure v_D for a Right-Triangle with Respect to the Location of z, $k = 1$.**

shading changes from light to dark to indicate and increase in magnitude for v_D. The origin of the figure also corresponds to the location of the Center of Rotation for the original displacement. In this example, the minimum value of v_D corresponds to z located at the center of the circle.

2.5   Optimal Motion Analysis

In this section, the optimal planar rigid body motion is characterized based on the measure developed. From (2.15), to minimize v_D it is sufficient to minimize the values of $\bar{r}$ and $\beta(\theta, k)$. This is only true because the functions are decoupled and positive; there are no common variables between them, meaning that they are separable. The minimum value of $\bar{r}$ corresponds to the optimal location of the frame undergoing the geodesic motion. The minimum value of $\beta(\theta, k)$ corresponds to the optimal rate of rotation that minimizes the measure. Each term will be
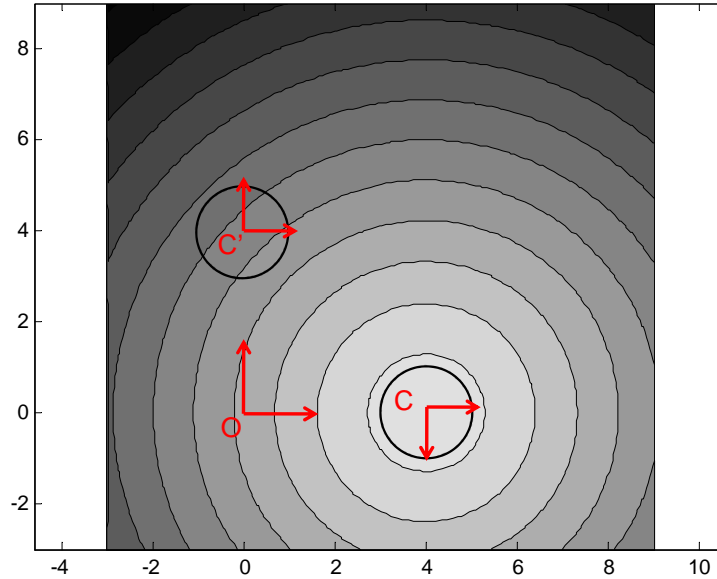
Figure 2.5: Best Motion Measure v_D for a Circle with Respect to the Location of z, $k = 1$.

analyzed separately in the following two sections.

2.5.1 Optimal Geodesic Frame Location

Recall that $\bar{r}$ is a function of the location z of the frame undergoing the geodesic and the body geometry. As stated previously and shown in (2.16), $\bar{r}$ indicates the average distance of all of the specified rigid body's particles from the location of the frame undergoing the geodesic (or partial geodesic) motion. The expression for $\bar{r}$ is very similar to the expression of the second moment of area for the perpendicular axis to the plane, $\hat{k}$, that passes through z, as widely studied in mechanics of materials [22]:

$$I_{\hat{k}} = \int_A d^2 \, dA, \tag{2.18}$$

where d is the distance from the frame located at z in consideration to the element dA, or in our case, q. However, minimizing the average distance from z to the body is analogous to minimizing the average of the square of the distance from z to the body. Therefore, the location z of the frame that minimizes $\bar{r}$ and the location z' of the frame that minimizes $I_{\hat{k}}$ are the same. It is known that $I_{\hat{k}}$ is minimum at the geometric center of the body, and the parallel axis theorem is a well-known consequence of this result. It follows that the value of $\bar{r}$ for a body is minimal when z is at the body's geometric center. The best instantaneous motion then corresponds to a geodesic (or partial geodesic) of a frame at the geometric center.

2.5.2 Optimal Rate of Rotation

Here, the optimal rate of rotation for the body is considered. The influence of the scalar k (which indicates the magnitude of the rotational motion relative to the translational motion for a specific displacement) on motion quality is assessed.

Recall that the best motion for a frame follows a minimum distance path [2, 7] which is referred to as the geodesic for that frame. The motion associated with a geodesic curve corresponds to a straight line path of the body frame from one location to the other (at constant velocity), with rotation about the axis normal to the plane passing through the frame origin (at constant angular velocity). Using the measure v_D, the motion of all particles on a finite rigid body are explicitly considered. As such, the motion of particles away from the frame performing the geodesic (or partial geodesic) motion are considered.

The motions evaluated above need not be geodesic motions. As stated previously, all motions considered carry a specified particle from its current location to its desired location on a straight-line path. If the motion is executed for a selected time τ the specified particle attains its desired location. Other particles

also attain their desired location if the value of $k = 1$. Below the optimal value of k is considered. Note that in the measure of velocity quality v_D, (2.15), only the first component β depends on either θ or k. This component is given by (2.17):

$$\beta = \frac{\sqrt{(2 - 2\cos\theta - 2k\theta\sin\theta + k^2\theta^2)}}{\tau}$$

To minimize v_D, the stationary point of $\mathrm{d}\beta/\mathrm{d}k$ is identified. This is given by:

$$
\begin{aligned}
\frac{1}{\tau}\frac{\mathrm{d}(\sqrt{2 - 2\cos\theta - 2k\theta\sin\theta + k^2\theta^2})}{\mathrm{d}k} &= 0 \\
\frac{1}{\tau}\frac{-2\theta\sin\theta + 2k\theta^2}{2\sqrt{2 - 2\cos\theta - 2k\theta\sin\theta + k^2\theta^2}} &= 0 \\
\Rightarrow -2\theta\sin\theta + 2k\theta^2 &= 0 \\
\Rightarrow k &= \frac{\sin\theta}{\theta} \qquad (2.19)
\end{aligned}
$$

Therefore, the relative magnitude of angular velocity to translational velocity is, in general, less than 1 and as θ approaches 0, the best relative angular velocity approaches 1. This makes sense intuitively. For a given rigid body displacement, the best instantaneous motion to reduce particle displacement is to primarily undergo rigid body translation. Then, as the body approaches its desired location, the best instantaneous motion is one that equally emphasizes translation and rotation. An example illustrating this is provided in the following section.

2.5.3 Optimal Motion Example

Consider the same rectangle configuration problem detailed in Section 2.4.1. Recall that k only appears in $\beta(\theta, k)$. Now, let us change the value of k to graphically show how v_D varies with respect to k. Figure 2.6 illustrates the value of v_D when $k = 0$. This corresponds to considering only pure translational motions, since $k = 0$ corresponds to $\omega = 0$. Larger values of v_D are obtained when

$k = 0$ (Fig. 2.6) versus $k = 1$ (Fig. 2.3), indicating that these motions are not as good as those for $k = 1$. The value of v_D is best for z located at the geometric center, as expected. The corresponding minimum value is $v_{D_{k=0}} = 1.678$.
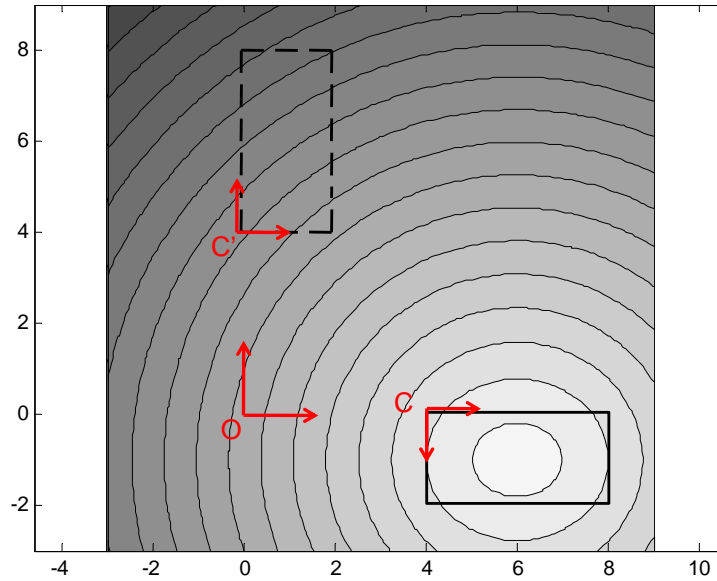
Figure 2.6: Best Motion Measure v_D for a Rectangle with Respect to the Location of z, $k = 0$.

Next consider, the case when $k = -1$. In this case the body is rotating in the opposite direction relative to how the error in orientation is measured. In other words, the body's orientational error is increasing. Figure 2.7 shows the value of v_D relative to the location of the frame z undergoing the partial geodesic motion. For this case, the minimum value of v_D for z located at the geometric center is $v_{D_{k=-1}} = 1.820$. Therefore, it is concluded that this motion is worse than any of the two motions described previously. This result makes intuitive sense, since it can be justified that to translate a body to its proper position without any rotation is a better motion than to translate the body while rotating it away from

proper orientation (and thereby increasing the discrepancy in orientation).
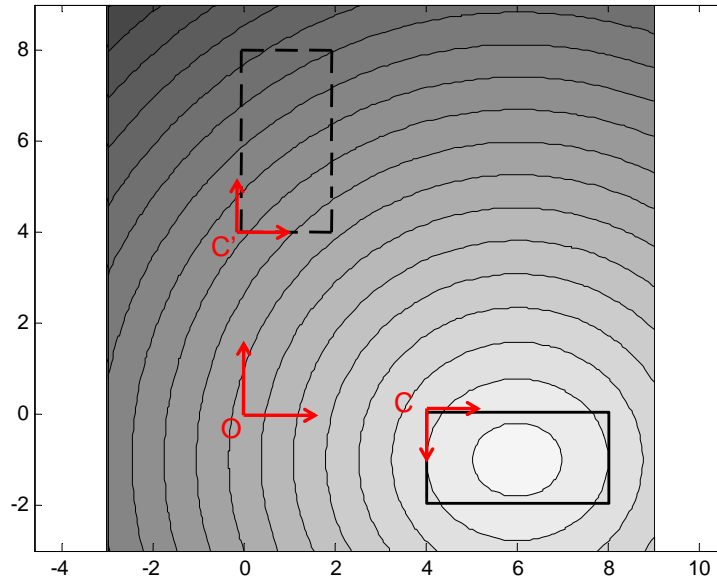
Figure 2.7: Best Motion Measure v_D for a Rectangle with Respect to the Location of z, $k = -1$.

For the configuration considered in this problem, the optimal value of k is $\frac{2}{\pi}$, given by (2.19). Figure 2.8 illustrates the values of v_D when $k = \frac{2}{\pi}$. Again, the best value is obtained when z is located at the geometric center. In this case, $v_{D_{k=2/\pi}} = 1.187$, which is significantly lower than that obtained when $k = 1$.

2.6 Instantaneous Center Representation for the Best Motion Measure

Motions can also be described by determining the instantaneous center, the point at where the instantaneous translational velocity is 0. This is an alternative to describing the motion using the location of the frame z undergoing the geodesic
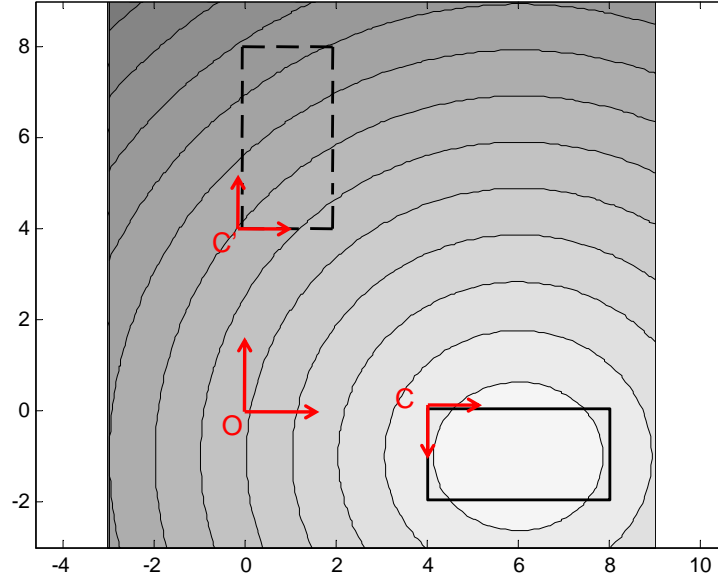
Figure 2.8: Best Motion Measure v_D for a Rectangle with Respect to the Location of z, $k = 2/\pi$.

motion as a reference.

Let o be the location of the instantaneous center for a given velocity corresponding to the geodesic motion that the frame located at z undergoes. By definition,

$$\mathbf{v}_o = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.20}$$

Furthermore, the velocity at o and the velocity at z can be related using (2.9):

$$\begin{aligned} \mathbf{v}_z &= \mathbf{v}_o + \omega \times \mathbf{r}_{oz} \\ &= [\omega_\times](\mathbf{r}_{Oz} - \mathbf{r}_{Oo}) \end{aligned} \tag{2.21}$$

However, from (2.8), $\mathbf{v}_z$ can be rewritten as:

$$\mathbf{v}_z = (\mathbf{R}(\theta) - \mathbf{I}) \ \mathbf{r}_{Oz}/\tau. \tag{2.22}$$

Combining (2.21) and (2.22) results in:

$$[\omega_\times](\mathbf{r}_{Oz} - \mathbf{r}_{Oo}) = (\mathbf{R}(\theta) - \mathbf{I}) \ \mathbf{r}_{Oz}/\tau. \tag{2.23}$$

Therefore, the location of the instantaneous center o for the velocity resulting from the partial geodesic of the frame located at z relative to the center of rotation O is then given by:

$$\mathbf{r}_{Oo} = [\mathbf{I} - \frac{1}{\tau}[\omega_\times]^{-1}(\mathbf{R}(\theta) - \mathbf{I})] \ \mathbf{r}_{Oz}. \tag{2.24}$$

Figure 2.9 illustrates the quality of motions based on the location of the instantaneous center. This figure may be used to compare results with the binary measure provided by Fig. 1.1.

2.7 Finite Motion Example

Note that the motion which has been identified as "best" is instantaneous and only applies to a *unique planar displacement.* Therefore, the motion considered to be "best" will change as the body configuration changes. The best motion measure is used to determine the best motion for this resulting configuration, and the process is reiterated until the body reaches its target position and orientation. To further illustrate this point, a simple example is developed next.

