Admittance Selection for Force Guided Assembly with Optimal Motion

Fernando Rodriguez Anton
Marquette University

ADMITTANCE SELECTION FOR FORCE GUIDED ASSEMBLY WITH OPTIMAL MOTION

by

Fernando Rodriguez Anton, B.S

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 2013

Dedicated to my family, for all their support they have given me.

ABSTRACT
ADMITTANCE SELECTION FOR FORCE GUIDED ASSEMBLY WITH OPTIMAL MOTION

Fernando Rodriguez Anton

Marquette University, 2013

Current robots lack the precise relative positioning necessary to complete automatic assembly tasks. Several solutions have been proposed. Some approaches use complex vision and force sensing systems to generate corrective motion if misalignment is present in the assembly task. Other solutions rely on generating elastic behavior, known as compliance, between the end effector and the held movable part. This compliant mechanism helps guide the movable part of the assembly into its proper position.

The project focuses on designing a process by which passive compliant systems can achieve successful assembly for a range of misalignment and generate error-reducing motion that is considered of high quality. This is accomplished by using a velocity metric as the goal of a constrained optimization. The metric uses the average discrepancy of all the particle motion from an established "best motion". This motion minimizes the discrepancy in the velocity of all particles motion from their ideal motion towards their proper position. This procedure identifies the best worst case scenario for a representative set of configurations.

The results obtained for optimization over polygonal geometries of 3, 4, and 5 vertices, demonstrate the effectiveness of the procedure in designing passive compliant behavior resulting in high quality error-reducing motion. Results also show that high quality motion is not only achieved for a set of finite configurations but also for all intermediate ones.

ACKNOWLEDGEMENTS

Fernando Rodriguez Anton

I would like to acknowledge several people for their significant contributions to this project. First of all, my advisor, Dr. Joseph Schimmels, for all his guidance during my time at Marquette University and his willingness to teach me that which I did not understand. Dr. Mark Nagurka for his support and the opportunities he has provided me with to become a better engineer. I would also like to thank Dr. Shuguang Huang for teaching me about rigid body dynamics which were essential for the development of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

$\mathbf{v}$	velocity
$\mathbf{v_0}$	translational nominal velocity
$\mathbf{v_i}$	translational ideal velocity
ω_0	rotational nominal velocity
ω_i	rotational ideal velocity
$\mathbf{t}$	twist
$\mathbf{t_0}$	nominal twist
$\mathbf{t_i}$	ideal twist
$\mathbf{w}$,$\mathbf{W}$	contact wrench (single-point, multiple-point)
$\mathbf{w_t}$,$\mathbf{W_t}$	tangential contact wrench (single-point, multiple-point)
$\mathbf{w_n}$,$\mathbf{W_n}$	normal contact wrench (single-point, multiple-point)
$\mathbf{A}$	3 by 3 Admittance Matrix
$a_{(i,j)}$	(i,j) element of the Admittance Matrix
ϕ,$\boldsymbol{\phi}$	magnitude of contact force (single-point, multiple-point)
V_m	Quality of Motion when compared against the ideal velocity

1 INTRODUCTION

This thesis presents a process by which compliant passive behavior can be selected to assist the automatic assembly of different planar geometries. This process guarantees successful assembly and high quality motion.

This chapter provides background information regarding force-guided assembly and motivates this research project. First, the motivation for the project will be presented. Next, the state of the art and previous strategies will be presented. Important terms used in the research project will be defined. Finally, the strategy employed to identify the best compliance to accomplish successful assembly will be discussed.

1.1 Project Motivation

Current industrial robots have high repeatability but lack the relative positioning necessary for most assembly tasks. This can be attributed to several factors such as encoder resolution, misalignment in the fixturing of the parts, and user error. Most robots do not have the ability to determine when an error has occurred and even fewer have the capability of automatically correcting their path. These errors can be costly; not only can the parts being assembled be damaged but if contact forces are large enough, can cause significant damage to the robot. This can result in costly delays to an assembly line, requiring either robot repair, human operator intervention, or both.

1.2 State of the Art

Several approaches have been proposed to accomplish assembly tasks with a robotic system. Much of the early work addressed peg in hole assemblies [24]. The developed strategies and systems can be broken down into two main categories, passive and active systems.

Passive control strategies refer to those which rely on generating elastic behavior to guide the part to its assembled position. Gross positioning assumes no relative position-

ing errors; and fine positioning is done without any external actuation taking place. One of the earliest and most well known approaches was developed at MIT [24], and is known as the remote center of compliance (RCC). The RCC device generates a specific type of elastic behavior at the end effector of the robot. This type of elastic behavior locates the center of compliance (point at which decoupling of forces and torques takes place) at the bottom of the peg to be assembled. The center of compliance is the location at which the translation and rotation mapping between forces and motions becomes decoupled, i.e, where a translational force does not cause rotational motion, and a moment does not cause translation. The contact forces are used to generate motion towards the properly assembled position. A physical realization of this type system is shown in Figure 1.1.

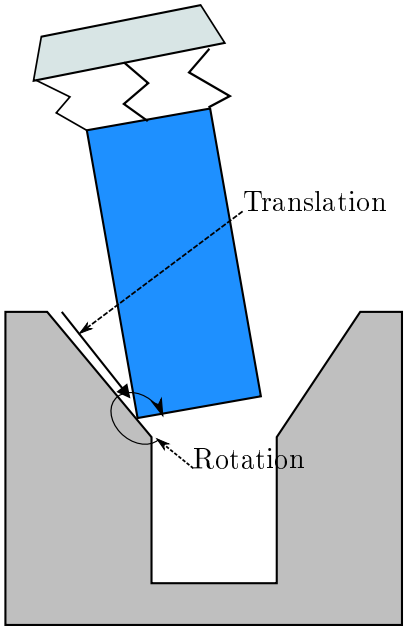


Figure 1.1: Example Compliant Wrist. The contact forces experienced by the moving part are transformed into motion in the correct direction.

Other similar approaches have been proposed. In [9] a passive system was proposed that relied on a RCC device capable of vertical and horizontal assembly. In [13] and [6] systems capable of changing the position of the compliant behavior were designed. Passive systems for more complex assemblies have also been developed [23] [22]. In [1] it was suggested that the use of vibrations combined with a passive compliance mechanism to could be used to accomplish assembly.

While passive compliant based approaches such as those relaying on RCC have been successfully implemented in an industrial environment and are sold as industrial products [4], they do not guarantee successful assembly of the parts for any configuration

that might occur within the range of misalignment and are limited to a certain type of geometry. This results in compliant systems that generate a corrective motion and likely not an ideal corrective motion. A system capable of changing its compliance allowing for the assembly of different geometries has also been presented in [13] [6]; however it does not determine what the "best" compliant system is for a given geometry or guarantee successful assembly.

Active approaches do require external actuation and a control loop to reduce the misalignment present in the assembly. Some approaches use a force sensor which detects the contacts experienced by the part and determines correct motion by modifying the compliance via either a mechanism or electronically [3] [12] [7] [20].

Not all active approaches need the use of a force sensor. Some systems use complex imaging techniques to determine the manner in which the parts are coming into contact. The contact information is used to generate error reducing motion within a control loop [2] [29]. Moreover, there are approaches that combine both visual and force sensors along with compliance to achieve successful assembly [11] [26] [28].

Approaches relying on force and visual sensors are complex and expensive. Furthermore these systems are still limited by other factors such as encoder resolution. Visual sensors also require line of sight of the parts being assembled which cannot always be obtained.

Work has been done at Marquette University to address the shortcomings of these approaches by generating a process by which passive compliant mechanisms can be designed. These compliant mechanisms result in error-reducing motion with close to ideal motion for certain configurations. These mechanisms ensure that contact forces generate motion towards the successfully assembled position.

Huang [10] identified sufficient conditions that ensure force guidance for single and two point contacts for planar assembly. The sufficient conditions guarantee that, for any configuration, the motion of the assembled parts will be towards the successful assembly position. These conditions are imposed on a finite number of possible configurations so that when they are satisfied for this subset, they are satisfied for all configurations. The finite number of configurations are the boundaries of possible misalignment, the extremal configurations.

In [17] a strategy necessary to identify the configuration extremals necessary for automatic and reliable constraint implementation was created. By using a growth function, which determines the amount of penetration between two polygons [16], the program was capable of generating a list of all possible types of forms of contact and their respective extremal configurations within a given misalignment. It is important to note that this approach is a combination of different approaches to generate a list of all possible forms of contact [18]. Previous work either required the user to submit seed points of what the forms of contact could be, and then determine the contact states from that point [27] or required an evaluation of all possible forms of contact using a numerical optimization [8] which is expensive. In [14], an approach was described which constrains the forms of contact within a range of misalignment; however it requires significant effort to prevent the part from falling into a local minima when identifying feasible forms of contact. The way by which to automatically generate the constraints on the extremals remained difficult. By combining both [10] and [17] generation of compliant systems resulting in successful assembly can be accomplished with relative ease.

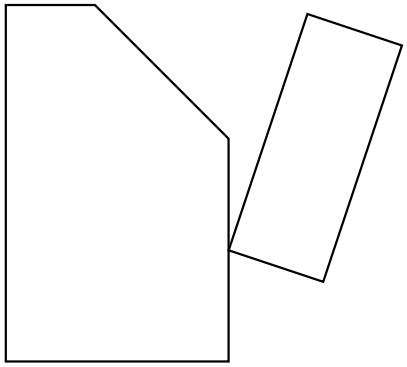


Figure 1.2: Example Contact State. This is defined as an Edge-Vertex contact state, one of many possible forms of contact.