Consider the 4×2 rectangular body and configuration shown in Fig. 2.10. The initial orientational displacement is given by $\theta = \pi/2$; the initial translational
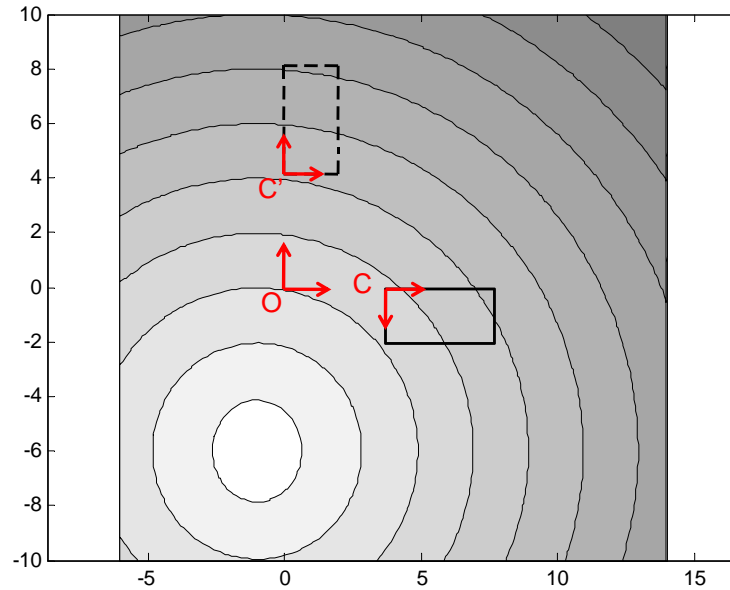
Figure 2.9: Best Motion Measure v_D for a Rectangle with Respect to the Location of o.

displacement of the frame located at C is $\mathbf{r}_{CC'} = [-4, 4]^T$. The time required to complete the motion is $\tau = 10$ seconds. Once the best motion is determined, it is executed for 0.1 seconds. Then, the new configuration parameters are calculated. The best motion for the new displacement is determined using the new configuration parameters and setting τ to the time remaining to complete the task. The result of this finite motion example is shown in Fig. 2.10.

Figure 2.10 shows the initial and final position of the body, as well as three intermediate positions. These intermediate positions correspond to time intervals of 2.5 seconds. Note that there is a greater emphasis on correcting the error in translation in the initial stages of the motion, with larger orientational correction towards the end. The geometry center of the body follows a straight-line path from its initial position to its target position as expected. These results are consistent with the mathematical analysis of the best motion measure presented in
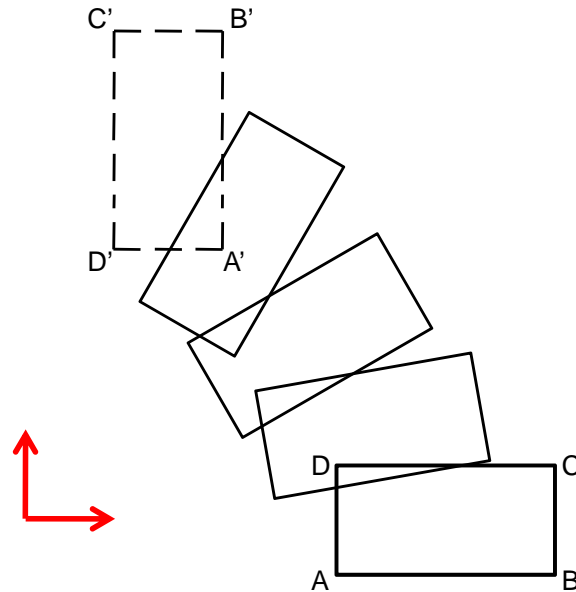
Figure 2.10: Finite Motion Example for a Rectangular Body.

Section 2.4.

2.8 Summary: Best Motion Measure

For a given displacement, the best velocity measure is given by the average difference in translational velocity (at each particle) between an available and an ideal motion. The calculation of the measure is body-specific and relates to $\bar{r}$, the average distance from the frame undergoing the straight-lie motion with rotation about this frame.

The optimal location of this frame is the geometric center of the body. The optimal angular velocity is body independent and the rate of rotation relative to translation is given by $k = \sin\theta/\theta$ where θ is the orientational discrepancy between configurations.

3. VELOCITY METRIC FOR RIGID-BODY PLANAR MOTION

As described in Chapter 1, the objective of this work is to obtain a measure of velocity quality that satisfies the mathematical requirements of a metric. First, it is shown that the measure described in Chapter 2 does not satisfy the metric requirements for comparing one rigid body motion with another. Then, a new measure is developed and shown to be a metric. A simple example related to a common assembly problem is provided.

3.1   Limitations of the Best Motion Measure

The velocity measure v_D defined by (2.14) evaluates planar rigid-body motion relative to an ideal, yet physically impossible, motion for each particle on a rigid body. Although v_D evaluates the discrepancy between two motions, one of the two motions is not an element of the set of planar rigid body motions. In other words, since the "ideal" velocity established in Chapter 2 is not a valid rigid body motion, it cannot be used to define a valid metric.

The mathematical requirements of a metric are formally described below. Let e_1, e_2, and e_3 be elements of a set X. X is a metric space if for any two elements e_1 and e_2, there is an associated real number $d(e_1, e_2)$ called the *distance* from e_1 to e_2, such that [23]:

$$d(e_1, e_2) > 0 \text{ if } e_1 \neq e_2 \,; d(e_1, e_1) = 0 \tag{3.1}$$

$$d(e_1, e_2) = d(e_2, e_1) \tag{3.2}$$

$$d(e_1, e_3) \leq d(e_1, e_2) + d(e_2, e_3). \tag{3.3}$$

The mathematical definitions provided by (3.1), (3.2), and (3.3) are commonly known as positive definiteness, symmetry, and triangle inequality respectively, and will be referred to as such in the following sections. For this particular problem, the metric space X is the set of all valid planar rigid-body motions, and e_1, e_2, and e_3 are elements of this set, planar twists.

It can be shown that with some simple modifications v_D can be transformed into a positive definite function. No simple modifications, however, can transform v_D into a function that will satisfy the principles of symmetry or triangle inequality. Therefore, a new measure used to define a velocity metric is required.

The new velocity measure is based on the evaluation of a valid planar rigid-body motion relative to another valid planar rigid-body motion. In this way, a sense of *distance* can be established between two arbitrary motions. The best motion measure defined in Chapter 2, v_D, is useful for proper characterization of the optimal motion for a given planar displacement. By choosing to evaluate all valid planar-rigid body motions relative to the best motion identified using v_D, it is possible to measure the quality of a motion relative to what is considered to be best.

The first step to establish the velocity metric is to develop a new measure in a similar manner to the way in which the best motion measure was developed in Chapter 2. Then, it will be shown that this new measure satisfies the mathematical requirements of a metric.

3.2   Velocity Metric Derivation

The velocity metric is based on a new measure similar to the best motion measure. The new measure is also based on the average velocity discrepancy due

to particle motion. In this case, however, both motions are valid rigid-body motions. This means that no "ideal", yet physically impossible, motions are considered. The first step is to define the valid rigid body motion that will be evaluated and the valid rigid-body motion used as a reference for the evaluation.

Define two arbitrary motions, or planar twists, applied to the same body at a frame C anywhere in the plane:

$$\mathbf{t_1} = \begin{bmatrix} v_{C_{x_1}} \\ v_{C_{y_1}} \\ \omega_1 \end{bmatrix}, \mathbf{t_2} = \begin{bmatrix} v_{C_{x_2}} \\ v_{C_{y_2}} \\ \omega_2 \end{bmatrix}$$

Let $\Delta\mathbf{v}_q$ be the difference in velocity for an arbitrary particle on the body, q. By definition:

$$\Delta\mathbf{v}_q = \mathbf{v}_{q_2} - \mathbf{v}_{q_1} \tag{3.4}$$

From the kinematic constraint for a rigid-body:

$$\mathbf{v}_{q_1} = \mathbf{v}_{C_1} + [\omega_{1_\times}]\mathbf{r}_{Cq} \tag{3.5}$$

$$\mathbf{v}_{q_2} = \mathbf{v}_{C_2} + [\omega_{2_\times}]\mathbf{r}_{Cq}, \tag{3.6}$$

where $[\omega_{i\times}]$ is given by: $\begin{bmatrix} 0 & \omega_i \\ -\omega_i & 0 \end{bmatrix}$. Combining (3.4), (3.5), and (3.6):

$$
\begin{aligned}
\Delta\mathbf{v}_q &= \mathbf{v}_{q_2} - \mathbf{v}_{q_1} \\
&= (\mathbf{v}_{C_2} + [\omega_{2\times}]\mathbf{r}_{Cq}) - (\mathbf{v}_{C_1} + [\omega_{1\times}]\mathbf{r}_{Cq}) \\
&= (\mathbf{v}_{C_2} - \mathbf{v}_{C_1}) + ([\omega_{2\times}] - [\omega_{1\times}])\mathbf{r}_{Cq} \\
&= \Delta\mathbf{v}_C + [\Delta\omega_\times]\mathbf{r}_{Cq}
\end{aligned}
\tag{3.7}
$$

The difference in velocity can be rewritten in matrix form:

$$
\begin{aligned}
\Delta\mathbf{v}_q &= \begin{bmatrix} 1 & 0 & -\mathbf{r}_{Cq_y} \\ 0 & 1 & \mathbf{r}_{Cq_x} \end{bmatrix} \begin{bmatrix} \Delta v_{C_x} \\ \Delta v_{C_y} \\ \Delta\omega \end{bmatrix} \\
&= [\mathbf{I}\ \mathbf{r}_\times]\,\Delta t \\
&= \mathbf{P}_{Cq}\,\Delta t,
\end{aligned}
\tag{3.8}
$$

where $P_{Cq} = [\mathbf{I}\ \mathbf{r}_\times]$ is the 2×3 transformation matrix from the difference between the twists at C to the difference in translational velocity at q.

The velocity metric, v_{m_q}, is defined as the L^2 norm of the vector described by (3.7). In other words, the square root of the dot product of this vector upon itself:

$$
\begin{aligned}
v_{m_q} &= ||\Delta\mathbf{v}_q|| \\
&= [\Delta\mathbf{v}_q^T \Delta\mathbf{v}_q]^{1/2}
\end{aligned}
\tag{3.9}
$$

Expanding,

$$
\begin{aligned}
v_{m_q} &= [\Delta\mathbf{v}_q^T \Delta\mathbf{v}_q]^{1/2} \\
&= [(\Delta\mathbf{v}_C + [\Delta\omega_\times]\mathbf{r}_{Cq})^T(\Delta\mathbf{v}_C + [\Delta\omega_\times]\mathbf{r}_{Cq})]^{1/2} \\
&= [(\Delta\mathbf{v}_C^T + \mathbf{r}_{Cq}^T[\Delta\omega_\times]^T)(\Delta\mathbf{v}_C + [\Delta\omega_\times]\mathbf{r}_{Cq})]^{1/2} \\
&= [\Delta\mathbf{v}_C^T\Delta\mathbf{v}_C + \Delta\mathbf{v}_C^T[\Delta\omega_\times]\mathbf{r}_{Cq} + \mathbf{r}_{Cq}^T[\Delta\omega_\times]^T\Delta\mathbf{v}_C + \dots \\
&\qquad \mathbf{r}_{Cq}^T[\Delta\omega_\times]^T[\Delta\omega_\times]\mathbf{r}_{Cq}]^{1/2} \\
&= [\Delta\mathbf{v}_C^T\Delta\mathbf{v}_C + 2\Delta\mathbf{v}_C^T[\Delta\omega_\times]\mathbf{r}_{Cq} + \mathbf{r}_{Cq}^T[\Delta\omega_\times]^T[\Delta\omega_\times]\mathbf{r}_{Cq}]^{1/2}
\end{aligned}
$$

$$(3.10)$$

Now, when all 3 elements of (3.10) are expanded separately:

$$
\begin{aligned}
\Delta\mathbf{v}_C^T\Delta\mathbf{v}_C &= \begin{bmatrix} \Delta v_{C_x} & \Delta v_{C_y} \end{bmatrix} \begin{bmatrix} \Delta v_{C_x} \\ \Delta v_{C_y} \end{bmatrix} \\
&= \Delta v_{C_x}^2 + \Delta v_{C_y}^2 \tag{3.11} \\
2\Delta\mathbf{v}_C^T[\Delta\omega_\times]\mathbf{r}_{Cq} &= 2\begin{bmatrix} \Delta v_{C_x} & \Delta v_{C_y} \end{bmatrix} \begin{bmatrix} 0 & -\Delta\omega \\ \Delta\omega & 0 \end{bmatrix} \begin{bmatrix} r_{Cq_x} \\ r_{Cq_y} \end{bmatrix} \\
&= 2\Delta\omega r_{Cq_x}\Delta v_{C_y} - 2\Delta\omega r_{Cq_y}\Delta v_{C_x} \\
&= 2\Delta\omega(r_{Cq_x}\Delta v_{C_y} - r_{Cq_y}\Delta v_{C_x}) \tag{3.12} \\
\mathbf{r}_{Cq}^T[\Delta\omega_\times]^T[\Delta\omega_\times]\mathbf{r}_{Cq} &= \begin{bmatrix} r_{Cq_x} & r_{Cq_y} \end{bmatrix} \begin{bmatrix} 0 & \Delta\omega^2 \\ \Delta\omega^2 & 0 \end{bmatrix} \begin{bmatrix} r_{Cq_x} \\ r_{Cq_y} \end{bmatrix} \\
&= \Delta\omega^2(r_{Cq_x}^2 + r_{Cq_y}^2) \tag{3.13}
\end{aligned}
$$

Combining and rearranging yields:

$$
\begin{aligned}
v_{m_q} &= [\Delta v_{C_x}^2 + \Delta v_{C_y}^2 + 2\Delta\omega(r_{Cq_x}\Delta v_{C_y} - r_{Cq_y}\Delta v_{C_x}) + \ldots \\
&\quad \Delta\omega^2(r_{Cq_x}^2 + r_{Cq_y}^2)]^{1/2} \\
&= [(\Delta v_{C_y}^2 + 2\Delta\omega r_{Cq_x}\Delta v_{C_y} + \Delta\omega^2 r_{Cq_x}^2) + \ldots \\
&\quad (\Delta v_{C_x}^2 - 2\Delta\omega r_{Cq_y}\Delta v_{C_x} + \Delta\omega^2 r_{Cq_y}^2)]^{1/2} \\
&= \sqrt{(\Delta\omega r_{Cq_x} + \Delta v_{C_y})^2 + (\Delta\omega r_{Cq_y} - \Delta v_{C_x})^2}. \quad (3.14)
\end{aligned}
$$

Note that the location of particle q relative to the frame C, $\mathbf{r}_{Cq}$, must be accounted for all particles on the body. The average velocity discrepancy for all particles on the body $q \in \mathcal{B}$ can be expressed as:

$$
\begin{aligned}
v_M &= \frac{1}{A}\int_{q\in\mathcal{B}} v_{m_q}\, dA \quad (3.15) \\
&= \frac{1}{A}\int_{q\in\mathcal{B}} \sqrt{(\Delta\omega r_{Cq_x} + \Delta v_{C_y})^2 + (\Delta\omega r_{Cq_y} - \Delta v_{C_x})^2}\, dA \quad (3.16)
\end{aligned}
$$

Another way to describe $\Delta\mathbf{v}_q$ is to determine the location of the instantaneous center for the motion also described by (3.7). Let o be the location in the plane where $\Delta\mathbf{v}_o = \mathbf{0}$. The location of o relative to C is obtained from the kinematic constraint:

$$
\begin{aligned}
\Delta\mathbf{v}_o &= \Delta\mathbf{v}_C + [\Delta\omega_\times]\mathbf{r}_{Co} \\
\Rightarrow \mathbf{r}_{Co} &= -[\Delta\omega_\times]^{-1}\Delta\mathbf{v}_C \quad (3.17)
\end{aligned}
$$

Then the discrepancy in translational velocity at particle q of the body is

expressed as:

$$\Delta\mathbf{v}_q \;=\; \Delta\mathbf{v}_o + [\Delta\omega_\times]\mathbf{r}_{oq}$$

$$\;=\; [\Delta\omega_\times]\mathbf{r}_{oq}, \tag{3.18}$$

Then, v_{m_q} becomes:

$$v_{m_q} \;=\; [\Delta\mathbf{v}_q^T \Delta\mathbf{v}_q]^{1/2}$$

$$\;=\; [([\Delta\omega_\times]\mathbf{r}_{oq})^T ([\Delta\omega_\times]\mathbf{r}_{oq})]^{1/2}$$

$$\;=\; [(\mathbf{r}_{oq}^T [\Delta\omega_\times]^T)([\Delta\omega_\times]\mathbf{r}_{oq})]^{1/2}$$

$$\;=\; [\mathbf{r}_{oq}^T [\Delta\omega_\times]^T [\Delta\omega_\times]\mathbf{r}_{oq}]^{1/2}. \tag{3.19}$$

Simplifying,

$$\mathbf{r}_{oq}^T [\Delta\omega_\times]^T [\Delta\omega_\times]\mathbf{r}_{oq} \;=\; \begin{bmatrix} r_{oq_x} & r_{oq_y} \end{bmatrix} \begin{bmatrix} 0 & \Delta\omega^2 \\ \Delta\omega^2 & 0 \end{bmatrix} \begin{bmatrix} r_{oq_x} \\ r_{oq_y} \end{bmatrix}$$

$$\;=\; \Delta\omega^2 (r_{oq_x}^2 + r_{oq_y}^2). \tag{3.20}$$

Therefore, the new velocity measure for a single particle can alternately be expressed as:

$$v_{m_q} \;=\; \sqrt{\Delta\omega^2 (r_{oq_x}^2 + r_{oq_y}^2)}$$

$$\;=\; |\Delta\omega| \sqrt{\mathbf{r}_{oq}^T \mathbf{r}_{oq}}. \tag{3.21}$$

The location of o relative to q can be calculated easily by combining (3.7) and

(3.18):

$$\Delta \mathbf{v}_C + [\Delta \omega_\times] \mathbf{r}_{Cq} = [\Delta \omega_\times] \mathbf{r}_{oq}$$

$$\Rightarrow \mathbf{r}_{oq} = \mathbf{r}_{Cq} + [\Delta \omega_\times]^{-1} \Delta \mathbf{v}_C \tag{3.22}$$

Similarly, the average velocity discrepancy for all particles on the body $q \in \mathcal{B}$ can be expressed as:

$$v_M = \frac{|\Delta \omega|}{A} \int_{q \in \mathcal{B}} \sqrt{\mathbf{r}_{oq}^T \mathbf{r}_{oq}} \, dA \tag{3.23}$$

Note that the instantaneous center is located at infinity when $\Delta \omega = 0$. In this particular case, (3.23) will not yield the proper result. The original expression for v_M (3.16) indicates that when this is the case, the metric consists solely of the difference in translational velocity. This result is consistent with what is expected of a situation in which the angular velocity of two motions is identical.

The calculation of v_M using (3.23) again reduced to calculating $\bar{r}$, the average particle distance from a specified location in space. This is explicit in

$$v_M = \gamma(\Delta \omega) \; \bar{r}(\mathcal{B}, \mathbf{r}_{oC}), \tag{3.24}$$

where

$$\bar{r}(\mathcal{B}, \mathbf{r}_{oC}) = \frac{1}{A} \int_{q \in \mathcal{B}} \sqrt{\mathbf{r}_{oq}^T \mathbf{r}_{oq}} \; dA, \tag{3.25}$$

and

$$\gamma(\Delta \omega) = |\Delta \omega|. \tag{3.26}$$

Unlike v_D, the elements of v_M are not decoupled. Note that $\mathbf{r}_{oq}$ is a function of $\Delta\omega$ as well (3.22). The heaviest computational burden, the integration over the area of the body, remains the same. The only significant difference between v_D and v_M is the need to calculate the location of o. However, this calculation is trivial. Therefore, there is no significant additional computational burden associated with evaluating v_M relative to v_D. Furthermore, v_M is a metric in the space of all valid rigid-body planar motions, as shown next.

3.3   Satisfying the Mathematical Requirements of a Metric

It will now be shown that the velocity error measure given by (3.15) satisfies the mathematical conditions of a metric. Let $v_{M_{t_1 t_2}}$ be the measure for two arbitrary motions $\mathbf{t_1}$ and $\mathbf{t_2}$. Similarly, $v_{M_{t_2 t_3}}$ and $v_{M_{t_1 t_3}}$ represent the discrepancy between $\mathbf{t_2}$ and $\mathbf{t_3}$, and $\mathbf{t_1}$ and $\mathbf{t_3}$ respectively.

Recall from (3.15) that v_M is given by:

$$v_{M_{t_1 t_2}} = \frac{1}{A} \int_{q \in \mathcal{B}} v_{m_{q12}} \, dA, \tag{3.27}$$

where $v_{m_{q12}}$ is the norm of the difference in translational velocity at q. From (3.8):

$$
\begin{aligned}
v_{m_{q12}} &= \|\mathbf{P}_{Cq} \Delta \mathbf{t_{12}}\| \\
&= \|\mathbf{P}_{Cq} (\mathbf{t_2} - \mathbf{t_1})\| \tag{3.28}
\end{aligned}
$$

The L^2 norm of a vector is always positive or 0, so it follows that $v_{m_q} \geq 0$ for all q. Recall from (3.8) that $\mathbf{P}_{Cq}$ is a transformation matrix, uniquely determined by the location of q. Therefore, $v_{m_{q12}}$ is 0 for all q if Δt_{12} is 0, which is equivalent to having $\mathbf{t_1} = \mathbf{t_2}$. Since $v_{M_{t_1 t_2}}$ is the average of $v_{m_{q12}}$ for all q in the body, the

measure can be either positive, or 0 when the two twists are identical. In conclusion, the measure given by v_M satisfies positive definiteness.

For the measure to satisfy symmetry, it is necessary that $v_{M_{t_1 t_2}} = v_{M_{t_2 t_1}}$. To prove this identity, it is necessary to show that $v_{m_{q_{12}}} = v_{m_{q_{21}}}$ for all q, since the average of identical elements is identical. Also recall that $||x|| = || - x||$ for all x. Expanding terms:

$$
\begin{aligned}
v_{m_{q_{12}}} &= ||\mathbf{P}_{Cq}(\mathbf{t_2} - \mathbf{t_1})|| \\
&= || - \mathbf{P}_{Cq}(\mathbf{t_1} - \mathbf{t_2})|| \\
&= ||\mathbf{P}_{Cq}(\mathbf{t_1} - \mathbf{t_2})|| \\
&= v_{m_{q_{21}}}
\end{aligned}
\tag{3.29}
$$

It follows that the measure given by v_M satisfies symmetry.

To show that a metric satisfies the triangle inequality, it must be shown that $d(t_1, t_2) + d(t_2, t_3) \geq d(t_1, t_3)$ for three arbitrary motions. It will be shown that this statement holds for any arbitrary particle q in the body. Then, by adding this result for all $q \in \mathcal{B}$, the measure will satisfy the triangle inequality for the body.

Recall from (3.9) that, for an arbitrary q on the body, the measure is defined as the squared root of the norm of the discrepancy in velocity: $v_{m_{q_{12}}} = ||\Delta \mathbf{v}_{q_{12}}||$. Note that $\Delta \mathbf{v}_{q_{13}} = \mathbf{v}_{q_3} - \mathbf{v}_{q_1} = (\mathbf{v}_{q_3} - \mathbf{v}_{q_2}) + (\mathbf{v}_{q_2} - \mathbf{v}_{q_1}) = \Delta \mathbf{v}_{q_{23}} + \Delta \mathbf{v}_{q_{12}}$. Therefore, it must be shown that $||\Delta \mathbf{v}_{q_{12}}|| + ||\Delta \mathbf{v}_{q_{23}}|| \geq ||\Delta \mathbf{v}_{q_{12}} + \Delta \mathbf{v}_{q_{23}}||$. However, this is a known property of the L^2 norm. Therefore, v_M satisfies the triangle inequality. Since v_M

satisfies all three requires properties, it is concluded that v_M is a metric.

The measure v_M has been properly defined and it has been shown that it satisfies the mathematical conditions of a metric. It is possible to use v_M to evaluate the quality several motions in a typical assembly problem, as shown next.

3.4   Example of Motion Evaluation

Consider the classic peg-in-the-hole assembly problem involving the 3×2 rectangular body and configurations illustrated in Fig. 2. The orientational displacement is: $\theta = -\pi/6$ (ccw: $+$); the translational displacement of the frame located at the geometric center G is: $\mathbf{p}_{GG'} = [4, -4]^T$. In this particular example, the time selected to complete the assembly is $\tau = 4$. The frame at which the twists are defined is located at the geometric center.
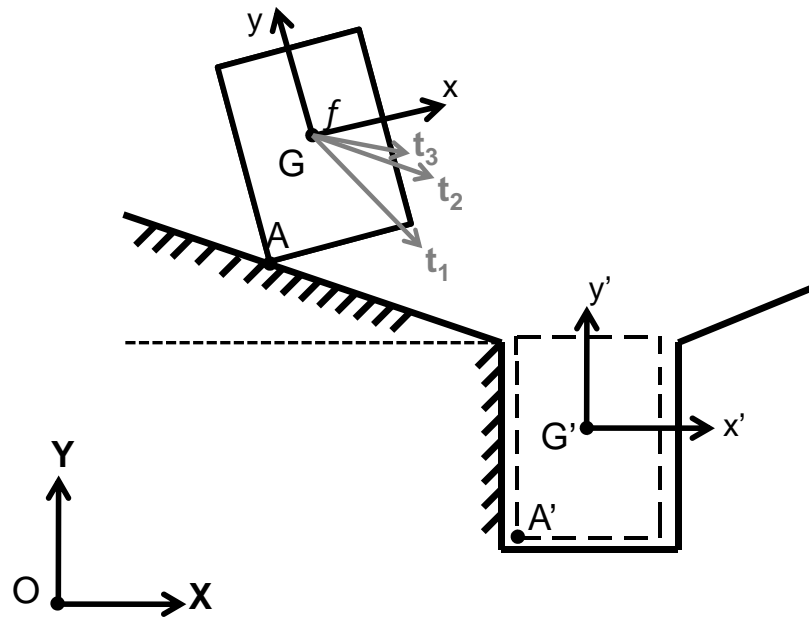
Figure 3.1: Rectangular Body Configuration Problem.

Recall that the best motion as identified using the best motion measure. It

corresponds to direct translation of the geometric center from its current position to its desired position while rotating about the axis perpendicular to the plane of motion. The translational velocity corresponding to the best motion is given by:

$$\mathbf{v}_1 = \frac{\mathbf{r}_{GG'}}{\tau} \tag{3.30}$$

For the misalignment illustrated, the translational displacement of the geometric center G is given by $\mathbf{r}_{GG'} = [4, -4]^T$. Hence,

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{3.31}$$

The body configuration determines the optimal rate of rotation $k_{\min}$:

$$k_{\min} = \sin\theta/\theta = \frac{\sin(-\pi/6)}{-\pi/6} = 0.955. \tag{3.32}$$

It follows that the optimal angular velocity is:

$$\omega_1 = \frac{k\theta}{\tau} = \frac{0.955(-\frac{\pi}{6})}{4} = -0.125. \tag{3.33}$$

Therefore, the planar twist $\mathbf{t}_1$ that describes the best motion for the frame f is:

$$\mathbf{t}_1 = \begin{bmatrix} \mathbf{v}_1 \\ \omega_1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -0.125 \end{bmatrix} \tag{3.34}$$

Note that in this particular example, the best motion is unavailable because it would cause the body to penetrate into the chamfer. This situation may arise depending on the magnitude of the angle of the chamfer. Regardless, this motion

can still be used as a standard in determining the quality of other motions. The second planar twist is chosen to be a physically available pure-translation motion. This motion consists of pure translation of the body with particle A maintaining contact with the chamfer. The direction is determined by the chamfer geometry. Therefore, the second planar twist at frame f is given by:

$$\mathbf{t}_2 = \begin{bmatrix} \mathbf{v}_2 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.5 \\ 0 \end{bmatrix} \tag{3.35}$$

The velocity metric v_M can be used to provide a measure of the quality of the planar motion represented by $\mathbf{t_2}$ relative to what is considered to be the best rigid-body motion: $\mathbf{t_1}$. Note that:

$$\Delta\mathbf{v}_f = \mathbf{v}_2 - \mathbf{v}_1 = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \tag{3.36}$$

$$[\Delta\omega_\times] = \begin{bmatrix} 0 & 0.125 \\ -0.125 & 0 \end{bmatrix} \tag{3.37}$$

The first step is to determine the location of o relative to f, as expressed in (3.17):

$$\begin{aligned} \mathbf{r}_{fo} &= -[\Delta\omega_\times]^{-1}\mathbf{v}_f \\ &= -\begin{bmatrix} 0 & -8 \\ 8 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \\ &= \begin{bmatrix} 4 \\ 0 \end{bmatrix} \end{aligned} \tag{3.38}$$

The metric v_M can now be calculated using (3.24). The first function γ is easily calculated from (3.26): $\gamma = |\Delta\omega| = 0.125$. The average distance from o to the particles on the body, $\bar{r}$, can be calculated using the analytical form for a rectangle described in Appendix B. It follows that for this particular example:

$$v_M(\mathbf{t}_1, \mathbf{t}_2) = 1.112. \tag{3.39}$$

For the same assembly/configuration problem, consider a third planar motion $\mathbf{t}_3$. This planar twist is a reciprocal motion for this particular assembly problem. It also maintains contact at A. The instantaneous center of a reciprocal motion lies on the line normal to the chamfer through the point of contact A. The best location for the instantaneous center along the normal line can be calculated given the body and chamfer geometry. The angular velocity of $\mathbf{t}_3$ is equal to the angular velocity of $\mathbf{t}_1$, which was calculated using the best motion measure. It follows that the planar motion $\mathbf{t_3}$ is expressed as:

$$\mathbf{t}_3 = \begin{bmatrix} \mathbf{v}_3 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} 0.621 \\ -0.329 \\ -0.125 \end{bmatrix} \tag{3.40}$$

The difference in translational velocity at frame f, $\Delta\mathbf{v}_f$ is given by:

$$\begin{aligned} \Delta\mathbf{v}_f &= \mathbf{v}_3 - \mathbf{v}_1 \\ &= \begin{bmatrix} 0.621 \\ -0.239 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} -0.379 \\ 0.761 \end{bmatrix} \end{aligned} \tag{3.41}$$

Equation (3.21) can no longer be used to calculate v_M for $\mathbf{t_1}$ and $\mathbf{t_3}$ since $\Delta\omega = 0$. Therefore, (3.16) must be used instead:

$$
\begin{aligned}
v_M(\mathbf{t}_1, \mathbf{t}_3) &= \|\Delta\mathbf{v}_f\| \\
&= \sqrt{(-0.379)^2 + (0.761)^2} \\
&= 0.850
\end{aligned}
\tag{3.42}
$$

From (3.39) and (3.42), note that $v_M(\mathbf{t}_1, \mathbf{t}_3) < v_M(\mathbf{t}_1, \mathbf{t}_2)$. Therefore, $\mathbf{t}_3$ is *closer* to the best motion than $\mathbf{t}_2$. In this example where the best motion is not available because of a physical boundary constraint, $\mathbf{t}_3$ is the better choice at that particular instant in time. Note that this analysis is instantaneous, meaning that is only valid for this exact moment in time. Once the configuration changes, the discrepancy between motions changes as well.