Wiemer [25] created a program capable of generating error reducing compliant systems for the planar case, and identify the best compliance matrix. The best compliance was selected to be that which is capable of tolerating the highest coefficient of friction that still ensures error reduction. Successful compliant systems for triangular, rectangular and stake shaped pegs where identified. The program automatically identified the extremals of the assembly task and automatically generated constraints for them. This program identified systems capable of tolerating coefficients of friction as high as 0.8. However, the only guarantee on the quality of the resulting motion is that the motion

of selected part features moved towards the correct assembled position. There are no guarantees on the rate of error reduction. It was also found that in some situations the obtained compliant systems where extremely close to violating the error reducing constraints. The use of the friction coefficient as the objective function for the optimization also presents problems as the obtained system might satisfy the constraints at a high coefficient of friction but might only marginally satisfy the constraints for assemblies at a lower value. For the purposes of being implemented in an industrial environment, a system guaranteeing high quality motion for a selected friction coefficient would be better suited.

The concept of developing a manner by which to judge the quality of rigid body motion is not straight forward. In [5] an average particle velocity metric was suggested. This metric compares the motion of all particles on a body relative to a selected motion. Ideal constrained and unconstrained motions for a given configuration were also identified. Using these motions, the quality of any motion relative to an ideal one can be determined. Using this metric, compliant systems resulting in high quality motion for certain configurations can be generated.

1.3 Project Objective

The purpose of this project is to create a process by which successful force-guided assembly can be obtained. The resulting compliance will provide error-reducing motion for all possible configurations within the range of misalignment of the robot used for assembly. The compliance will also provide close to optimal motion for a number of selected configurations representative of the assembly task. These configurations represent the extremal configurations of the assembly tasks. The optimal motion is obtained by modifying the previously developed process by using the quality of the motion as the objective function instead of the tolerable friction coefficients. The following sections will explain and define the important terms used in this project.

1.4 Notation and Terminology

1.4.1 Force-Guided Assembly

Force-guided assembly is the process by which contact forces are used to generate a beneficial change in the motion. Force-assembly is described as a special type of force guided assembly for which for each possible misalignment within a range and each possible force at each misalignment, a control law with constant compliance will result in a motion that reduces the misalignment instantaneously [19].

1.4.2 Twists and Wrenches

Because the assembled parts are assumed to not be deformed by contact, part motion is described in terms of rigid-body motion. In this project the motions and contact forces are expressed in screw notation [15]. The motion of the part is written as a twist; and the force and torque as a wrench. These are based on the concept that any spatial rigid-body movement can be expressed as a motion along and about an axis with a given pitch, with the pitch being the ratio of translation to rotation.

A twist $\mathbf{t}$ identifies an angular velocity about an axis and a translation along that same axis. Points further away from the axis have a greater translational velocity.

A wrench $\mathbf{w}$ identifies a generalized force and torque acting at a given point. It contains a translational component (pure force) and an angular component (pure moment). Every wrench applied to a rigid body is equivalent to a force applied along a fixed axis plus a pure couple about the same axis [15].

It is important to note that the description of both twists and wrenches depend on the coordinate frame at which they are defined [15].

In order to transform a twist or wrench into a different frame a screw transformation is used. A frame transformation consisting of translation without rotation can be obtained

using Equation 1.1.

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -y & x & 1 \end{pmatrix} \tag{1.1}$$

1.4.3 Control Law

An admittance control law relates the contact forces to the motion of the part to be assembled. Admittance can be defined as the frequency-dependent generalization of compliance. It incorporates not only stiffness, but also inertia and damping into it. The admittance control law for planar motion selected for use is:

$$\mathbf{t} = \mathbf{t_o} + \mathbf{AW}\phi, \tag{1.2}$$

where $\mathbf{t}$ corresponds to the resulting motion, expressed as a 3-vector twist, in which the first 2 components correspond to translational velocity and the final one to the rotational velocity. The term $\mathbf{t_o}$ is the nominal twist, also expressed as a 3-vector. $\mathbf{A}$ is a 3 by 3 matrix which corresponds to a linear admittance. $\mathbf{W}$ is either a 3 by 2 matrix or a 3-vector containing information about the contact forces, expressed as a wrench at each point of contact. Finally ϕ is either a scalar or a 2-vector corresponding to the magnitude of the force experienced at each point of contact. The contact wrench depends on the geometry and configuration of the parts and the coefficient of friction between the parts to be assembled.

In order to be realized passively, the admittance matrix $\mathbf{A}$ must be symmetric [10]; therefore it has to have the following form:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} \tag{1.3}$$

Where each component relates wrenches to twists in a given direction.

- a_{11} relates force in the x direction to velocity in the x direction

- a_{22} relates force in the y direction to velocity in the y direction
- a_{33} relates torque to rotational velocity
- a_{12} relates force in the y direction to velocity in the x direction, and force in the x direction to velocity in the y direction
- a_{13} relates torque to velocity in the x direction, and force in the x direction to rotational velocity
- a_{23} relates torque to velocity in the y direction, and force in the y direction to rotational velocity

The contribution from each of the components of the admittance matrix is apparent when expanding Equation 1.2

$$\mathbf{t} = \begin{pmatrix} V_{0x} + a_{13}\tau_z + a_{11}F_x + a_{12}F_y \\ V_{0y} + a_{23}\tau_z + a_{12}F_x + a_{22}F_y \\ a_{33}\tau_z + a_{13}F_x + a_{23}F_y \end{pmatrix} \tag{1.4}$$

where F_x refers to force along the x axis, F_y to force along the y axis, τ_z is the torque about the z axis, and t_0 corresponds to the nominal motion. It is clear that each component executes a transformation of a force/torque in a given direction to a motion along either the same direction or another one.

The selected admittance matrix must also be positive definite to guarantee that it can be realized passively. In order for a matrix to be positive definite it must satisfy one of the following equivalent requirements:

- All its principal minors are positive
- All its eigenvalues are positive

1.4.4 Compliance Center

As described previously, a compliant center is the point in space at which the translational and rotational components of the admittance become decoupled. This means that applying a translational force does not generate a rotational motion and vice versa.

Figure 1.3 illustrates this concept. As with the description of twists and wrenches, the description of the compliant center depends on the frame on which it is expressed. It

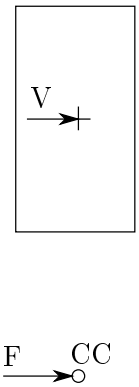


Figure 1.3: The Compliant Center. The point (CC) at which the application of a force yields pure translation (no rotation) of the body.

can be proven that the location of the compliant center for a planar admittance matrix expressed in a given frame can be obtained by the values of $-a_{23}/a_{33}$ and a_{13}/a_{33} for its x and y location respectively relative to the specific frame. If the admittance matrix $\mathbf{A}$ is transformed to the location of the compliant center as previously defined, the forces and torques become decoupled.

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a_{13}/a_{33} & -a_{23}/a_{33} & 1 \end{pmatrix} \tag{1.5}$$

$$\mathbf{A}^* = \mathbf{T}^\mathbf{T}\mathbf{A}\mathbf{T} = \begin{pmatrix} a_{11} - \frac{a_{13}^2}{a_{33}} & a_{12} - \frac{a_{13}a_{23}}{a_{33}} & 0 \\ a_{12} - \frac{a_{13}a_{23}}{a_{33}} & a_{22} - \frac{a_{13}^2}{a_{33}} & 0 \\ 0 & 0 & a_{33} \end{pmatrix} \tag{1.6}$$

Where $\mathbf{T}$ is the screw transformation matrix, and $\mathbf{A}^*$ is the transformed admittance matrix. Using this expression of the admittance for in Equation 1.2, the expression for the resulting motion $\mathbf{t}$ becomes the following:

$$\mathbf{t} = \begin{pmatrix} V_{0x} + F_x \left(a_{11} - \frac{{a_{13}}^2}{a_{33}}\right) + F_y \left(a_{12} - \frac{a_{13}\,a_{23}}{a_{33}}\right) \\ V_{0y} + F_y \left(a_{22} - \frac{{a_{23}}^2}{a_{33}}\right) + F_x \left(a_{12} - \frac{a_{13}\,a_{23}}{a_{33}}\right) \\ a_{33}\,\tau_z \end{pmatrix} \tag{1.7}$$

In this expression it can clearly be observed that translational and rotational components of the admittance become decoupled. The translational motion is only dependent on the force being applied to the system, and the rotational motion only depends on the moment being applied.

1.4.5 Contact State

Contact states are defined as the way in which the different features of the two parts to be assembled (fixed and moving) can contact each other. For planar polygonal bodies, these are the manner in which the vertices and edges of the two parts can come into contact. For all the planar parts these can be further broken into two types: single and two point contact states. All single point contact states are divided into <edge, vertex>, and <vertex, edge> contact states. For simplicity these will be referred to as <E-V> and <V-E> respectively. The possible two point contacts are: <edge-vertex,vertex-edge>, <vertex-edge,edge-vertex>, <edge-vertex,edge-vertex>, <vertex-edge,vertex-edge> and <edge-edge> contact states. These are abbreviated as <E-V,V-E>, <E-V, E-V>, <V-E, V-E> and <E-E> respectively. Since the order of reference is not important the <vertex-edge,edge-vertex> contact state is the same as the <edge-vertex,vertex-edge> and as such is abbreviated as <E-V,V-E>.

Each contact state has allowable motion that maintains the contact state. In other words, for each contact state there is an infinite number of *contact configurations* as can be seen in Figure 1.4. Contact variation within each contact state can be described with a reduced number of variables. Each type of single point contact (<V-E>, or <E-V>) possesses 2 degrees of freedom, hence all configurations within the contact state can be expressed as a function of a translation and an orientation. In this project, δ is used for translation, and θ for orientation, as shown in Figure 1.5.

Two point contact states for planar motion have only one degree of freedom, a rotation about a point. However for consistency in the calculations they are also expressed as being dependent on two independent variables, a displacement δ and an angle θ.

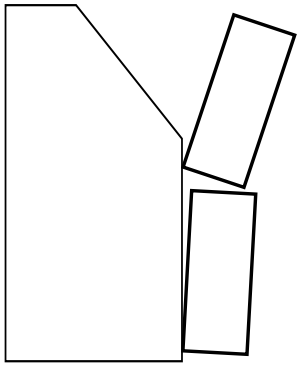


Figure 1.4: Multiple Configurations within a Edge-Vertex Contact State. Each configuration corresponds to a different relative position or orientation of the parts.