3.5 Summary: Velocity Metric

For two planar motions $\mathbf{t}_1$ and $\mathbf{t}_2$ at a frame f, the velocity metric is given by the integration of the difference in translational velocity for all particles on the body. If the difference in angular velocity $\Delta\omega$ is 0, the calculation of the metric is simplified to calculating the difference in translational velocity at f between the motions.

If $\Delta\omega \neq 0$, the metric is a function of the absolute value of the discrepancy in angular velocity between the two motions, and the average distance from the instantaneous center associated with the twist discrepancy to every particle on the body, $\bar{r}$. The location of this instantaneous center can alternately be viewed as the location of a frame where the translational velocities associated with $\mathbf{t}_1$ and $\mathbf{t}_2$ are

identical.

Note that for both the velocity metric and the best motion measure, the calculation is reduced to determining the average distance from a location in the plane to the particles on the body. Although analytical forms of $\bar{r}$ are available for simple geometries such as rectangles, right-triangles, and circles, an analytical form for an arbitrary polygonal geometry is not available. The next chapter provides a method to calculate $\bar{r}$ for any polygonal body with polygonal and/or circular holes.

4. CALCULATION OF r̄ FOR A POLYGONAL BODY

Recall that the best motion measure v_D for a given displacement can be written as the product of $\beta(\theta, k)$ and $\bar{r}(\mathcal{B}, \mathbf{r}_{zf})$, while the velocity metric v_M can be written as the product of $\gamma(\Delta\omega)$ and $\bar{r}(\mathcal{B}, \mathbf{r}_{of})$. The components identified as $\beta(\theta, k)$ and $\gamma(\Delta\omega)$ are independent of body geometry. Therefore, only the average distance from a location in the plane to the particles on the body, $\bar{r}$, depends on body geometry. An analytical form of $\bar{r}$ for simple geometries (rectangles, right-triangles and circles) was presented in Chapter 2. This chapter presents an algorithm to calculate $\bar{r}$ for a general polygonal body.

The value of $\bar{r}$ for general polygonal bodies with holes can be calculated by decomposing a polygon into separate simple sub-bodies, then adding (or subtracting) the $\bar{r}$ value calculated for each simple sub-body. This chapter presents an algorithm used to calculate $\bar{r}$ for any polygonal body with polygonal or circular holes. The algorithm is divided into two parts. The first part consists of decomposing the main body (with holes) into an equivalent group of simple geometry bodies for which $\bar{r}$ can be calculated analytically. The second part of the algorithm computes the value of $\bar{r}$ for: 1) the set of simple geometry bodies obtained from the decomposition, and 2) the specified location from which average body distance, $\bar{r}$, is needed. Note that the decomposition needs to be run only once for the body, while the $\bar{r}$ calculation for a specific body can be efficiently calculated in real time for any specified location of the point in the plane.

4.1 Strategy Overview

The approach used to calculate $\bar{r}$ for composite bodies is similar to the approach used to calculate the mass moment of inertia of a polygonal body. The value of $\bar{r}$ can be calculated by adding simple bodies or subtracting simple bodies (holes). The calculation for a composite body $\mathcal{B}$ made of N simple bodies and M simple holes is given by

$$\bar{r}(\mathcal{B}, \mathbf{H}_{CC'}) = \frac{\sum_{i=1:N} \bar{r}(\mathcal{B}_i, \mathbf{H}_{C_i C_i'}) A_i - \sum_{j=1:M} \bar{r}(\mathcal{B}_j, \mathbf{H}_{C_j C_j'}) A_j}{\sum_{i=1:N} A_i - \sum_{j=1:M} A_j}, \quad (4.1)$$

where A_k is the area of a sub-body or sub-hole $\mathcal{B}_k$, and $\mathbf{H}_{C_k C_k'}$ is the displacement of sub-body or sub-hole $\mathcal{B}_k$ described by body frame C_k.
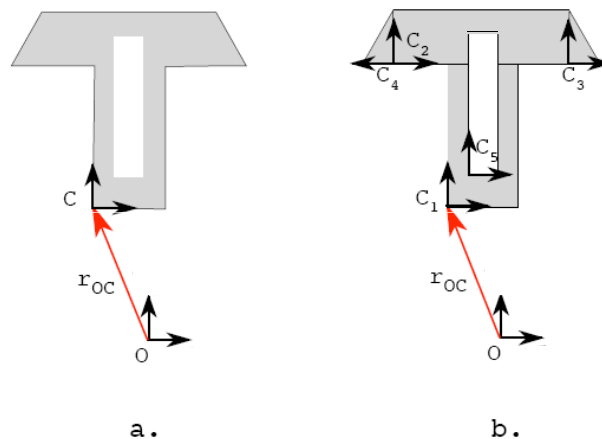
Figure 4.1: Decomposition of Example Polygon with a Hole.

The decomposition of an example polygonal body is shown in Fig. 4.1. The body illustrated is composed of 4 sub-bodies and 1 void-body or hole. The average distance $\bar{r}$ is calculated for each body independently and then weight-averaged to

determine $\bar{r}$ for the composite body, using (4.1). Figure 4.1a shows the polygonal body's position and orientation relative to a chosen frame. Figure 4.1b illustrates the 5 simple-geometry sub-bodies into which the original body was decomposed. Note that the decomposition of the polygonal body only needs to be performed once. The calculation of $\bar{r}$ for different locations relative to the body can be performed as many times as necessary once the decomposition has been performed.

This section has provided only a conceptual description of how to calculate the average distance from a point in the plane to an arbitrary polygonal body using a body decomposition strategy. The next section details the algorithm used to calculate $\bar{r}$ for an arbitrary polygonal body using polygon triangulation.

4.2   Algorithm to Calculate $\bar{r}$

As stated previously, the calculation of $\bar{r}$ for a general polygonal body has been separated into two tasks. The first task is to decompose an arbitrary polygonal body with polygonal and circular holes into a group of sub-bodies for which an analytical form of $\bar{r}$ is available for each. The second task is to efficiently calculate $\bar{r}$ for each simple geometry using the corresponding analytical form, and then calculate $\bar{r}$ for the body using (4.1) for the original geometry. Each of these two tasks is described in detail in the following two subsections. The functions used to calculate necessary elements in each task are discussed in the next section.

4.2.1   Decomposition of an Arbitrary Polygonal Body

This subsection describes the first part of the algorithm used to calculate the average distance from a point in the plane to a polygonal body with polygonal and/or circular holes. This task consists of decomposing the body and each hole into a set of right-triangles or circles, which are geometries for which an analytical

form of $\bar{r}$ is available. A general overview of the polygonal decomposition routine is presented in Fig. 4.2.

```
Input: Body Geometry in DXF Format


    Identify Main Body Polygon
    Identify # of Polygonal Holes (P)
    Identify # of Circular Holes (C)
    Decompose main body into "t" right-triangles
    Store Geometry & Positioning of each
    Right-rtriangle in body array (BodyTri)
    For i = 1:P
        Decompose polygonal hole into right-triangles
        Store geometry & positioning of each in single
        polygonal hole array (HoleTri)
        Increase total number of polygonal
            holes counter (p=p+1)
    For i = 1:C
        Store geometry & positioning of each in single
        circular hole array (HoleCir)
Output:
    Main Body Array (5 x t)
    Polygonal Hole Array (5 x p)
    Circular Hole Array (3 x C)
```

Figure 4.2: *PolyDecomp.m* **Overview.**

This routine initially prompts the user for a DXF geometry file. DXF files provide all the necessary information concerning body geometry in ASCII format. The information required from the DXF file consists of the vertices of each polygonal sub-body given in counter-clockwise direction, and the radius and

location of the center of each circle. MATLAB scripts that accomplish this are available for public use in the MATLAB Central file-sharing website [24], but these scripts extract more information from the DXF file than is necessary. Therefore, one of the many available scripts, *DXF.m* [25], was modified to obtain only the relevant information identified above. The modified function, *ReadDxf.m*, reads a DXF geometry file and outputs two arrays containing polygonal body and circular body data. The function can be found in Appendix E. The polygon and circle data is then classified into three groups: the polygonal main body, the polygonal holes and the circular holes. Since no analytical form is available for arbitrary polygonal bodies, the next step is to decompose the main body and each of the polygonal holes into right-triangles.

Polygonal bodies (or holes) are divided into right-triangles in two steps. The first step is to divide each polygonal body (or hole) into triangles using a process called triangulation. The second step consists of splitting each resulting triangle into two right-triangles. The triangulation procedure is analyzed in detail in Section 4.3. The decomposition of a triangle into two right-triangles and the function used to obtain the necessary body geometry and configuration information is explained in detail in Section 4.4.

To summarize, the polygonal decomposition routine reads a DXF geometry file and outputs three arrays: 1) body geometry and configuration information for right-triangular bodies, 2) body geometry and configuration information for right-triangular holes, and 3) body geometry and configuration for circular holes. The MATLAB script for the routine described above, *PolyDecomp.m*, can be found in Appendix F.

4.2.2 Computation of r̄ for a Decomposed Geometry

This subsection describes the second part of the algorithm used to calculate the average distance from a point in the plane to a polygonal body with holes. This task consists of calculating: 1) $\bar{r}$ for each of the simple geometries into which the original body geometry was decomposed, and 2) calculating $\bar{r}$ of the original body using (4.1). A general overview of the $\bar{r}$ computation algorithm is given by Fig. 4.3.

The $\bar{r}$ computation routine uses the three data arrays obtained from the polygonal decomposition routine as input. These arrays contain body geometry and configuration information for right-triangular bodies, right-triangular holes and circular holes. The location of the point in the plane for which the average distance to the body will be calculated must also be provided. The average distance $\bar{r}$ for each body (or hole) is calculated using the analytical form provided in Appendix B and C for a right-triangle and a circle respectively. The average distance for the original body is calculated using the composite body decomposition provided in (4.1).

The function *ComputeRbar.m* found in Appendix G is used to execute the algorithm described above. The next section provides further insight on each of the functions used in the polygonal decomposition routine.

4.3 Polygonal Right-Triangulation

Recall that the function *ReadDxf.m* outputs three arrays containing geometry and configuration information for polygonal bodies, polygonal holes and circles. A polygon is uniquely defined based on the location of each vertex given in counter-clockwise order relative to a specified world-frame. In this section, the triangulation algorithm for polygonal bodies and holes is provided. The output of

```
Input:
    Main Body Array - BodyTri (5 x t)
    Polygonal Hole Array - HoleTri (5 x p)
    Circular Hole Array - HoleCir (3 x C)
    Location of interest – Pt(x,y)


    RA = sum of r-bar*Area for all bodies
    For i = 1:t
        Calculate r-bar & area for each triangular body
        RA = RA+ r-bar*Area(i)
    For i = 1:p
        Calculate r-bar & area for each triangular hole
        RA = RA - r-bar*Area(i)
    For i = 1:C
        Calculate r-bar & area for each circular hole
        RA = RA - r-bar*Area(i)
    r-bar for original body = RA/Total Area


Output:
    r-bar of the original polygonal body (with polygonal
    and circular  holes
```

Figure 4.3: *ComputeRbar.m* **Overview.**

the triangulation algorithm is then used to divide each triangle into right-triangles.

4.3.1   Algorithm Overview

The polygonal bodies and polygonal holes are each stored as an $n \times 2$ array, where each row identifies the $x-$ and $y-$ locations of each vertex sequentially in counter-clockwise direction. The vertices are numbered in the order in which they are provided, with the first vertex assigned 1 and the last-vertex being n for an

n-vertex polygon. The sides or edges of the polygon are determined by the union of consecutive vertices and are therefore labeled as such. In general, $(i, i+1)$ is the i-th edge associated with vertex i. Note that the last edge that "closes" the polygon is given by $(n, n+1)$, where $n+1$ corresponds to vertex 1.

The function used for polygonal triangulation, *Triangulation.m*, splits a polygon into a set of triangles by dividing the original polygon (at least 4 vertices) into two polygons of lesser order. The process is repeated with the resulting polygons until only triangles remain. Then, the vertices that define each triangle are stored in a $6 \times (n-2)$ array, where $(n-2)$ is the number of triangles resulting from decomposing a polygon with n vertices. Figure 4.4 provides a high-level overview of this function. The algorithm used to divide a polygon into two polygons of lesser order varies depending on the type of polygon (convex or concave).

The function *Triangulation.m* that takes the initial polygon and outputs the vertices of each triangle, the result of the decomposition process, can be found in Appendix H. The function *PolySplit.m* used to separate a polygon into two polygons of lesser order is analyzed in the next section.

4.3.2 Polygonal Decomposition into Two Polygons of Lesser Order

A polygon can be decomposed into two polygons of lesser order by drawing a line segment that connects two non-adjacent vertices, as long as this line segment is entirely contained in the original polygon. The function *PolySplit.m* scans the vertices of a polygon looking for a pair of suitable vertices to perform this division. The selection of the vertices to be connected is based on the type of polygon. Polygons can be classified into two categories depending on the characteristics of their internal angles: convex and concave polygons. The internal

```
Input:

    Polygonal Array (2 x n)


    Store input polygon in the 1st row of a Temp array

    Decompose into two polygons, Poly1 and Poly2,
        based on polygon type (concave or convex)

    Store Poly1 and Poly2 into next 2 rows of Temp array

    if next row of Temp is not empty

        Read next row of Temp (Poly)

        if # vertices > 3

            Decompose Poly into two polygons, Poly1 and
            Poly2, based on polygon type (concave or
            convex)

            Store into next 2 rows of Temp

        if # vertices = 3

            Store Poly in Triangle array

    if next row of temp is empty

        Terminate algorithm


Output:

    Triangle array (6 x n-2)
```

Figure 4.4: *Triangulation.m* **Overview.**

angle of a vertex is the angle defined by its two adjacent edges on the inner side of the polygon. If all the internal angles in a polygon are less than 180°, then the polygon is convex. Concave polygons are then those that have at least one internal angle greater than 180°.

In a convex polygon, the line segment that joins a vertex with any non-adjacent vertex is always entirely contained in the polygon. Therefore, it does

not matter which two vertices are selected to split the polygon as long as the two vertices are not consecutive. However, concave polygons are different. A convex vertex in a concave polygon may not have a suitable non-adjacent vertex that can be used to split the polygon, as shown in Fig. 4.5. No splitting line segment between vertices exists that uses vertex 1 as one of its end points and is entirely contained in the polygon. On the other hand, there is at least one vertex which can be used along with a concave vertex to split the original polygon into two polygons. Using the concave vertex guarantees that there will be at least one non-adjacent vertex such that the line segment between them is entirely contained within the polygon. The second vertex to be used is calculated by systematically going through every non-adjacent vertex and checking to see if the line segment is entirely contained within the polygon. In summary, a means to determine whether the polygon under question is concave or convex, is required to decompose a polygon into two polygons. If the polygon is concave, then the concave vertex must be identified as well.

Although a polygon can be classified as convex or concave depending on the measure of its internal angles, a different approach to properly classify polygons is required. The internal angle cannot be measured accurately since it is impossible to determine which is the internal angle by knowing just the two edges that form the angle. However, we can take advantage of the fact that the vertices are given in counter-clockwise order. Graham [26] described an algorithm to compute the convex hull of a set of points in the plane, called the Graham scan. The algorithm is based on the notion of whether a "right-turn" or a "left-turn" is necessary to go from two consecutive vertices to the next. If the vertices are given in counter-clockwise order and a right-turn is necessary, then the vertex is concave. If a left-turn is necessary, then the vertex is convex. A graphical representation of a
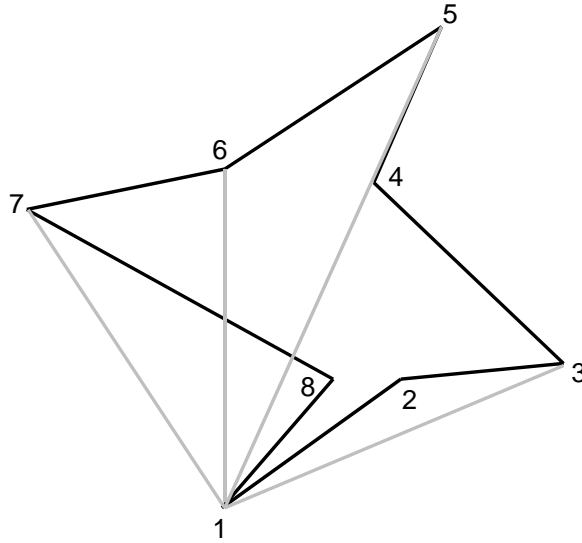
Figure 4.5: Concave Octagon Example.

left-handed turn, Fig. 4.6a, and a right-handed turn, Fig. 4.6b, for consecutive

vertices given in counter-clockwise order is shown.

To determine whether three consecutive vertices constitute a left-turn or a

right-turn, it is not required to compute the actual angle between the two edges.

This result can be achieved with simple arithmetic only. For three consecutive

vertices $i-1$, i and $i+1$, determining the direction of the cross-product of the two

vectors defined by $(i-1, i)$ and $(i-1, i+1)$ is enough. This is characterized by

the sign of C, where C is given by:

$$C = (x_i - x_{i-1})(y_{i+1} - y_{i-1}) - (y_i - y_{i-1})(x_{i+1} - x_{i-1}). \tag{4.2}$$

If the result is 0, then the points are collinear. If the result is positive, the three

vertices constitute a left-turn and vertex i is convex. If the result is negative, then

the three vertices constitute a right-turn and vertex i is concave. The function
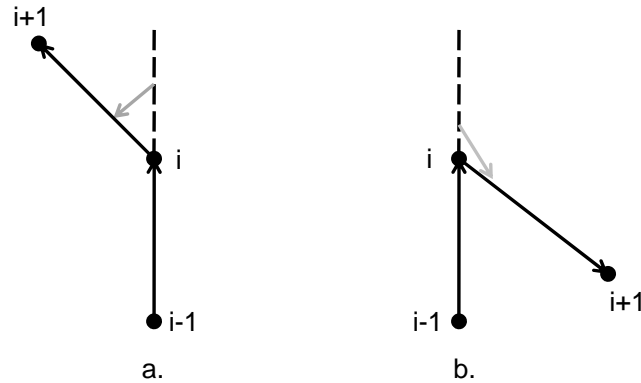
Figure 4.6: Left-handed Turn vs. Right-handed Turn.

ConcaveTest.m used to test for vertex convexity is found in Appendix I.

If a vertex is determined to be convex, then the next vertex is analyzed. If the vertex is concave, then the polygon is concave and the function is terminated. The process continues until a concave vertex has been found or all vertices have been analyzed. The algorithm outlined in Fig. 4.7 is the basis of function *FindConcave.m* used to determine if the polygon is concave or convex. If the polygon is concave, then the function also identifies which is the concave vertex based on the order in which the vertices were analyzed. If the polygon is convex, the vertex labeled as 1 is used as one of the end points of the splitting line segment used to decompose the polygon. The function can be found in Appendix J.

In general, convex polygons are preferred, since all line segments that connect two non-adjacent vertices lie entirely inside the polygon. Therefore, if a concave polygon must be decomposed, it is preferable to select the concave vertex

```
Input:
    Polygon Vertices Array (2 x n)


    For i=1:n
        Evaluate if vertex "i" is convex or concave


        If vertex i is concave
            Set polygon concavity flag to 1
            Set concave vertex to index "i"
            Change flag to end loop


        If vertex i is convex
            Set polygon concavity flag to 0
            Set concave vertex to 0, indicating that none
            of the vertices is concave so far

Output:
    Polygon concavity flag (0 or 1)
    Concave vertex (number between 0 and n)
```

Figure 4.7: *FindConcave.m* **Overview.**

since it is always possible to find a non-adjacent vertex such that the line segment between the two is entirely contained in the polygon. The function *PolySplit.m* then scans all the vertices, searching for a suitable line segment that can be used to divide the polygon into two polygons of lesser order. This is done by finding the intersection point between a line segment and the edges of the polygon. A line segment that satisfies the necessary conditions is one that never intersects any of

the edges of the polygon and is entirely contained within it. The two outputs of this function are the two arrays containing the coordinates of each vertex that define each new polygon. The algorithm is described in Fig. 4.8 and the function can be found in Appendix K.

```
Input:

    Polygon Vertices Array - Poly (2 x n)


    Determine if polygon is concave or convex
    If polygon is convex
        Store vertices 1,2, and 3 into Poly1
        Store vertices 3, 4, …, n, and 1 in Poly2


    If polygon is concave
        Retrieve concave vertex "i"
        Find vertex "j" such that line segment i-j is
        entirely contained within Poly
        if i < j
            Store vertices i,i+1, … and j into Poly1
            Store vertices j, j+1, …, n, 1, 2, …, i in Poly2
        if j < i
            Store vertices j,j+1, … and i into Poly1
            Store vertices i, i+1, …, n, 1, 2, …, j in Poly2

Output:
    2 Polygon Vertices Arrays (Poly1 & Poly2)
```

Figure 4.8: *PolySplit.m* **Overview.**

4.3.3 Decomposition of Triangles into Right-Triangles

The polygon has now been decomposed into triangles. However, $\bar{r}$ has been previously calculated for a right-triangle, by placing a frame at the $90°$ angle and aligning the axes of the frame with the width and the height of the triangle (Appendix C). Therefore, it is necessary to decompose the triangles into right-triangles for which $\bar{r}$ has a known analytical solution. The process is outlined next.

First, select one of the vertices of the triangle. Then calculate the intersection of the perpendicular line from this vertex to the opposite side, the height of the triangle. This will result in one of two possibilities: the triangle has been divided into two right-triangles if the height is entirely contained within the triangle (the base point lies between two vertices), or the height lies outside the triangle (the base point lies outside the two vertices). Each case is shown in Fig. 4.9.

For the case described in Fig. 4.9a, $\bar{r}$ for triangle ABC can be calculated with (4.1):

$$\bar{r}_{ABC} = \frac{\bar{r}_{AHB}A_{AHB} + \bar{r}_{AHC}A_{AHC}}{A_{AHB} + A_{AHC}}. \tag{4.3}$$

In the same way, $\bar{r}$ for triangle DEF in Fig. 4.9b can be calculated by treating triangle DHE as a body and triangle DHF as a hole:

$$\bar{r}_{DEF} = \frac{\bar{r}_{DHE}A_{DHE} - \bar{r}_{DHF}A_{DHF}}{A_{DHE} - A_{DHF}}. \tag{4.4}$$

The function *TriSplit.m* described in Fig. 4.10 takes as input the three vertices of the triangle that must be decomposed into two right-angle triangles. The code
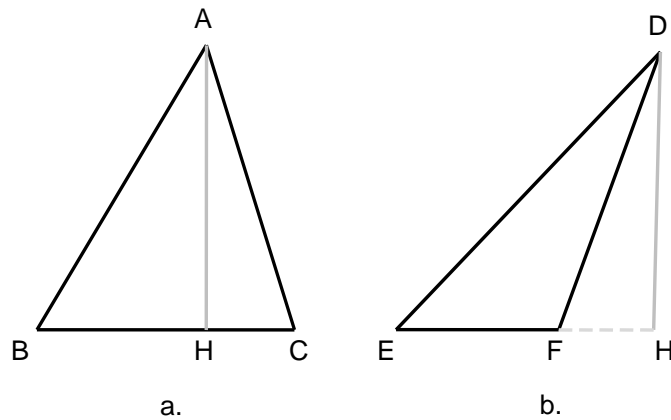
Figure 4.9: Dividing a Triangle into Two Right-triangles.

selects the third vertex and draws the perpendicular from this point to the line

segment defined by the first and second vertex. The intersection of these two lines

is the location of H. Also, the routine identifies whether H lies inside or outside the

triangle by looking at the first and second vertices, as well as H, which must be

collinear. The outputs of the function are the location of the three vertices of the

two resulting right triangles, and a binary variable set to 1 if the two resulting

right-triangles must be added or -1 if they must be subtracted. Effectively, the

first vertex has been replaced by the new point H in the first right-triangle and the

second vertex has been replaced with H in the second right-triangle. The function

can be found in Appendix L.

Knowing the location of the vertices of a right-angle triangle is enough

information to calculate $\bar{r}$ but the necessary parameters are not explicit. The

analytical form of $\bar{r}$ for a right-triangle shown in Appendix C requires that the

```
Input:

Three Triangle Vertices

P₁(x₁,y₁)

P₂(x₂,y₂)

P₃(x₃,y₃)


    Find Eq. of line that passes through P₁ and P₂ (L₁₂)

    Find Eq. of perpendicular line to L₁₂ through P₃ (L₃)

    Find intersection of L₁₂ and L₃ - H(xₕ,yₕ).

    Store 1ˢᵗ Right Triangle: Tri1 = (x₁,y₁,xₕ,yₕ,x₃,y₃)

    Store 2ⁿᵈ Right Triangle: Tri2 = (x₂,y₂,xₕ,yₕ,x₃,y₃)


    Determine if H lies between P₁ and P₂

    if H lies between P1 and P2

        set flag to 1 (add triangles)

    if H does not lie between P1 and P2

        set flag to 0 (subtract triangles)


Output:

Tri1, Tri2, flag
```

Figure 4.10: *TriSplit.m* **Overview.**

body-frame is located at the right-angle, with its axes aligned with the width and the height of the right triangle. The function *FixTri.m* shown in Appendix M is used to determine the orientational discrepancy between the world frame and the body-frame located at the right-angle and whose axes are oriented along the triangle's sides, the width and the height of the triangle and the location of the right-angle. These parameters are used to calculate $\bar{r}$ using *ComputeRbar.m* as

described previously.

The next section shows the results obtained after executing the polygonal decomposition routine and the $\bar{r}$ calculation routine for the complex body geometry introduced at the beginning of this chapter.

4.4   Algorithm Results for an Example Complex Body

Consider the polygonal body with a hole described in Fig. 4.1. This complex body-and-hole is decomposed into right-triangles using the polygonal decomposition routine, *PolyDecomp.m*. The right-triangular sub-bodies and sub-holes resulting from the decomposition procedure are shown in Fig. 4.11. The average distance $\bar{r}$ is calculated for a specific location of the frame of reference in the plane using *ComputeRbar.m*. Figure 4.12 illustrates how $\bar{r}$ varies with the location of the frame of reference in the plane.
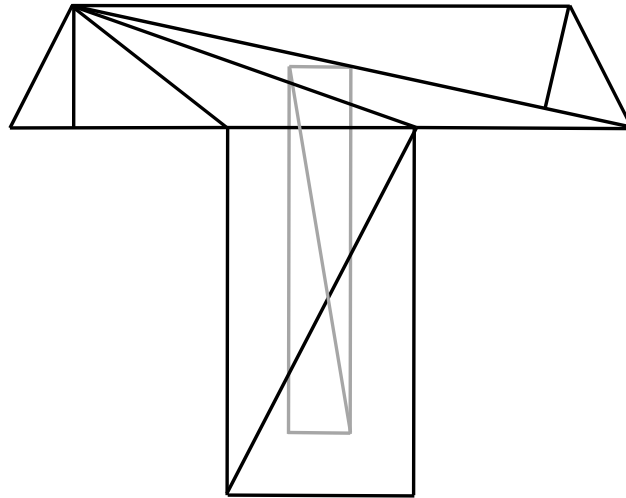
Figure 4.11: Result of Polygonal Decomposition Routine.
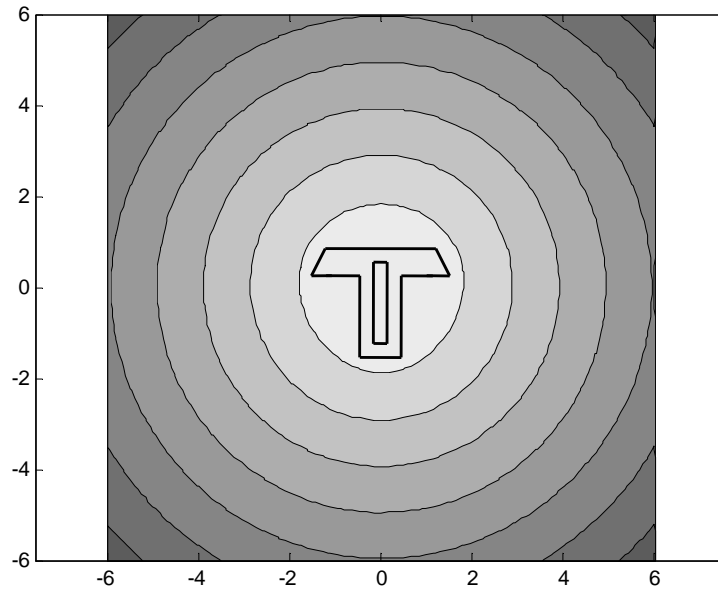
Figure 4.12: Average Distance $\bar{r}$ for a Composite Body with Respect to the Location of z.

4.5   Summary

A means to calculate the average distance from a location in the plane to an arbitrary polygonal body with polygonal and circular holes, $\bar{r}$, was presented in this chapter. The process was divided into two parts. The first part consists of decomposing the original body into simple geometry bodies for which an analytical form of $\bar{r}$ is known for each. This decomposition needs to be performed only once. Then, $\bar{r}$ can be calculated in real time for any number of locations in the plane. A detailed description of each of the subfunctions required to execute both tasks has been presented as well.

5. CONTRIBUTION AND FUTURE WORK

There is currently a need to mathematically characterize the accuracy of different configurations. However, this is a complex issue due to the need to consider rotational and translational error simultaneously. A measurable quality is required to evaluate the distance between an obtained rigid body position and a target configuration. This problem arises frequently in general mechanism design, robot calibration, and motion planning and has been addressed in this paper.