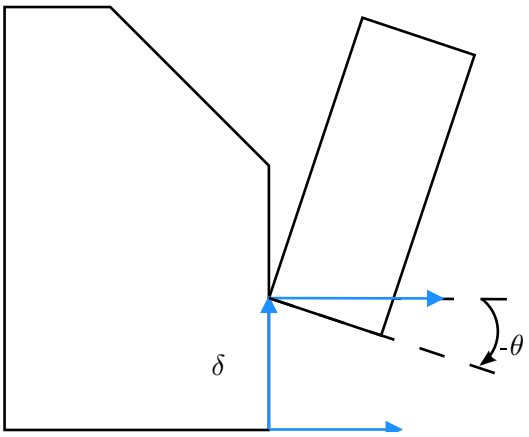


Figure 1.5: Variables Used to Determine Configuration of a Contact State. δ determines translation and θ corresponds to orientation.

1.4.6 Extremals

Within each contact state there is a maximum and minimum value for the two variables describing the configuration. These extremals identify the range of possible configurations within a given contact state. It is important to identify these locations in the process to guarantee successful force guided-assembly with error reduction [10]. These extremals define a rectangular space of all possible configurations for a contact state and are used to define constraints that guarantee successful assembly for intermediate configurations. The bounds on these configurations are the maximum values for rotation and translation within a contact state. All possible configurations within a contact state are bounded conservatively by its extremals.

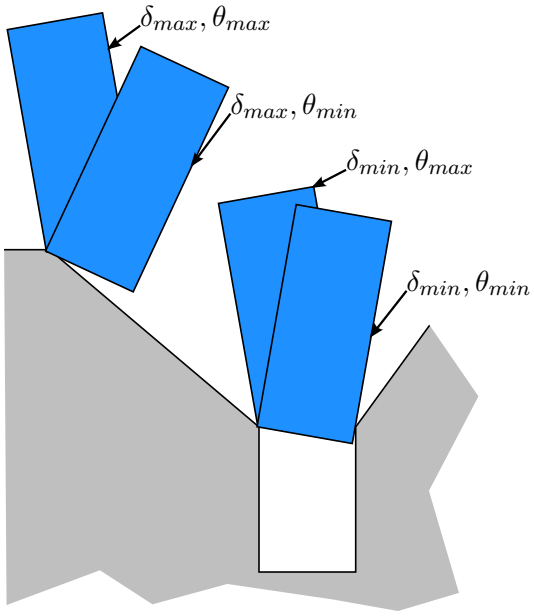


Figure 1.6: E-V Contact State Extremals for Constraint Implementation. Each configuration corresponds to maximum or minimum translation and orientation.

1.5 Shapes Investigated

The shapes investigated were chosen to be the same as those used by Wiemer [25]. These shapes are chosen such that they are:

- Planar
- Convex (movable part)
- Symmetric about the vertical axis
- Assemblable in plane (widest at top)
- Contain few vertices (3,4 and 5)

The peg geometries chosen are a rectangle, triangle and stake as shown in Figure 1.7.

1.6 Overview

This thesis presents the means of identifying the best compliant behavior to achieve force-guided assembly despite misalignment in part relative positioning. The compliant behavior yields the best motion towards the assembled position for a widely sampled subset of part configurations. Important terms and concepts related to force-guided

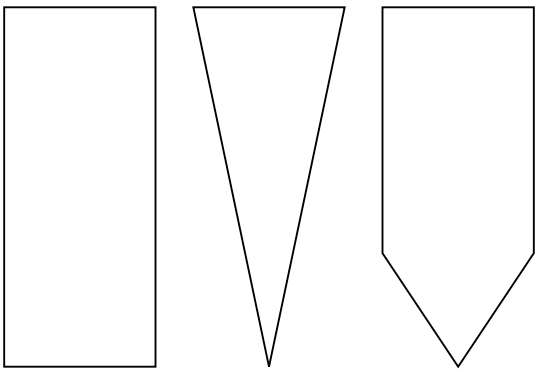


Figure 1.7: Shapes investigated. These include rectangular, triangular and stake shaped pegs.

assembly were explained and the approach taken to identify the compliance matrices presented. The following chapter will provide a more in-depth explanation of the metric and process used to measure the quality of the resulting motions. Chapters 3 through 5 present the results for triangular, rectangular and stake shaped pegs. Chapter 6 presents a numerical investigation of the effectiveness of the approach. Chapter 7 presents the contribution of the project along with recommendations for further study.

2 ADMITTANCE SELECTION FOR OPTIMAL MOTION

As stated previously, the objective of this work is to identify procedures for admittance selection that guarantee successful assembly for all possible configurations within the given bounds of an assembly task while providing a motion close to the optimal one. In order to obtain optimal motion, a manner by which to judge the quality of motions has to be used. This chapter presents the strategy used to apply a velocity metric to the admittance design process.

The following sections will present the strategy used to obtain an admittance that ensures high quality force-guided assembly. The strategy for admittance selection being implemented will be shown along with discussion of some of the critical aspects. This section also presents what is considered the best unconstrained and constrained motion for a given configuration. The best unconstrained motion is used as the basis for the evaluation of motion quality.

2.1 Strategy for Matrix Selection Based on Best Available Motion

This section presents the process to obtain the admittance matrix that provides force guidance with optimal motion for a given assembly task. The process can be divided into three main components. The first component follows the process outlined in [25], identifying all the possible contacts that can occur within the bounds of the assembly task. The second component identifies the contact state extremals. Two types of contact state extremals are used. One type is used to determine constraints that guarantee successful assembly for all configurations in the misalignment range. The second type provides a widely ranging sample of configurations used to evaluate the quality of the resulting motion to provide a measure of the effectiveness of the selected admittance behavior. The final component deals with optimizing the admittance for the best worst-case scenario for the resulting motion of all configurations considered. Figure 2.1 presents a high level overview of the process.

The following sections follow the order presented in Figure 2.1, describing each of the main components in detail.

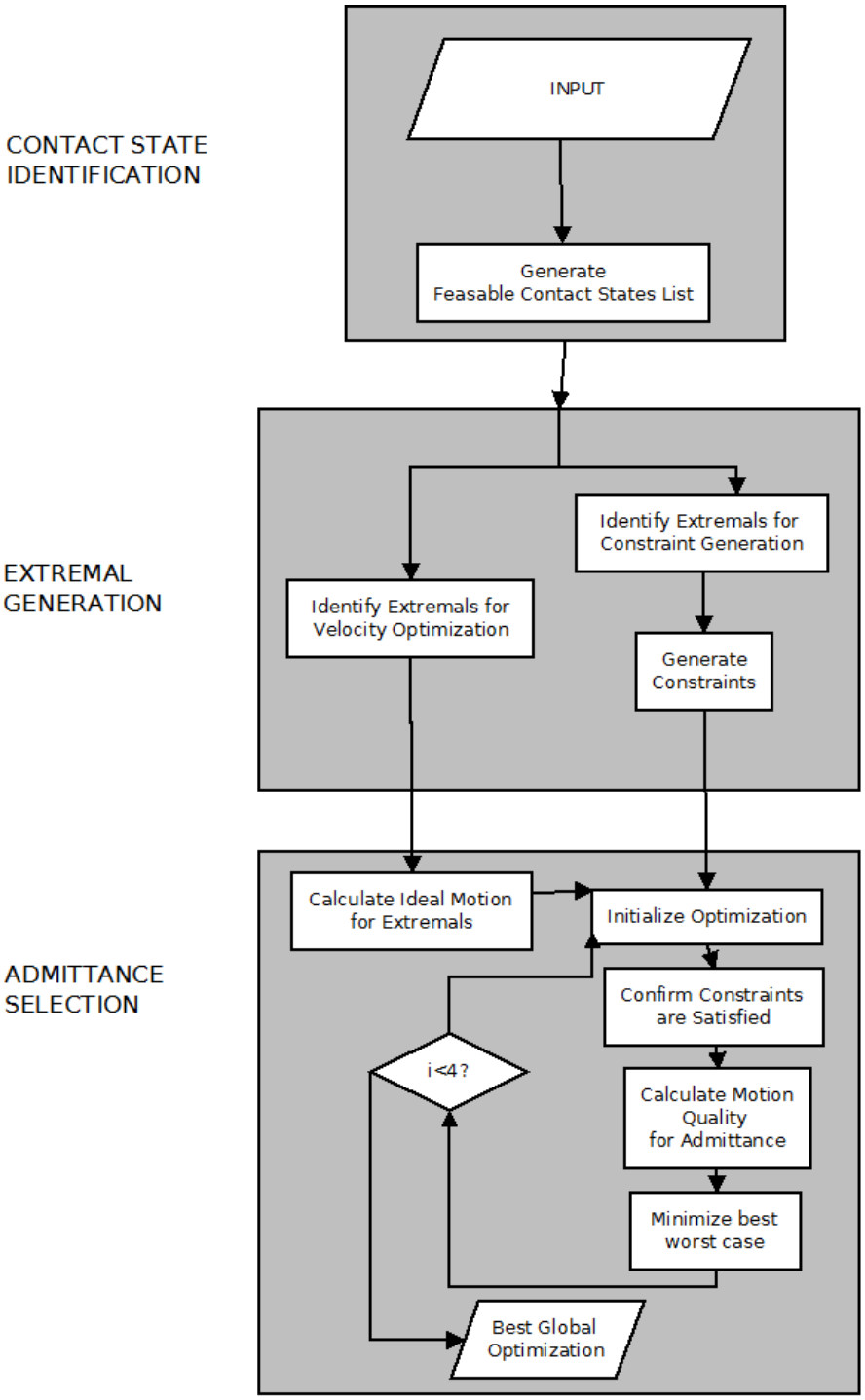


Figure 2.1: Program Flowchart for Velocity Metric Based Optimization. Program is divided into three main components: 1) Contact State Identification, 2)Extremal Generation, and 3) Admittance Selection.