There are two main contributions in this thesis. The first of these contributions is the description of the parameters that correspond to the optimal motion for a rigid body that must undergo a planar displacement. Using the best instantaneous motion measure v_D, it is possible to calculate the optimum value for the body's angular velocity and the location of the frame undergoing a partial or complete geodesic motion. This expression can be used as long as the orientational discrepancy between configurations is not equal to zero. When this is the case, the best motion is pure translation of the body. The optimal instantaneous angular velocity was found to be less than that associated with a complete geodesic and is determined by the magnitude of the orientational displacement. The best instantaneous motion corresponds to a partial geodesic motion of a frame located at the body's geometric center. The best motion measure v_D does not satisfy the mathematical requirements of a metric.

The development of a velocity metric is the other important result of this paper. This metric, identified as v_M, can be used to determine the quality of an arbitrary motion relative to a specified motion. As was the case with the best

motion measure, the metric can be simplified as the average distance from a particular point in the plane to the body. This expression can be used when the angular velocity of the two motions being analyzed is not equal. If the angular velocity for the two motions is the same, an different form of v_M can be used to calculate the quality of one motion relative to the other.

Future work in this area will be directed towards evaluating the quality of rigid body *spatial* motions. Several changes in the approach used to develop the best motion measure and the velocity metric will be required. The first change has to do with the way in which the motion is described. No longer can the Center of Rotation solely be used to describe body configuration. In planar motion, a motion could be described as rotation about a stationary point. In spatial motion, a motion can only be characterized as translation along an axis and a rotation about this axis, also known as a screw motion. Additionally, geodesic motions in space must satisfy an additional set of mathematical requirements that did not factor in planar motion. Also, the calculation of $\bar{r}$ becomes more complex, as body volumes must be considered instead of areas.

REFERENCES

[1] Stéphanos, M., 1883. "Memoire sur la representation des homeographies binaire par des points de l'space avec application a l'etude des rotations spheriques". *Matematische Annalen,* **22**, pp. 299–367.

[2] Park, F. C., 1995. "Distance metrics of the rigid-body motions with applications to mechanism design". *ASME Journal of Mechanical Design,* **117**(1).

[3] Blaschke, W., 1911. "Euklidische kinematik und nichteuklidische geometrie i,ii". *Zeitschrift fur Mathematik und Physik,* **60**, pp. 61–91 and 203–204.

[4] Study, E. "Gemetrie der dynamen". *Teubner Verlag.*

[5] Ravani, B., and Roth, B., 1984. "Mappings of spatial kinematics". *ASME Journal of Mechanisms, Transmissions, and Automation in Design,* **106**(3), pp. 341–347.

[6] Eberharter, J. K., and Ravani, B., 2004. "Local metrics for rigid body displacements". *ASME Journal of Mechanical Design,* **126**(1), pp. 805–812.

[7] Zefran, M., Kumar, V., and Croke, C. B., 1998. "On the generation of smooth three-dimensional rigid body motions". *IEEE Transactions on Robotics and Automation,* **14**(4), pp. 576–589.

[8] Kuffner, J. J., 2004. "Effective sampling and distance metrics for 3d rigid body path planning". In 2004 IEEE International Conference on Robotics and Automation.

[9] Shoemake, K., 1985. "Animating rotation with quaternion curves". In SIGGRAPH '85, pp. 245–254.

[10] Larochelle, P. M., and Murray, A., 2005. "Projection metrics for rigid-body displacements". In ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference.

[11] Lin, Q., and Burdick, J. W., 2000. "Objective and frame-invariant kinematic metric functions for rigid bodies". *International Journal of Robotics Research,* **19**(6).

[12] Kazerounian, K., and Rastegar, J. "Object norms: A class of coordinate and metric independent norms for displacements". *Flexible Mechanisms, Dynamics, and Analysis,* **47**, pp. 271–275.

[13] Martinez, J. M. R., and Duffy, J., 1995. "On the metrics of rigid body displacements for infinite and finite bodies". *ASME Journal of Mechanical Design,* **117**(1), pp. 41–47.

[14] Chirikjian, G. S., and Zhou, S., 1998. "Metrics on motion and deformation of solid models". *ASME Journal of Mechanical Design,* **120**.

[15] Latombe, J. C., 1990. *Robot Motion Planning.* Springer.

[16] Biagiotti, L., and Melchiorri, C., 2008. *Trajectory Planning for Automatic Machines and Robots.* Springer.

[17] Spivak, M., 1965. *Calculus on Manifolds.* HarperCollins.

[18] Jost, J., 2008. *Riemannian Geometry and Geometric Analysis (5th ed.).* Springer-Verlag.

[19] Huang, S., and Schimmels, J. M., 2003. "Sufficient conditions used in admittance selection for planar force-guided assembly". *IEEE Transactions on Robotics and Automation,* **19**(4).

[20] Huang, S., and Schimmels, J. M., 2004. "Admittance selection for force-guided assembly of polygonal parts despite friction". *IEEE Transactions on Robotics,* **20**(5).

[21] Huang, S., and Schimmels, J. M., 2006. "Admittance selection conditions for frictionless force-guided assembly of polyhedral parts in single principle contact". *IEEE Transactions on Robotics and Automation,* **22**(2).

[22] Pilkey, W. D., 2002. *Analysis and Design of Elastic Beams.* John Wiley & Sons.

[23] Rudin, W., 1953. *Principles of Mathematical Analysis.* McGraw-Hill.

[24] MathWorks.com. Matlab central dxf files. http://www.mathworks.com/matlabcentral/fileexchange/term=tagdxf.

[25] Sebastian, 2009. Read dxf file data. http://www.mathworks.com/matlabcentral/fileexchange/24572.

[26] Graham, R. L., 1972. "An efficient algorithm for determining the convex hull of a finite planar set". *Information Processing Letters,* **1**, pp. 132–133.

APPENDIX A
CHARACTERIZATION OF ALL MOTIONS AS GEODESICS

Any rigid-body motion which results in a body particle moving at a known velocity (with known angular velocity as well) can be described in terms of the geodesic (or partial-geodesic) motion for a frame located in the plane. This result is a consequence of the kinematic constraint present in rigid body motion and can be shown as follows.

Let $\mathbf{v_P}$ be the translational velocity of particle P on a rigid body. Associated with the translational velocity for this particle, the body rotates at angular velocity ω. Let C be the location in the plane of the proposed frame that will undergo a geodesic motion. The velocity $\mathbf{v_P}$ can be expressed as follows:

$$
\begin{aligned}
\mathbf{v}_P &= \mathbf{v}_C + \mathbf{v}_{CP} \\
&= \mathbf{v}_C + \omega\hat{\mathbf{k}} \times \mathbf{r}_{CP} \quad\quad\quad (A.1)
\end{aligned}
$$

However, it has been shown that the velocity for a particle moving in a straight line path is given by:

$$
\begin{aligned}
\mathbf{v}_C &= \mathbf{r}_{CC'}/\tau \\
&= \left(\mathbf{R}(\theta) - \mathbf{I}\right)\mathbf{r}_{OC}/\tau \quad\quad\quad (A.2)
\end{aligned}
$$

Combining the last two results, it is possible to express the velocity of particle P

in terms of the location of a frame undergoing a geodesic motion, C:

$$
\begin{aligned}
\mathbf{v}_P &= (\mathbf{R}(\theta) - \mathbf{I})\, \mathbf{r}_{OF}/\tau + \omega \times \mathbf{r}_{FP} \\
&= (\mathbf{R}(\theta) - \mathbf{I})\, \mathbf{r}_{OF}/\tau + \omega \times (\mathbf{r}_{OP} - \mathbf{r}_{OF})
\end{aligned}
\tag{A.3}
$$

Now, solving for $\mathbf{r}_{OC}$:

$$
\mathbf{r}_{OC} = \left(\frac{\mathbf{R}(\theta) - \mathbf{I}}{\tau} - [\omega_\times] \right)^{-1} (\mathbf{v}_P - [\omega_\times]\mathbf{r}_{OP})
\tag{A.4}
$$

Using (A.4), $\mathbf{r}_{OC}$ can be calculated for all possible planar pose discrepancies, except when $\theta = 0$. If the rotational discrepancy is 0, the first matrix of (A.4) is the zero-matrix. It is impossible to calculate the inverse of such a matrix. $\theta = 0$ corresponds to the pure translation problem, for which the best motion is pure translation of the body.

In conclusion, it has been shown that any planar motion involving rotation can be represented as the instantaneous motion associated with the geodesic of a unique frame in the plane.

APPENDIX B

ANALYTICAL FORM OF $\bar{r}$ FOR A RECTANGLE

The calculation for the average particle distance of a rectangle of dimension $l_1 \times l_2$ from a specified location in space (the rotation center) follows. Locate the body frame C describing the rectangle at the vertex so that the x-axis is along edge l_1 and the y-axis is along edge l_2. The average distance of the rectangle from the rotation center is:

$$
\begin{aligned}
\bar{r} \;=\; \frac{1}{l_1 l_2} \times &\left[\frac{1}{3} ac\sqrt{a^2 + c^2} + \frac{1}{6} a^3 \log\left(c + \sqrt{a^2 + c^2}\right) - \frac{1}{3} bc\sqrt{b^2 + c^2} \dots \right. \\
&- \frac{1}{6} b^3 \log\left(c + \sqrt{b^2 + c^2}\right) - \frac{1}{3} ad\sqrt{a^2 + d^2} - \frac{1}{6} a^3 \log\left(d + \sqrt{a^2 + d^2}\right) \dots \\
&+ \frac{1}{3} bd\sqrt{b^2 + d^2} + \frac{1}{6} b^3 \log\left(d + \sqrt{b^2 + d^2}\right) - \frac{1}{6} d^3 \log\left(a + \sqrt{a^2 + d^2}\right) \dots \\
&\left. + \frac{1}{6} c^3 \log\left(a + \sqrt{a^2 + c^2}\right) - \frac{1}{6} d^3 \log\left(b + \sqrt{b^2 + d^2}\right) - \frac{1}{6} c^3 \log\left(b + \sqrt{b^2 + c^2}\right) \right],
\end{aligned}
$$

$$(\text{B.1})$$

where $a = x_{OC}$, $b = x_{OC} + l_1$, $c = y_{OC}$, $d = y_{OC} + l_2$, and $(x_{OC}, y_{OC}) = \mathbf{r}_{OC}$ is the location of the body frame relative to the rotation center.

Figure B.1: Location of the Rectangle Relative to the Rotation Center.

APPENDIX C

ANALYTICAL FORM OF $\bar{r}$ FOR A RIGHT-TRIANGLE

The calculation for the average particle distance of a right-triangle of dimension $l_1 \times l_2$ from a specified location in space (the rotation center) follows. Locate the body frame C describing the rectangle at the right-angle vertex. The frame is oriented so that the x-axis is along edge l_1 and the y-axis along edge l_2. The average distance of the right-triangle from the rotation center is:

$$
\begin{aligned}
\bar{r} \;=\; \frac{1}{6u^3} \times \Bigg[& um^2d^3 \log\left(md + g + w\right) + 2acu\sqrt{a^2 + c^2} - 2adu\sqrt{a^2 + d^2} - muvc^2 \ldots \\
& + g^3 \log\left(\frac{d + dm^2 + mg + wu}{u}\right) - g^3 \log\left(\frac{c + cm^2 + mg + vu}{u}\right) - 2gcvm^2 \ldots \\
& - um^2c^3 \log\left(mc + g + v\right) - muvg^2 + muwg^2 + 2dguw - 2cguv \ldots \\
& + ud^3 \log\left(md + g + w\right) - uc^3 \log\left(mc + g + v\right) + uc^3 \log\left(a + \sqrt{a^2 + c^2}\right) \ldots \\
& + ua^3 \log\left(c + \sqrt{a^2 + c^2}\right) - ud^3 \log\left(a + \sqrt{a^2 + d^2}\right) - ua^3 \log\left(d + \sqrt{a^2 + d^2}\right) \ldots \\
& + 2dguwm^2 - 2acum^2\sqrt{a^2 + c^2} - 2adum^2\sqrt{a^2 + d^2} + uwd^2m^3 \ldots \\
& - um^2a^3 \log\left(d + \sqrt{a^2 + d^2}\right) - um^2d^3 \log\left(a + \sqrt{a^2 + d^2}\right) + muwd^2 \ldots \\
& + um^2c^3 \log\left(a + \sqrt{a^2 + c^2}\right) + um^2a^3 \log\left(c + \sqrt{a^2 + c^2}\right) - uvc^2 2m^3 \Bigg],
\end{aligned}
$$

$$\text{(C.1)}$$

where:

$$ u = \sqrt{1 + m^2} \tag{C.2} $$

$$ v = \sqrt{c^2 + c^2m^2 + 2mcg + g^2} \tag{C.3} $$

$$ w = \sqrt{d^2 + d^2m^2 + 2mdg + g^2} \tag{C.4} $$

and where $a = x_{OC}$, $b = x_{OC} + l_1$, $c = y_{OC}$, $d = y_{OC} + l_2$, $(x_{OC}, y_{OC}) = \mathbf{r}_{OC}$ is th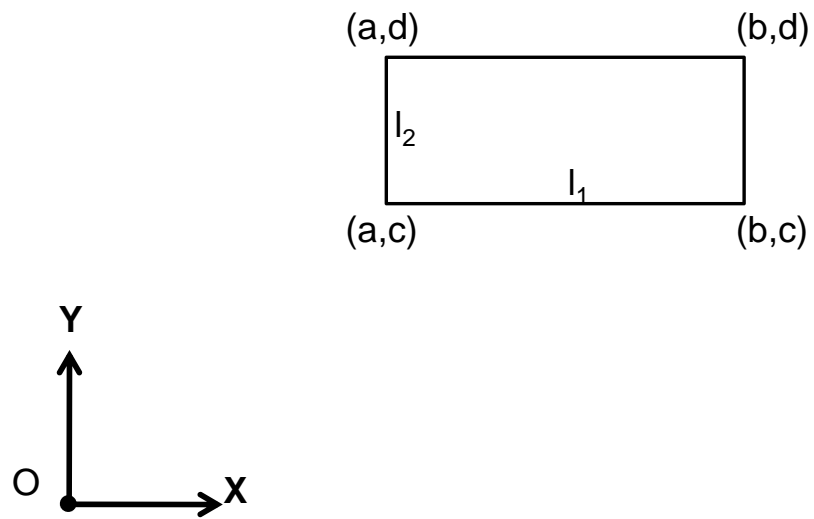e location of the body frame relative to the rotation center, $m = (b - a)/(c - d)$ is the inverse slope of the hypotenuse, and $g = a - md$ is the intersection of the hypothenuse with the x-axis passing through the CoR. The equation of the hypothenuse as seen from the CoR is $x = my + g$.
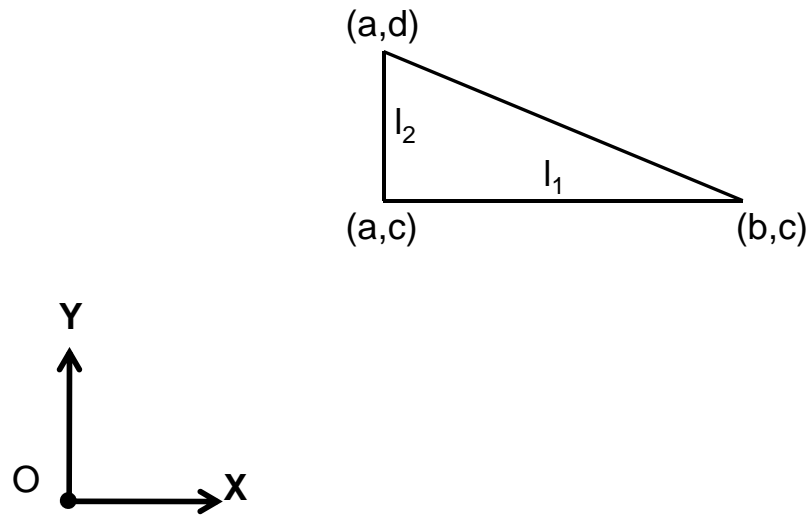
Figure C.1: Location of the Right-Triangle Relative to the Rotation Center.

APPENDIX D
ANALYTICAL FORM OF $\bar{r}$ FOR A CIRCLE

The calculation for the average particle distance of a circle of radius r from a specified location in space (the rotation center) follows. Locate the body frame C describing the rectangle at the center of the circle. The distance from the center of rotation to the center of the circle is L. The average distance of the circle from the rotation center is:

$$
\begin{aligned}
\bar{r} \;=\; & \frac{1}{45r^2(L+r)} \times \Big[13Kr^4L^2 - 2Kr^2L^4 + 6ErL^5 + 34Er^3L^3 + 136Er^5L \\
& + 85Er^4L^2 + 20Er^2L^4 - 8Kr^6 - 3KL^6 + 3EL^6 + 68Er^6 \Big] \cdots \\
& + \frac{1}{2}L^2 \Big[(L-r)(F_2 - F_1) + (L+r)(IncE_2 - IncE_1) \Big] \cdots \\
& - \frac{1}{90(L+r)r^2} \Big[\left(76L^3r^3 - 6L^5r + 35L^4r^2 + 4r^5L + 40r^4L^s + 2r^6 - 3L^6 \right) \times \\
& (IncE_2 - IncE_1) - \left(37L^4r^2 - 38L^2r^4 + 3L^6 - 2r^6 \right)(F_2 - F_1) \Big],
\end{aligned}
$$

$$(D.1)$$

where:

$$K \;=\; \text{EllipticK}(p) \tag{D.2}$$

$$E \;=\; \text{EllipticE}(p) \tag{D.3}$$

$$F_1 \;=\; \text{EllipticF}(\cos 0, p) \tag{D.4}$$

$$F_2 \;=\; \text{EllipticF}(\cos \pi, p) \tag{D.5}$$

$$IncE_1 \;=\; \text{EllipticE}(\cos 0, p) \tag{D.6}$$

$$IncE_2 \;=\; \text{EllipticE}(\cos \pi, p), \tag{D.7}$$

and $p = 2\sqrt{\frac{Lr}{L+r^2}}$.

The Elliptic Integral functions are defined as follows: Incomplete elliptic integral of the first kind:

$$\mathrm{EllipticF}(x, p) = \int_0^x \frac{dt}{\sqrt{(1 - t^2)(1 - pt^2)}} \qquad (\mathrm{D.8})$$

Incomplete elliptic integral of the second kind:

$$\mathrm{EllipticE}(x, p) = \int_0^x \frac{\sqrt{1 - pt^2}}{\sqrt{1 - t^2}} dt \qquad (\mathrm{D.9})$$

Complete elliptic integral of the first kind:

$$\mathrm{EllipticK}(p) = \int_0^1 \frac{dt}{\sqrt{(1 - t^2)(1 - pt^2)}} \qquad (\mathrm{D.10})$$

Complete elliptic integral of the second kind:

$$\mathrm{EllipticE}(p) = \int_0^1 \frac{\sqrt{1 - pt^2}}{\sqrt{1 - t^2}} dt \qquad (\mathrm{D.11})$$

APPENDIX E
READDXF.M

The following is the function *ReadDxf.m* used to import a geometry DXF file into MATLAB. The function prompts the user to select a DXF file and outputs two cell arrays. The first cell array contains all polygon data. The second cell array contains all circle data. This function was generated by modifying a previously developed script [25]. The Copyright disclaimer is observed:

```
Copyright (c) 2009, Sebastian TM
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
```

```matlab
function [Poly,Cir] = ReadDxf()

%% Read file
clear all;
[filename, pathname] = uigetfile('*.dxf','Open
File','Multiselect','off');
% choose file to open
addpath(pathname); % add path to the matlab search path
fId = fopen(filename);
c_ValAsoc = textscan(fId,'%d%s','Delimiter','n');
fclose(fId);
% Code Group Matrix
m_GrCode = c_ValAsoc1;
% Associated value String Cell
c_ValAsoc = c_ValAsoc2;

%% Entities
m_PosCero = find(m_GrCode==0);
%Is searched by (0,SECTION),(2,ENTITIES)
indInSecEnt =
strmatch('ENTITIES',c_ValAsoc(m_PosCero(1:end-1)+1),'exact');
```

```
%(0,ENDSEC)

m_indFinSecEnt =

strmatch('ENDSEC',c_ValAsoc(m_PosCero(indInSecEnt:end)),'exact');

% Entities Position

m_PosCero = m_PosCero(indInSecEnt:indInSecEnt-1+m_indFinSecEnt(1));

% Variable initiation

c_Line = cell(1,2);

c_Poly = cell(1,2);

c_Cir = cell(1,2);

c_Arc = cell(1,2);

c_Poi = cell(1,2);

%

iLine = 1;

iPoly = 1;

iCir = 1;

iArc = 1;

iPoi = 1;

% Loop on the Entities

for iEnt = 1:length(m_PosCero)-2

m_GrCodeEnt = m_GrCode(m_PosCero(iEnt+1):m_PosCero(iEnt+2)-1);

c_ValAsocEnt = c_ValAsoc(m_PosCero(iEnt+1):m_PosCero(iEnt+2)-1);

nomEnt = c_ValAsocEnt1; %c_ValAsocEntm_PosCero(iEnt+1)

%In the entity's name is assumed uppercase

switch nomEnt

case 'LINE'

% (Xi,Yi,Zi,Xj,Yj,Zj) start and end points

c_LineiLine,1 =
```

```
[str2double(f_ValGrCode(10,m_GrCodeEnt,c_ValAsocEnt)),...

str2double(f_ValGrCode(20,m_GrCodeEnt,c_ValAsocEnt)),...

str2double(f_ValGrCode(30,m_GrCodeEnt,c_ValAsocEnt)),...

str2double(f_ValGrCode(11,m_GrCodeEnt,c_ValAsocEnt)),...

str2double(f_ValGrCode(21,m_GrCodeEnt,c_ValAsocEnt)),...

str2double(f_ValGrCode(31,m_GrCodeEnt,c_ValAsocEnt))];

% Layer

c_Line(iLine,2) = f_ValGrCode(8,m_GrCodeEnt,c_ValAsocEnt);

% Color

%if no exist is ByLayer (256)

%c_LineiLine,3 = str2double(f_ValGrCode(62,m_GrCodeEnt,c_ValAsocEnt));

% XData

%c_Line(iLine,4) =

f_XData(GroupCode,'XDataName',m_GrCodeEnt,c_ValAsocEnt);

% Add properties

iLine = iLine+1;

case 'LWPOLYLINE'

% (X,Y) vertexs

%Is not take into account the budge (group code 42, arc in the

polyline).

m_Coord = [str2double(f_ValGrCode(10,m_GrCodeEnt,c_ValAsocEnt)),...

str2double(f_ValGrCode(20,m_GrCodeEnt,c_ValAsocEnt))];

c_PolyiPoly,1 = m_Coord;

% Layer

c_Poly(iPoly,2) = f_ValGrCode(8,m_GrCodeEnt,c_ValAsocEnt);

% Add properties

iPoly = iPoly+1;
```

```matlab
case 'CIRCLE'
% (X Center,Y Center,Radius)
c_CiriCir,1 =
[str2double(f_ValGrCode(10,m_GrCodeEnt,c_ValAsocEnt)),...
str2double(f_ValGrCode(20,m_GrCodeEnt,c_ValAsocEnt)),...
str2double(f_ValGrCode(40,m_GrCodeEnt,c_ValAsocEnt))];
% Layer
c_Cir(iCir,2) = f_ValGrCode(8,m_GrCodeEnt,c_ValAsocEnt);
% Add properties
%
iCir = iCir+1;
case 'ARC'
% (X Center,Y Center,Radius,Start angle,End angle)
c_ArciArc,1 =
[str2double(f_ValGrCode(10,m_GrCodeEnt,c_ValAsocEnt)),...
str2double(f_ValGrCode(20,m_GrCodeEnt,c_ValAsocEnt)),...
str2double(f_ValGrCode(40,m_GrCodeEnt,c_ValAsocEnt)),...
str2double(f_ValGrCode(50,m_GrCodeEnt,c_ValAsocEnt)),...
str2double(f_ValGrCode(51,m_GrCodeEnt,c_ValAsocEnt))];
% Layer
c_Arc(iArc,2) = f_ValGrCode(8,m_GrCodeEnt,c_ValAsocEnt);
% Add properties
iArc = iArc+1;
case 'POINT'
% (X,Y,Z) Position
c_PoiiPoi,1 =
[str2double(f_ValGrCode(10,m_GrCodeEnt,c_ValAsocEnt)),...
```

```
str2double(f_ValGrCode(20,m_GrCodeEnt,c_ValAsocEnt)),...

str2double(f_ValGrCode(30,m_GrCodeEnt,c_ValAsocEnt))];

% Layer

c_Poi(iPoi,2) = f_ValGrCode(8,m_GrCodeEnt,c_ValAsocEnt);

% Add properties

iPoi = iPoi+1;

%case Add Entities

end

end

function c_Val = f_ValGrCode(grCode,m_GrCode,c_ValAsoc)

c_Val = c_ValAsoc(m_GrCode==grCode);

end


function c_XData = f_XData(grCode,XDatNom,m_GrCode,c_ValAsoc)

m_PosXData = find(m_GrCode==1001);

if  isempty(m_PosXData)

indInXData =

m_PosXData(strmatch(upper(XDatNom),c_ValAsoc(m_PosXData),'exact'));

m_indFinXData = find(m_GrCode(indInXData+2:end)==1002)+indInXData+1;

m_IndXData = indInXData+2:m_indFinXData(1)-1;

c_XData =

f_ValGrCode(grCode,m_GrCode(m_IndXData),c_ValAsoc(m_IndXData));

else

c_XData = {[]};

end

end
```

APPENDIX F

POLYDECOMP.M

The following is the routine *PolyDecomp.m* used to decompose an arbitrary geometry (body and holes) into right-triangles and circles.

```
function [BodyTri, HoleTri, HoleCir] = PolyDecomp.m()

% Import geometry file w/. function ReadDxf.m
[PolyCell, CirCell] = ReadDxf();

% Determine which is the external polygon (body)
numpoly = length(PolyCell(:,1));
for i = 1:numpoly
    Temp = cell2mat(PolyCell(i,1));
    order = length(Temp(:,1));
    Sum = 0;
    for j = 1:order
        if (j == 1)
            term = Temp(order,1)*Temp(j,2) - Temp(j,1)*Temp(order,2);
            Sum = Sum + term;
        else
            term = Temp(j-1,1)*Temp(j,2) - Temp(j,1)*Temp(j-1,2);
            Sum = Sum + term;
        end
    end
    Area(i) = Sum/2;
```

```matlab
end

% ``body" is the index of the polygon which is the body
% All the other polygons are holes
[MaxArea, body] = max(Area);

% Initialize Output Matrices
BodyTri = []; HoleTri = []; HoleCir = [];

% Decompose each polygonal body into Right-triangles (body and holes)
for i=1:numpoly
    Polygon = cell2mat(PolyCell(i,1));
    % Use function Triangulation.m to split polygon into triangles
    Tri = Triangulation(Polygon);
    % Number of triangles
    numtri = length(Tri(1,:));
    for j = 1:numtri
        % Divide triangle into two r-triangles w/. function
        % TriSplit.m
        % S = [1x2] matrix, s.t.
        % S(i) = 1 --> Body, S(i) = -1 --> Hole
        [Tri1,Tri2,S] = TriSplit(Tri(:,j));
        % Store info in the following manner w/. function FixTri.m:
        % RTri = [xh,yh,W,H,alpha]';
        RTri1 = FixTri(Tri1);
        RTri2 = FixTri(Tri2);

% Store information as either body or hole
```

```
    if (i == body)

        if (S(1) == 1)

            BodyTri = [BodyTri RTri1];

        else

            HoleTri = [HoleTri RTri1];

        end

        if (S(2) == 1)

            BodyTri = [BodyTri RTri2];

        else

            HoleTri = [HoleTri RTri2];

        end

    else

        if (S(1) == -1)

            BodyTri = [BodyTri RTri1];

        else

            HoleTri = [HoleTri RTri1];

        end

        if (S(2) == -1)

            BodyTri = [BodyTri RTri2];

        else

            HoleTri = [HoleTri RTri2];

        end

    end

    end

end

% Store information for circles (no modifications needed)
```

```matlab
numcir = length(CirCell(:,1));

for i=1:numcir

    % Circle = [xc, yc, radius]';

    Circle = cell2mat(CirCell(i,1));

    if (size(Circle) == 0)

        % No Circles

    else

        HoleCir(:,i) = [Circle]';

    end

end
```

APPENDIX G

COMPUTERBAR.M

The following is the function *ComputeRbar.m*, used to calculate $\bar{r}$ for an arbitrary body.

```
function rbar = ComputeRbar(BodyTri,HoleTri,HoleCir,P)

% Number of right-triangles as bodies
length1 = length(BodyTri(1,:));

% Calculate r-bar for each body
for i = 1:length1
    BodyPtRef = [BodyTri(1,i), BodyTri(2,i)];
    W = BodyTri(3,i);
    H = BodyTri(4,i);
    alpha = BodyTri(5,i);
    rbar1(i) = IntTri(BodyPtRef,P,W,H,alpha);
    Area1(i) = W*H/2;
    rA1(i) = rbar1(i)*Area1(i);
end

% Number of right-triangles as holes
length2 = length(HoleTri(1,:));

% Calculate r-bar and area for each triangular hole
for i = 1:length2
    BodyPtRef = [HoleTri(1,i), HoleTri(2,i)];
```

```
    yh = HoleTri(2,i);

    W = HoleTri(3,i);

    H = HoleTri(4,i);

    alpha = BodyTri(5,i);

    rbar2(i) = IntTri(BodyPtRef,P,W,H,alpha);

    Area2(i) = W*H/2;

    rA2(i) = rbar2(i)*Area2(i);

end


% Number of circles as holes

length3 = length(HoleCir(1,:));


% Calculate r-bar and area for each circular hole

for i = 1:length3

    BodyPtRef = [HoleCir(1,i), HoleCir(2,i)];

    r = BodyTri(3,i);

    rbar3(i) = IntCir(BodyPtRef,P,r);

    Area3(i) = pi*r*r;

    rA3(i) = rbar3(i)*Area3(i);

end


% Computer r-bar for original geometry

rbar =

(sum(rA1)-sum(rA2)-sum(rA3))/(sum(Area1)-sum(Area2)-sum(Area3));
```