2.2 Contact Identification

In the first component of the process, the contact states possible within the assembly task are identified. The component requires the geometry of the parts, the misalignment bounds of the robot, and the static of coefficient of friction for the assembly task. The program first identifies the set of single point contacts that are considered feasible within the prescribed misalignment bounds. This is accomplished using the process outlined in [17]. This requires the use of a genetic algorithm (GA) to identify which contacts can occur, using a growth function. The growth function provides the expansion or contraction that the parts required in order to be in contact without penetration.

Using the identified single point contact states, a list of two point contacts is generated. A process similar to that for single point contacts is used, resulting in a list of feasible two point contact states. Once this component of the process is complete, a high-level description of all the possible combinations of part features in of contact is obtained.

2.3 Extremal Generation

In this component of the process the contact state information is used to generate the extremals used for both the error-reduction constraints and the velocity quality optimization.

2.3.1 Extremals for Constraint Generation

This part of the process generates the extremals from which the error-reducing constraints are generated. The extremals are obtained using a process similar to contact state identification. A genetic algorithm and a growth function are used to identify the maximum and minimum ranges on the δ and θ parameters, defining a rectangular are with range $\{\theta_{min}, \theta_{max}\}$ and $\{\delta_{min}, \delta_{max}\}$. As Figure 2.2 shows it is possible for the edges of this range to result in penetration between the parts being assembled. As found in [25], conditions imposed on the V-E based contact states resulted in extremely conservative sufficient conditions on the admittance. In order to make these conditions less

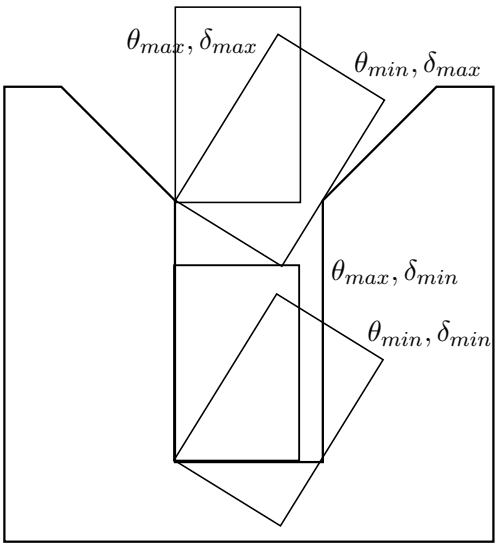


Figure 2.2: Example Extremal Configurations for Constraint Generation. Penetration between the parts can occur for certain configurations.

conservative the V-E contact states are decomposed into smaller sections along the δ parameter. These decompositions then are evaluated for their maximum and minimum values for their orientation θ. This results in less conservative sufficient conditions for error reduction. Once all contact states and their decompositions have their associated extremals identified, the conditions in [10] are applied, yielding the constraints to be used in the velocity optimization process.

2.3.2 Extremals for Velocity Optimization

Identifying the admittance that results in optimal motion requires the determination of the quality of the motion of a representative set of configurations. In this case this set is defined as the achievable configurations within a contact state.

From the previous section it is apparent that using the extremals as the corners of the rectangular area defined by the range $\{\theta_{min}, \theta_{max}\}$ and $\{\delta_{min}, \delta_{max}\}$ can result in configurations that result in part penetration and therefore cannot occur during an assembly task. For the purpose of the velocity metric optimization use of these extremals leads to incorrect results. Because the optimization requires the use of contact states representative of the assembly task, contact states such as those present a significant problem. The optimization tries to minimize the discrepancy between the motion of the selected set of configurations. The inclusion of an unachievable configuration has a

significant impact, dominating the results of the optimization routine. These extremals also result in configurations that could be considered successful assembly. An example of these configurations can be seen in Figures 2.3 and 2.4.

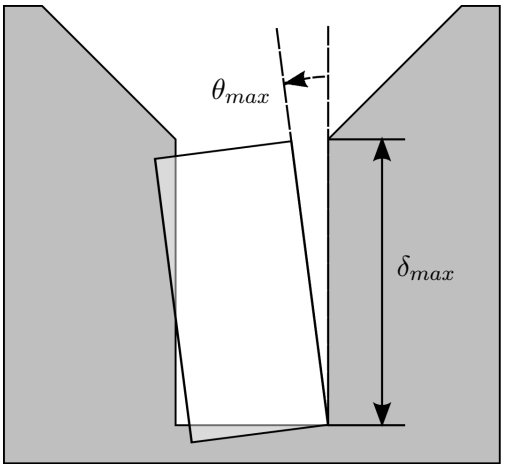


Figure 2.3: Example Impossible Extremal. Both corners of the range result in penetration into the fixed part.

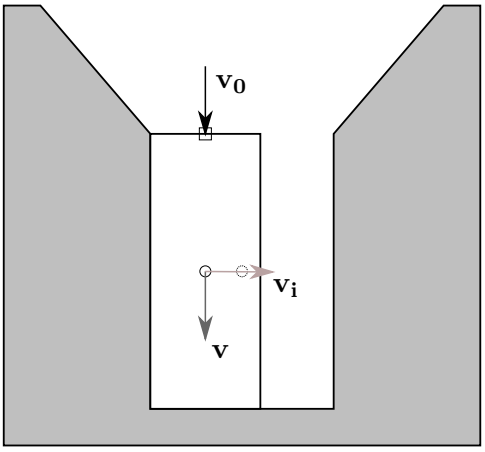


Figure 2.4: Extremal Associated with Successful Assembly. Extremal which requires a motion impossible to be achieved with a passive mechanism and can be considered proper assembly.

In order to solve this issue a new set of extremals is identified using a one parameter genetic algorithm. This optimization identifies the maximum and minimum values of θ for both δ_{min} and δ_{max}. These extremals represent the ones that are possible with the given geometry. Figure 2.5 shows the extremals obtained for a single E-V contact state in a rectangular peg assembly. Once this new extremal set is identified, extremals corresponding to successful assembly such as extremals 3 and 4 in Figure 2.5 are removed from the set. This results in a set of configurations representative of the assembly task.

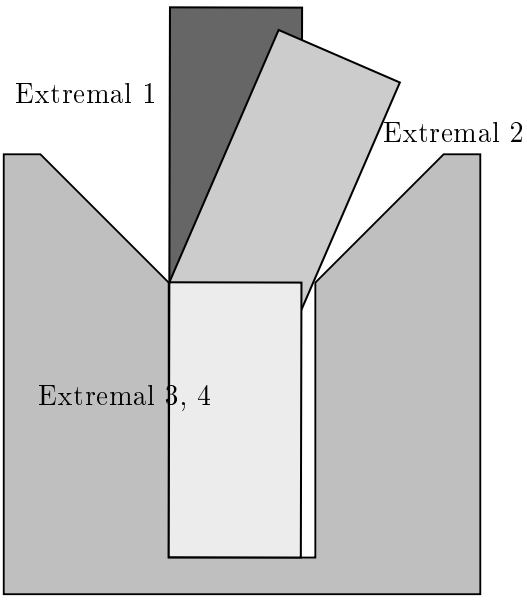


Figure 2.5: Extremals Used for Velocity Metric Optimization. These extremals do not result in penetration of the parts. Extremals 3 and 4, are considered successful assembly and therefore are not included in the set of configurations used in evaluating the motion quality.

This set always includes those having the maximum deviation from the successfully assembled position.

2.4 Admittance Selection

This section provides insight and an explanation of the operations performed in the last part of the procedure. In this component the admittance is selected based on two criteria: satisfaction of the sufficient conditions for error reduction and the best quality of the motions of the set of extremals identified in the previous component.

In order to determine the quality of the motions of the set of configurations evaluated and hence the overall performance of a selected admittance the rigid body velocity metric developed in [5] was used. This metric allows two different motions to be compared yielding a scalar quantity of the proximity of the two motions relative to each other. This is used to identify the admittance that yields close to ideal motions for a set of representative configurations. In order to apply this metric, a motion which is considered to be best for a given configuration has to be identified.

2.4.1 Ideal Motion Identification

The best motion for a given configuration corresponds to that which causes the body to move from the current position to the final assembled position at the same rate as the commanded motion if no constraints where present. This motion is represented by the velocity vector from the current position of the body's geometric center to the successfully assembled position of this point with simultaneous rotation of the body. This ideal motion is expressed as a twist, $\mathbf{t_i}$. This twist is only considered optimal for the configuration selected. This motion is not necessarily possible as in most situations it corresponds to penetration between the two parts being assembled. Figure 2.6 presents the parameters used in determining this motion.

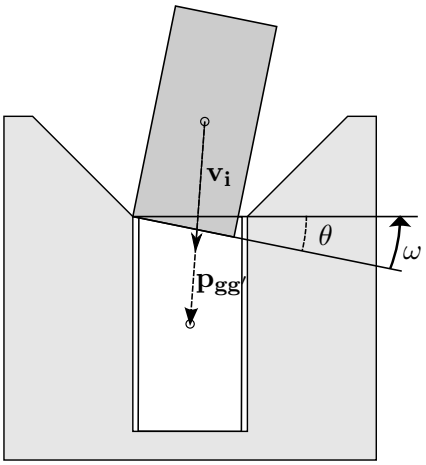


Figure 2.6: Ideal Unconstrained Motion. The ideal motion corresponds to the direct line of motion of the geometric center of the peg to its properly mated position. Vector $\mathbf{p_{gg'}}$ represents the position discrepancy of the geometric center and θ the discrepancy in orientation between the current and assembled position. Components $\mathbf{v_i}$ and ω represent the components of the twist $\mathbf{t_i}$.

It is important to note that the value of $\mathbf{v_i}$ and ω depends on the selected time τ. The value of τ scales the magnitude of the motion. The value is selected so the magnitude of the translational component of the ideal motion is close to that of the translational component of the nominal motion. This requirement is expressed by the following equality:

$$\tau = \frac{|\mathbf{p_{gg'}}|}{|\mathbf{V_0}|} \tag{2.1}$$

This results in an ideal motion that will possess the same magnitude in the translational component as that of the nominal motion $\mathbf{t_0}$, which in itself is the desired translational motion if no misalignment were present.