APPENDIX H

TRIANGULATION.M

The following is the function *Triangulation.m*, used to decompose or split a polygon into triangles. This function calls *PolySplit.m* to decompose a polygon into two polygons. The process is repeated until all remaining polygons are triangles. The input is a $2 \times n$ matrix containing the original polygon's n vertices and the output is a $6 \times n - 2$ matrix where each column has the $x-$ and $y-$ component information of the 3 vertices of a triangle and $n - 2$ is the total number of triangles.

```
function Tri = Triangulation(Poly)
% Polygon Triangulation Routine

% Polygon order
n = length(Poly(1,:));

% Index is what row of the storage matrix is currently being analyzed
Index = 1;

% Global Storage Matrix:
% counter
a = 1;
% col = vertex index, from 1 to n
for col = 1:n
    Stored(1,a) = Poly(1,col);
    Stored(1,a+1) = Poly(2,col);
```

```
    a = a + 2;
end

% Size = Matrix storing number of vertices of the corresponding
polygons
Size(1) = n;

% START TRIANGULATION ROUTINE

% TriCount is how many triangles have been found
TriCount = 0;

% StCount is what row of the storage matrix the polygon data
% is being added to
StCount = 1;

% Continue decomposing polygons until all polygons are triangles
while (Index <= StCount)

% Create a matrix containing info on the polygon being decomposed
clear Work
a = 1;
for col = 1:Size(Index)
    Work(1,col) = Stored(Index,a);
    Work(2,col) = Stored(Index,a+1);
    a = a+2;
end

% Check if the polygon under consideration has 4 or more vertices
```

```
% If its not a triangle, it must be split
if (Size(Index) > 3)

    % Find concave vertex and decompose into two polygons
    vertex = FindConcave(Work);

    if (vertex == 0)
        % Convex Polygon, any vertex can be used.  Use vertex 1
        vertex = 1;
        [Poly1,Poly2] = PolySplit(Work,vertex);
    else
        % Concave Polygon, use concave vertex
        [Poly1,Poly2] = PolySplit(Work,vertex);
    end

    size1 = length(Poly1(1,:));
    size2 = length(Poly2(1,:));

    % Save Poly1 to Storage Matrix:  Stored
    a = 1;
    for b = 1:size1
        Stored(StCount+1,a) = Poly1(1,b);
        Stored(StCount+1,a+1) = Poly1(2,b);
        a = a + 2;
    end

    % Save Poly 2 to Storage Matrix:  Stored
    a = 1;
```

```
    for b = 1:size2

        Stored(StCount+2,a) = Poly2(1,b);

        Stored(StCount+2,a+1) = Poly2(2,b);

        a = a + 2;

    end


    % Save the order of Poly1 and Poly2 to Size matrix.

    Size(StCount+1) = size1;

    Size(StCount+2) = size2;


    % Two Polygons have been added to list, update counter.

    StCount = StCount + 2;


else

    % The "active" polygon is a triangle, store data into Triangle
matrix

    % Add one to the Triangle Counter

    TriCount = TriCount + 1;

    for a = 1:6

        Tri(a,TriCount) = Stored(Index,a);

    end

end


% Move on to next polygon in list

Index = Index + 1;


end
```

APPENDIX I

CONCAVETEST.M

The following is the function *concavetest.m*, used to determine if the vertex of a polygon is concave or convex. If the vertex is convex, the output flag `concave` is set to 0 (false). If the vertex is concave, then the output variable is set to 1 (true).

```
function concave = ConcaveTest(P)
% P is a 2x3 matrix, where each column of P is the location of a
vertex
% in the plane (x,y)

% Obtain vertices' x- and y- components:
x1 = P(1,1);
y1 = P(2,1);
x2 = P(1,2);
y2 = P(2,2);
x3 = P(1,3);
y3 = P(2,3);

% Determine if vertex angle is concave (1) or convex (0) based on
% direction of cross product
% cross < 0 ==> RH-turn
% cross > 0 ==> LH-turn

cross = (x2-x1)*(y3-y1) - (y2-y1)*(x3-x1);
```

```
if (cross < 0)

    concave = 0;

else

    concave = 1;

end
```

APPENDIX J

FINDCONCAVE.M

The following is the function *FindConcave.m*, used to determine which vertex, if any, is concave. If there are no concave vertices meaning the polygon is convex, the output flag `vertex` is set to 0 (false). If a concave vertex is found, then the output variable is set to the corresponding vertex number according to the numeration established based on clockwise order. This function uses the function `concavetest.m` to evaluate each vertex.

```
function vertex = FindConcave(V)

% This function finds the concave vertex, if any for a n-gon.
% V is the 2 × n matrix with the x- and y- components
% of each vertex.

n = length(V(1,:));

% Counter
b = 0;
% Vertex index
i = 0;

% Test vertex to see if its concave or convex
% Continue till a concave vertex has been found or all vertices have
been
% analyzed
while (b == 0 && i < n)
```

```
    % Look at next vertex
    i = i + 1;
    % Look at three contiguous vertices
    P1 = [Rf(1,i), Rf(2,i)]';
    P2 = [Rf(1,i+1), Rf(2,i+1)]';
    P3 = [Rf(1,i+2), Rf(2,i+2)]';
    P = [P1 P2 P3];
    % Test points for convexity
    b = ConcaveTest1(P);
end

% Output Vertex Index
% Vertex index = 0 if polygon is convex
if (b == 0)
    vertex = 0;
else
    vertex = i;
end
```

APPENDIX K

POLYSPLIT.M

The following is the function *PolySplit.m*, used to decompose or split a polygon into two polygons of lesser order. This function calls both *concavetest.m* and *FindConcave.m* to determine which vertex is used as one of the end-points of the line segment used to split the polygon.

```
function [Poly1,Poly2] = PolySplit(P,i)


% Determine order of original polygon
n = length(P(1,:));


% Matrix that ensure there are no inconsistencies by making
% vertex 0 = n and vertex n+1 = 1
Pf = [P(:,n) P P(:,1)];


% Counter
a = 0;
% Other vertex used to split the polygon
j = 0;

    while (a == 0 && j < n)
        j = j + 1;
        % Determine if vertices b and j are adjacent
        diff = abs(i - j);
        % Counters
        c = 0;
```

```
    d = 0;

% if the edge is not adjacent to vertex, proceed
if (diff > 1)
    X = P(1,i)-P(1,j);
    Y = P(2,i)-P(2,j);
    if (X == 0)
        M = X/Y;
        G = P(1,i) - M*P(2,i);
    else
        M = Y/X;
        G = P(2,i) - M*P(1,i);
    end
    while (c == 0 && d < n)
        d = d + 1;
        xmin = min(Pf(1,d),Pf(1,d+1));
        xmax = max(Pf(1,d),Pf(1,d+1));
        ymin = min(Pf(2,d),Pf(2,d+1));
        ymax = max(Pf(2,d),Pf(2,d+1));
        dy = Pf(2,d+1) - Pf(2,d);
        dx = Pf(1,d+1) - Pf(1,d);

        if (X == 0)
            if (dx == 0)
                m = dx/dy;
                g = Pf(1,i) - m*Pf(2,i);
            elseif (dy == 0)
```

```
                intx = P(1,i);

                inty = Pf(2,i);

            else

                m = dx/dy;

                g = Pf(1,i) - m*Pf(2,i);

                inty = -(M-m)/(G-g);

                intx = inty*M + G;

            end

    else

        if (dx == 0)

            if (Y == 0)

                intx = Pf(1,i);

                inty = P(2,i);

            else

                M = X/Y;

                G = P(1,i) - M*P(2,i);

                m = dx/dy;

                g = Pf(1,i) - m*Pf(2,i);

                inty = -(M-m)/(G-g);

                intx = inty*M + G;

            end

        else

            m = dy/dx;

            g = Pf(2,i) - m*Pf(1,i);

            if (M == m)

            else

                intx = -(G-g)/(M-m);
```

```
                        inty = intx*M + G;

                  end

            end

      end

      if (M == m)

            c = 1;

      else

            if (intx > xmin && intx < xmax)

                  if (inty > ymin && inty < ymax)

                        c = 1;

                  else

                  end

            else

                  % Keep looking

            end

      end

end

      if (c == 0)

            % no intersecting edges were found, this points can be

used

            % as the second end-point

            a = 1;

      else

            % an intersecting edge was found, keep looking

            a = 0;

      end
```

```matlab
        else

            % do nothing, keep looking

        end

    end

    %

----------------------------------------------------------------

    % Store vertices of two resulting polygons

    % counters
    w = 0;
    ww = 0;

    if (j < i)
        for z = 1:n
            if (z < (i+1) && z > (j-1))
                w = w + 1;
                Poly1(1,w) = P(1,z);
                Poly1(2,w) = P(2,z);
            else
            end
            if (z > (i-1) || z < (j+1))
                ww = ww + 1;
                Poly2(1,ww) = P(1,z);
                Poly2(2,ww) = P(2,z);
            else
            end
        end
```

```
else

    for z = 1:n

        if (z < (i+1) || z > (j-1))

            w = w + 1;

            Poly1(1,w) = P(1,z);

            Poly1(2,w) = P(2,z);

        else

        end

        if (z > (i-1) && z < (j+1))

            ww = ww + 1;

            Poly2(1,ww) = P(1,z);

            Poly2(2,ww) = P(2,z);

        else

        end

    end

end
```

APPENDIX L

TRISPLIT.M

The following is the function *TriSplit.m*, used to decompose or split a triangle into two right-triangles. The input is a 6×1 matrix containing the original triangle's n vertices and the output is two matrices and a variables: two 6×1 matrices with the location of vertices that make up each triangle and a flag indicating whether the base of the height is inside or outside the triangle.

```
function [Tri1,Tri2,S] = TriSplit(Tri)

% Triangle vertices

x1 = Tri(1);
y1 = Tri(2);
x2 = Tri(3);
y2 = Tri(4);
x3 = Tri(5);
y3 = Tri(6);

% slopes
if (x1 == x2 || y1 == y2)
    % Do Nothing
else
    m12 = (y2-y1)/(x2-x1);
    m3 = -1/m12;
end
```

```
% y- intersects
if (x1 == x2 || y1 == y2)

    % Do Nothing

else

    b12 = y1 - m12*x1;

    b3 = y3 - m3*x3;

end


% Base of the Height (H)
if (x1 == x2)

    xh = x1;

    yh = y3;

elseif (y1 == y2)

    xh = x3;

    yh = y1;

else

    xh = -(b12 - b3)/(m12 - m3);

    yh = m3*xh + b3;

end


% Point H
Tri1 = [x1, y1, xh, yh, x3, y3]';
Tri2 = [x2, y2, xh, yh, x3, y3]';


% Add or Subtract?
% S = 1:  Add areas
% S = -1:  Subtract areas
a = x2 - x1;
```

```
b = y2 - y1;

c = xh - x1;

d = yh - y1;


d1h = sqrt((x1-xh)*(x1-xh) + (y1-yh)*(y1-yh));

d2h = sqrt((x2-xh)*(x2-xh) + (y2-yh)*(y2-yh));

d3h = sqrt((x3-xh)*(x3-xh) + (y3-yh)*(y3-yh));

A1h = d1h*d3h/2;

A2h = d2h*d3h/2;


if (x2 > x1)

    a = x2 - x1;

    c = xh - x1;

else

    a = x1 - x2;

    c = xh - x2;

end

if (y2 > y1)

    b = y2 - y1;

    d = yh - y1;

else

    b = y1 - y2;

    d = yh - y2;

end


if (y1 == y2)

    if (a > c && c > 0)

        S(1) = 1;
```

```
            S(2) = 1;

        else

            if (A1h > A2h)

                S(1) = 1;

                S(2) = -1;

            else

                S(1) = -1;

                S(2) = 1;

            end

        end

    else

        if (b > d && d > 0)

            S(1) = 1;

            S(2) = 1;

        else

            if (A1h > A2h)

                S(1) = 1;

                S(2) = -1;

            else

                S(1) = -1;

                S(2) = 1;

            end

        end

    end
```

APPENDIX M

FIXTRI.M

The function *FixTri.m* is used to calculate the width, the height and the error in orientation between the world-frame and the body-frame attached to the right-angle and oriented along the sides of the triangle. The input required is the vertices of the right-triangle relative to the world-frame.

```
function [W,H,alpha] = FixTri(Tri)

% Tri = [P1, H, P3] (the right angle must be the second vertex)
% Outputs the angle (in rad) that a triangle must rotate to be
% integrated consistently, as well as the resulting width and height

x1 = Tri(1) - Tri(3);
y1 = Tri(2) - Tri(4);
x3 = Tri(5) - Tri(3);
y3 = Tri(6) - Tri(4);

if (x1 == 0)
    if (y1 > 0)
        theta1d = 90;
    else
        theta1d = 270;
    end
elseif (x1 > 0)
    if (y1 > 0)
```

```
        theta1d = atand(y1/x1);

    else

        theta1d = 360 + atand(y1/x1);

    end

else

    theta1d = 180 + atand(y1/x1);

end

if (x3 == 0)

    if (y3 > 0)

        theta3d = 90;

    else

        theta3d = 270;

    end

elseif (x3 > 0)

    if (y3 > 0)

        theta3d = atand(y3/x3);

    else

        theta3d = 360 + atand(y3/x3);

    end

else

    theta3d = 180 + atand(y3/x3); end

% find the 90 degree angle

m = theta3d - theta1d;

if (y1 == 0)

    if (x1 > 0)
```

```
        if (y3 > 0)

            alpha = 0;

            W = x1;

            H = y3;

        else

            alpha = pi/2;

            W = abs(y3);

            H = x1;

        end

    else

        if (y3 > 0)

            alpha = -pi/2;

            W = y3;

            H = abs(x1);

        else

            alpha = pi;

            W = abs(x1);

            H = abs(y3);

        end

    end

elseif (y3 == 0)

    if (x3 > 0)

        if (y1 > 0)

            alpha = 0;

            W = x3;

            H = y1;

        else
```

```
                alpha = pi/2;

                W = abs(y1);

                H = x3;

            end

        else

            if (y1 > 0)

                alpha = -pi/2;

                W = y1;

                H = abs(x3);

            else

                alpha = pi;

                W = abs(x3);

                H = abs(y1);

            end

        end

    else

        if (m == 90)

            alpha = theta1d*pi/180;

            W = sqrt(x1*x1 + y1*y1);

            H = sqrt(x3*x3 + y3*y3);

        else

            alpha = theta3d*pi/180;

            W = sqrt(x3*x3 + y3*y3);

            H = sqrt(x1*x1 + y1*y1);

        end

    end
```