2.4.2 Calculation of Motion Quality

Once the ideal motion has been identified, calculation of the discrepancy of another motion $\mathbf{t}$ compared to the ideal motion can be performed. This value can be used to determine the overall performance of a selected admittance. The result, $\mathbf{t}$, of both nominal motion applied by the robot and the corrective motion generated by the admittance behavior as defined by the control law Equation 1.2. The discrepancy between the resulting and ideal motions is obtained by computing the average difference of the ideal and resulting motion of each particle contained in the body. This average discrepancy is represented by a scalar value V_m obtained using Equation 2.2.

$$V_m = |\Delta\omega|\bar{r}(\beta, \mathbf{r_{oC}}) \tag{2.2}$$

The value of $\bar{r}$ is a measure of the average distance from the instantaneous center associated with the motion discrepancy to the body. It is a function of the body β and the distance from the origin frame to the instantaneous center of the motion discrepancy, r_{oC}. Criales [5] obtained analytical expressions for the value of $\bar{r}$ for different simple geometries, as well as a process by which to decompose a complex shape into simpler geometries and the calculation of its corresponding $\bar{r}$ values. This process of evaluating the quality of a motion is used to evaluate the overall effectiveness of an admittance.

As stated in Chapter 1, the admittance does not change during the assembly process. A single admittance is selected to provide error reduction for all contact states. This means that in most situations the ideal motion for each configuration evaluated cannot be accomplished by a single admittance. In order to address this issue the best worst-case scenario is minimized. This means that for the extremals that are being used for the velocity optimization, the configuration which results in the worst case (highest V_m value) is minimized. All other configurations within the evaluated set will have better (lower) V_m values.

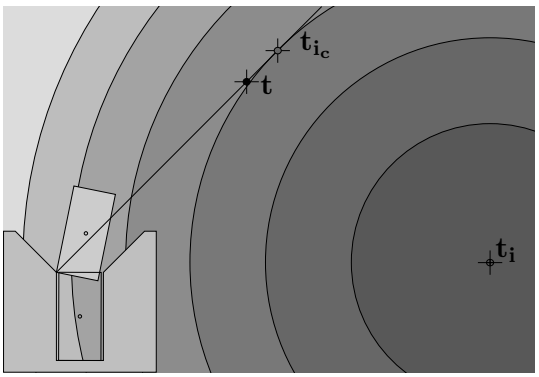


Figure 2.7: Important Instant Center Locations: $\mathbf{t_i}$ represents the ideal unconstrained motion, $\mathbf{t_{i_c}}$ corresponds to the best constrained motion which possesses the shortest (perpendicular) distance to $\mathbf{t_i}$, and $\mathbf{t}$ the best worst case motion obtained by the optimization. The instant centers located on the circles shown all possess the same V_m values.

Figure 2.7 presents a graphical representation of the process of selection of a high quality motion for a single point contact state. The best constrained motion $\mathbf{t_{i_c}}$ is the motion that would result in the lowest V_m, while still being possible within the assembly task. This means that the average difference for the motion of all particles towards the successfully assembled position is minimized. This motion does not cause penetration of the assembled parts and maintains contact due to its instant center being located on the normal of the surface in contact.

Due to issues explained previously it is impossible to obtain the best constrained motion for all configurations evaluated. In order to identify an admittance that results in high quality motion for all contacts states the quality of the motion at multiple configurations must be simultaneously taken into account. To accomplish this, the best admittance is the one that minimizes the worst-case scenario. Due to conflicting objectives in other contact states, the motion for a single configuration would not be $\mathbf{t_{i_c}}$, but $\mathbf{t}$ another motion located along the surface normal. This motion is close to optimal motion for this configuration. The admittance resulting in motion $\mathbf{t}$ satisfies all the conditions outlined by [25], maintaining contact and error reduction for all intermediate configurations. Furthermore it will accomplish high quality motion for all other extremal configurations being considered. To achieve this computationally the V_m value of the worst case is used as a measurement of overall effectiveness of the admittance matrix.

2.4.3 Best Worst-Case Minimization

The process by which the best worst-case scenario is minimized is a constrained minimax optimization. Due to issues with non-continuous space and presence of local minima traditional minimax gradient search techniques were not used. Instead a genetic minimax algorithm was used to find a global minimum. The algorithm generates a set of possible admittances that satisfy the error-reducing constraints, and then obtains the quality of the motion,V_m resulting from said admittances for all configurations being evaluated. Then it selects the worst performing one and attempts to minimize it. However, as with any random approach optimization, the algorithm is good at determining convergence within the population but cannot determine if the result obtained is a global or local minimum. Because of this, the genetic optimization for admittance selection is sequentially performed 4 times with the obtained solution of one optimization used in the initial population of the following one. While this does not guarantee a global minimum, it does give significant confidence that the admittance selected obtains high quality motion. Further explanation of genetic algorithms can be found in Appendix E.

In order to make the program faster the MATLAB program was compiled as a C++ program which was then submitted as a Condor script to a distributed computer network, increasing the speed of calculations significantly. Appendix F outlines the process of submitting a script to CONDOR.

2.5 Discussion

This chapter presented an introduction of the velocity metric used to select an admittance resulting in close to optimal motion. It also presented the necessary steps for implementing said metric into the process used for admittance selection, including configuration and time value selection. Finally, a high level overview of the developed strategy was discussed with emphasis on its implementation.

The following chapters present the admittances obtained by the developed program resulting in high quality force guided assembly for three simple geometries (triangular, rectangular and stake shaped peg). The chapters will show the effectiveness of this admittance selection strategy for force-assembly.

3 RECTANGULAR PEG ASSEMBLY

To evaluate the velocity metric based admittance selection procedure, results for different geometries are needed. This chapter presents the optimal admittance results for a rectangular peg assembly obtained when using the motion quality as the objective function. The peg assembly consists of a rectangular body held by the manipulator moving into a fixed chamfered rectangular hole. The results were obtained for a range of different geometries. Relationships between the part geometry and the optimal admittance are identified. Results are also compared with those for the previously used maximum friction based optimization.

First, the variation in part geometry considered is described. Next, results for the optimizations are presented and trends are investigated. These trends include those observed for the admittance components, the resulting quality of the best worst case motion and the location of the compliance center for the obtained admittances. Finally, an optimal admittance behavior obtained using this approach is compared to the results for the maximized friction selection strategy to determine the effectiveness of the velocity metric based optimization.

3.1 Assembly Description

This section identifies part geometry and part contacts considered in generating the optimal admittance for a rectangular peg assembly. Related items include the part clearance, successful assembly conditions, and contributing contact states.

Figure 3.1 shows the dimensions of the geometry being investigated. The variation in geometry used is the aspect ratio defined as L/W, where L is normalized with respect to W to obtain a clear picture of the behavior of the admittance and its components without having to consider units.

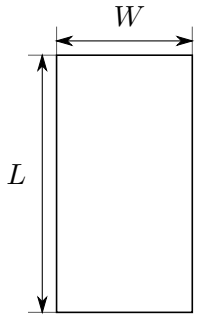


Figure 3.1: Rectangular Peg Dimensions. The length is normalized by the width.

3.1.1 Contact State Enumeration

Figure 3.2, presents the corresponding edge and vertex numbers for both the fixed and movable parts.

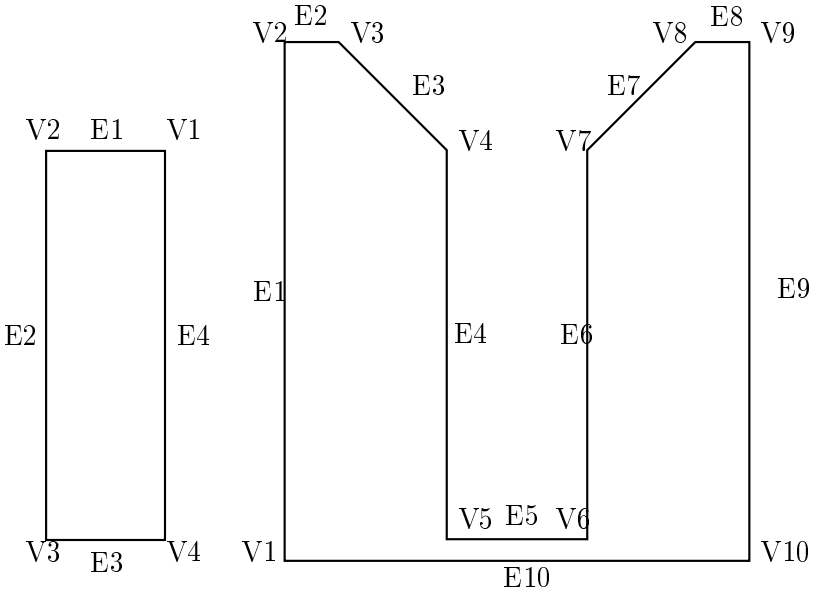


Figure 3.2: Feature Enumeration for Rectangular Peg Assembly.

In order to account for the misalignment present on the robot positioning the following bounds where selected:

- $\pm X_{RB} = [-1.87, 1.87]$
- $\pm Y_{RB} = [0, 24.5]$
- $\pm \Theta_{RB} = [-\frac{\pi}{36}, \frac{\pi}{36}]$

The X_{RB} and Y_{RB} values are unit less as the investigation is based on the L/W ratio. The Θ_{RB} is expressed in radians. These errors are selected as conservative estimates of the misalignment experienced by the robot which is completing the assembly task.

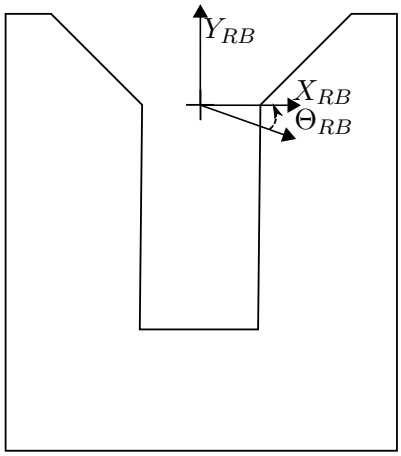


Figure 3.3: Frame Used as Basis to Describe Misalignment Bounds.

The selected clearance for the parts is applied on the x direction and is selected as 0.10 units.

A configuration is defined as successful assembly if the location of the top of the peg is within the selected bound:

- $y_{succesful} = \{0, 0.01\}$

It is necessary to select this bound in order to adjust the process for difficult configurations, for example situations where the ideal motion corresponds to complete horizontal movement, or where the peg is situated at the hole bottom and is misaligned by the clearance.

For a rectangular peg assembly the following contact states are not considered successful assembly and contribute to the optimization process and constraint generation.

- V4-E2

- V7-E4
- E3-V3
- E4-V3
- E6-V4
- E7-V4

For parts which have an aspect ratio larger that 1.2 the following contact states also affect the optimization.

- V4-E2, E6-V4
- V7-E4, E4-V3

This is dependent on the clearance between the assembly part and the assembly bounds. For a part with an aspect ratio less than 1.2 it is impossible for it to have two point contact within the robot bounds specified earlier, which results in fewer constraints needing to be satisfied.

3.2 Results

The following results are obtained for the values of the admittance components for different coefficients of friction. Each plot presents the values of the a_{33}, a_{12}, a_{13}, and a_{23} components as the aspect ratio of the part is increased. These components correspond to what was identified as the optimal mapping from contact forces to corrective motion as explained in Chapter 1.

The values of a_{22} (the mapping between vertical force to vertical motion) are not presented as they do not follow an identifiable pattern, which is in line with results obtained for the previous maximum friction based optimization [25]. It is important to note that due to the nonlinear nature of the optimization and the use of a random approach, the results tend to be have some noise.

The results indicate that both the increase in the friction value and the aspect ratio have an impact on the results. A significant pattern can be observed in all three of the Figures 3.4, 3.5, and 3.6. As the aspect ratio of the rectangular peg increases the values begin to plateau towards a value for all four components shown, with a_{12} and

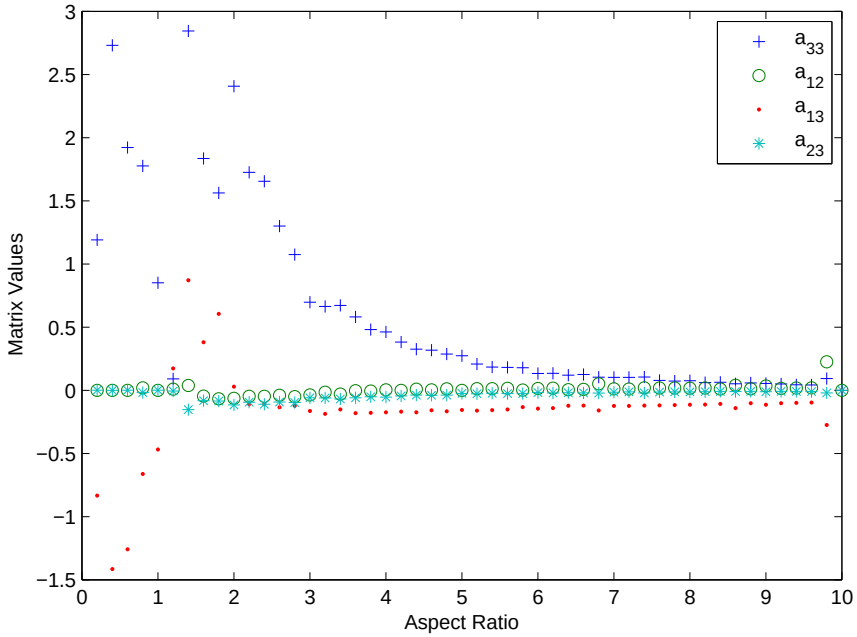


Figure 3.4: Admittance Characteristics for Rectangular Assembly Using a Coefficient of Friction of 0.3. Large aspect ratios converge to small values.

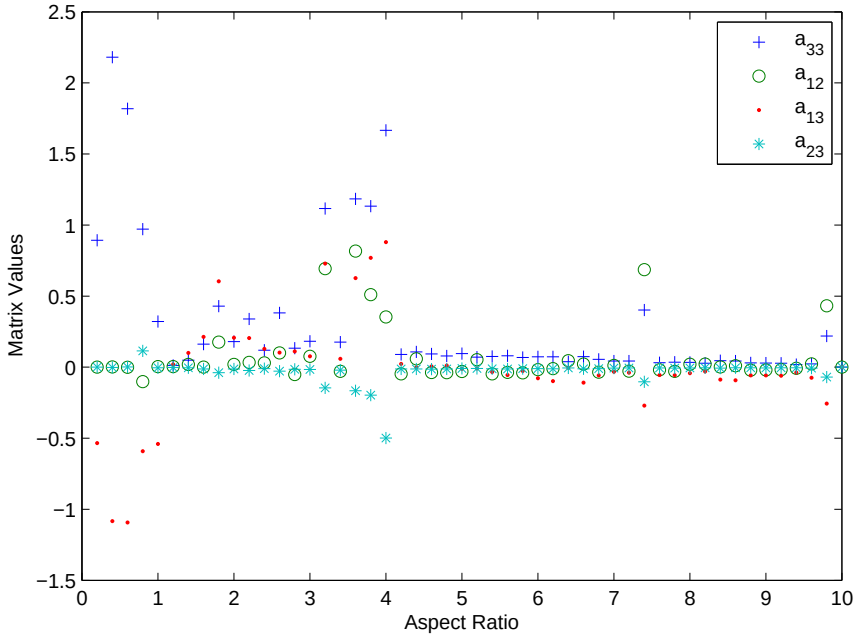


Figure 3.5: Admittance Characteristics for Rectangular Assembly Using a Coefficient of Friction of 0.5. Large aspect ratios converge to small values.

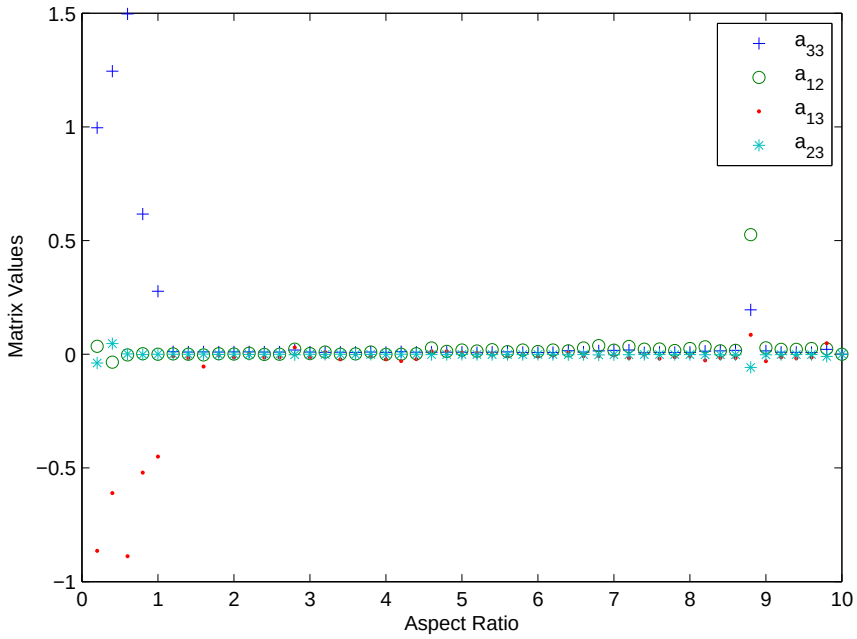


Figure 3.6: Admittance Characteristics for Rectangular Assembly using a Coefficient of Friction of 0.7. Large aspect ratios converge to small values.

a_{23} tending towards zero, a_{33} leveling out to a small positive value, and a_{13} remaining a small negative value. This pattern is observed for all coefficients of friction investigated.

The increase in the aspect ratio affects the number of possible admittances that satisfy the requirements of error reduction. As the aspect ratio increases the number of possible admittances resulting in force guidance is reduced. This is most apparent when looking at the discontinuity of the results in Figures 3.4, 3.5, and 3.6. This discontinuity occurs at the point where two point contacts are possible within the bounds of the assembly task, at an aspect ratio of 1.2. As found by Wiemer [25] two point contact states in general are the ones that constrain the optimization the most and hence have a significant impact in the amount of available solutions. For smaller aspect ratios, the space is significantly larger resulting in more varied results for the admittances.

The increase of the coefficient of friction also reduces the space of possible values for the admittance components. The larger the range for the coefficient of friction the more admittances that are eliminated from the space of possibilities. This is due to the fact that error reduction has to be achieved for all configurations in the range from 0 to the selected coefficient of friction. The following sections present an in depth discussion of

behavior of the each of the admittance components as a function of the rectangular aspect ratio, L/W, as well as evaluating the overall performance of the selected admittance.

3.2.1 a_{22} Component

As stated previously, the value of a_{22} does not seem to follow a pattern. The value of V_m seems to be largely unaffected by the value of a_{22} as shown by Figure 3.7. As found in [25] a_{22}'s main role is in generating admittance matrices that are positive definite, which is required for the admittance to be generated by a passive mechanism.

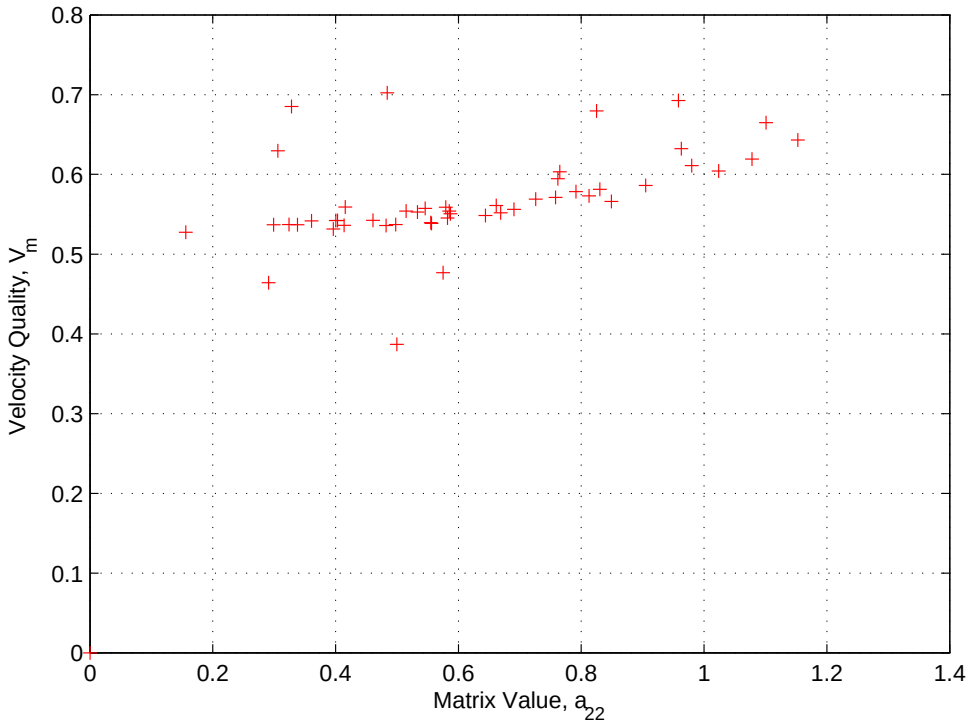


Figure 3.7: Relation between a_{22} and V_m for a Friction Coefficient of 0.3. The change in the value of a_{22} does not seem to have an effect on the quality of the resulting motion.

3.2.2 a_{12}, a_{23} Components

From Figure 3.8 it can be observed that the values for a_{12} and a_{23} do not vary greatly depending on the aspect ratio, with both components approaching values close to 0. Once again the discontinuity caused by the inclusion of two point contacts is apparent. These components tend to level out to values that are relatively small. As pointed out by Wiemer [25], the value of these components needs to be small in order to maintain symmetry for the motions. Due to Equation 1.2 it becomes necessary that these values are close to zero to guarantee that changing from one contact state to its mirror does not result in motion occurring in a non-error reducing direction.

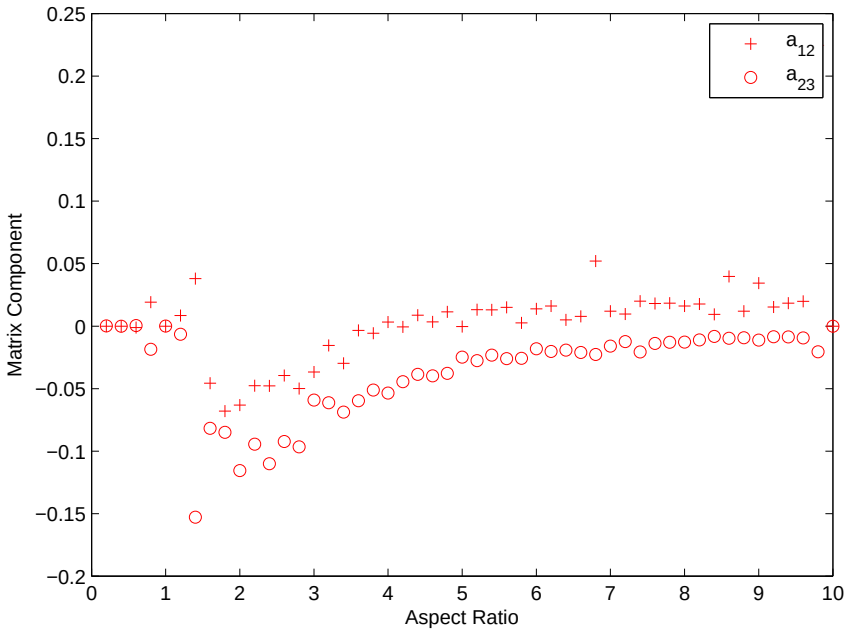


Figure 3.8: Resulting a_{12} and a_{23} Components for Varying Aspect Ratio for Rectangular Assembly for a Friction Coefficient of 0.3. The components approach a small due to part symmetry.

3.2.3 a_{13}, a_{33} Components

As shown in equation 1.3 the ratio of these two components represent the y location of the compliant center, y_{cc}; therefore, different values of a_{13} and a_{33} can result in the same y_{cc} location. This is a reason for the generation of a space of solution with multiple local minima. Figure 3.9 shows the location of the compliant center in the y direction. As the peg gets longer, y_{cc} moves lower. However, it is not located at the bottom of the peg, as with the RCC, but decreases with aspect ratio. It is important to note that as the coefficient of friction is increased the pattern for y_{cc} to move downward disappears, and significant noise is present in the obtained patterns. The reason for this is that the space of available admittance components becomes reduced by having to satisfy a larger range of coefficients of friction and the admittances resulting in the ideal compliant center are no longer being available.

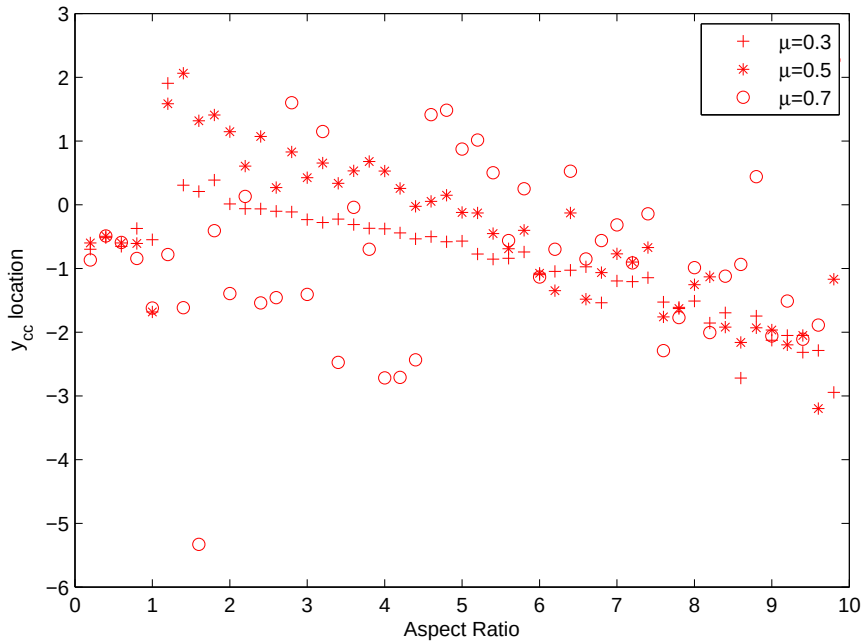
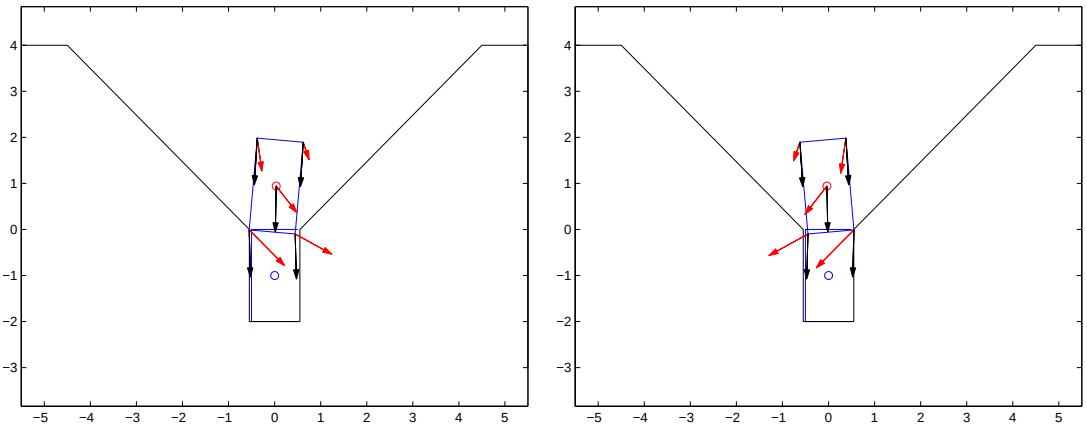


Figure 3.9: Resulting y_{cc} Location for Rectangular Peg Assemblies. The location of the compliant center moves downward from the top of the peg being assembled as the peg becomes longer.

3.2.4 Dominating Contact States

The following contact states represent the contacts which dominate the velocity optimization. These contact states possess extremal configurations with the lowest quality motion in the minimax optimization. The extremals for this case are the *E3-V3* and *E7-V4* contact states.

These extremal configurations are illustrated in Figure 3.10. In both figures the peg is located at bottom of the chamfer and is about to transition into another contact state. The high deviation from the ideal motion can be explained by the motion being constrained by the edge of the chamfer. The ideal motion is almost directly down into the hole, hence there is a limiting factor for how close the motions can be to each other.



(a) E3-V3 (b) E7-V4

Figure 3.10: Dominating Contact States for Rectangular Peg Assembly. These contact state configurations correspond to transition into a different contact state.

Table 3.1: Plateau V_m Values for Different Friction Coefficients.

μ	V_m
0.3	0.55
0.5	0.60
0.7	0.75

3.2.5 V_m Values for Optimization

Figure 3.11 shows the resulting motion quality obtained by the process for a range of aspect ratios and friction coefficients. It can be seen that there are clear correlations between the V_m value and the aspect ratio of the part and the friction coefficient. From the observed discontinuity it is determined that the inclusion of two point contact states has a significant effect on the performance of the admittance, meaning the quality of the motion of the worst-case. The aspect ratios for which two point contacts do not occur are seemingly unaffected by the coefficient of friction. The motion quality results for each friction coefficient plateau to a different result as aspect ratio increases, as shown in Table 3.1.

As can be observed there exists a change in behavior with an increase in the coefficient of friction. The lower coefficients lead to smaller values for longer aspect ratios while the higher coefficients of friction lead to the values of optimization reaching a plateau at a higher value with increasing aspect ratio. In order to explain these results graphical representations of the motion of the extremals are presented.

Figures 3.12a and 3.12c and Figures 3.13a and 3.13b demonstrate that for lower coefficients of friction, the optimization is dominated by geometry constraints. At a higher aspect ratio the discrepancy between the motions becomes smaller. This can be attributed to the relative magnitude of the desired motion and the effect of angular velocity on the resulting motion. The motion at the bottom corners seems to be similar for both aspect ratios; however the top corners (A and B in Figure 3.13) present significant increase in discrepancy at the lower aspect ratios. Since the velocity metric takes an average, the motion for the longer part possess higher quality. By using the average of the discrepancy of the motion of the four corners that result can be demonstrated. The shorter peg has an average discrepancy of 0.643 and the longer 0.550. This leads to the

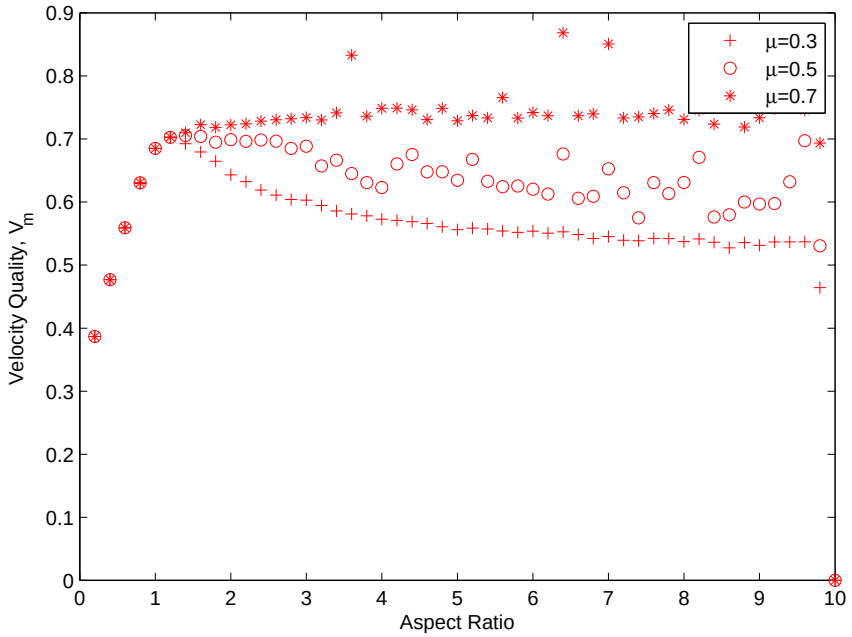
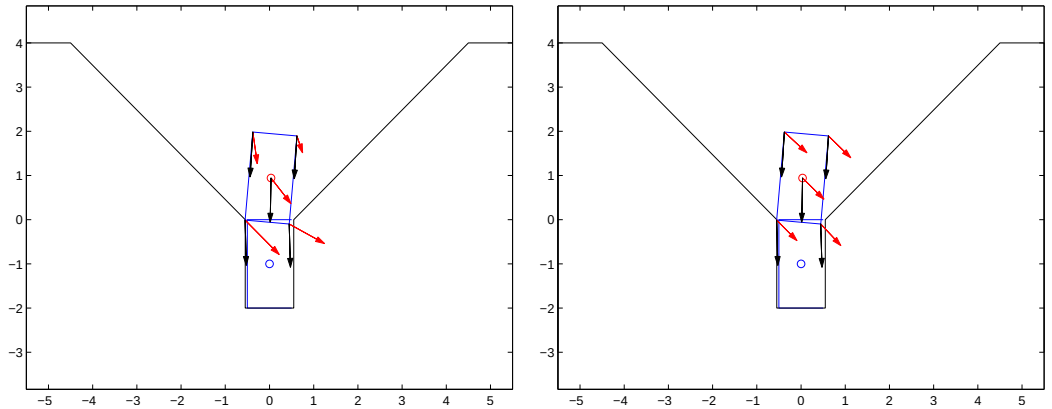


Figure 3.11: Velocity Quality Results for Rectangular Peg Assembly. The change in the coefficient of friction causes a change in performance.

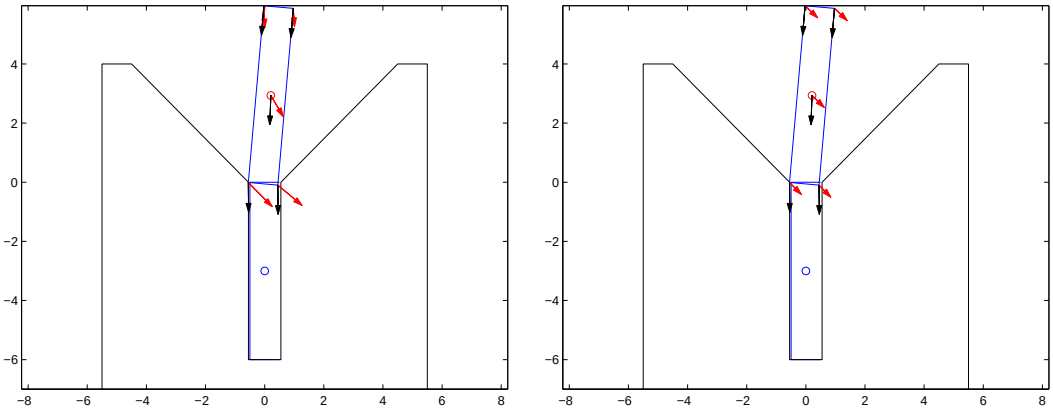
value of the optimization reaching a V_m value that is smaller with a larger aspect ratio for low coefficients of friction.

The pattern of the quality of the motion being larger for small aspect ratios is not maintained for large coefficients of static friction. In the case of large friction coefficients the optimization results are constrained by the magnitude of the friction force obtained at the contact point. The conditions for error reduction also become more difficult to satisfy and the space of possible admittance becomes smaller. As can be seen in Figures 3.12b and 3.12d, the resulting magnitude of motion for the longer aspect ratios becomes significantly smaller than the desired motion resulting in higher discrepancy between the actual and ideal motions. Because of this the high friction cases possess less quality than the lower ones.

The plateauing of the values can be explained by observing that at a certain point the increase in aspect ratio becomes relatively insignificant from one part to the next increase in length, hence with an increase of 1 unit the change in the motions becomes less significant.

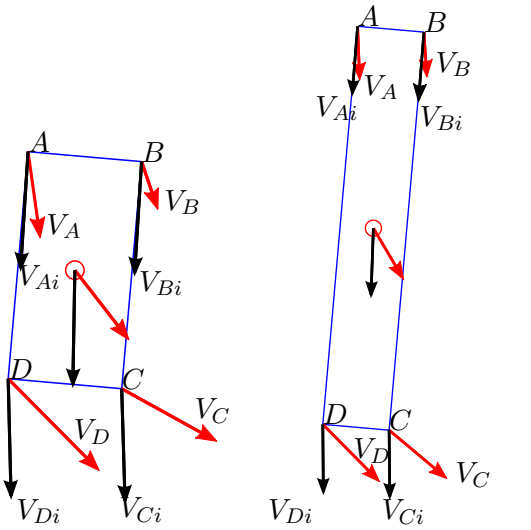


(a) $L/W = 2.0$ $\mu = 0.3$ (b) $L/W = 2.0$ $\mu = 0.7$



(c) $L/W = 6.0$ $\mu = 0.3$ (d) $L/W = 6.0$ $\mu = 0.7$

Figure 3.12: Resulting Velocities for Dominating Contact State. Friction coefficients behavior of the results.



(a) $L/W = 2.0$ $\mu = 0.3$ (b) $L/W = 6.0$ $\mu = 0.3$

Figure 3.13: Detail of Velocities for Dominating Contact State. Longer aspect ratios possess better V_m quality for low coefficients of friction.

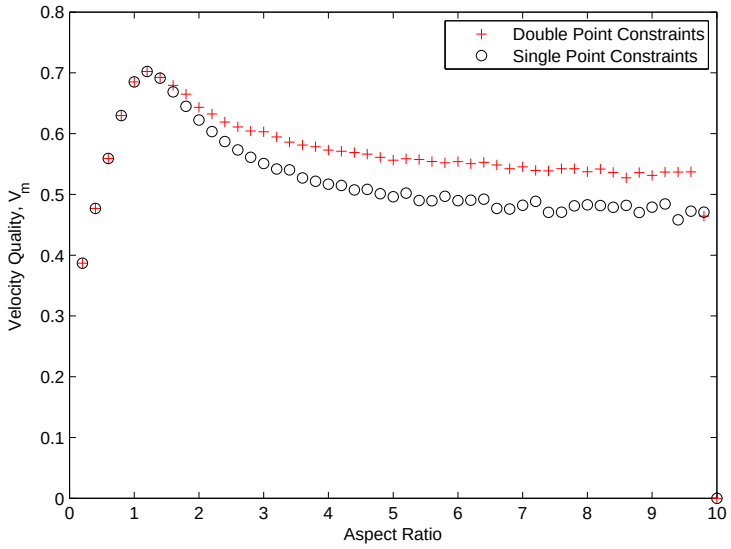


Figure 3.14: Velocity Quality Results for $\mu = 0.3$ with Varying Constraints for Rectangular Peg. The effect of the coefficient of friction is not significant.

3.2.6 Friction Coefficient

The effect of a change in friction is further corroborated when observing the effect of the inclusion of two point constraints into the optimization. The two point contact states are the ones that constrain the optimization the most. When these two point constraints are not included in the optimization, the results for a large coefficient greatly resemble those of the smaller coefficient of friction. This is shown in Figures 3.14 and 3.15. The increase in the coefficient of friction reduces the space of available admittances, by making the error reducing conditions harder to satisfy. This is due to an increase in the range for which friction has to be satisfied, as all admittances must work for lower coefficient of friction. With a reduced space of available admittances, optimal motion becomes harder to accomplish.

Figure 3.16 presents the obtained components for the maximum friction based optimization. Comparing those results to the ones obtained by the velocity metric based approach, whose results are shown in Figure 3.6, it is observed that they are quite similar. Once again this corroborates that an increase in friction results in a reduced space of possible admittances. As the friction increases the acceptable number of admittances is reduced.

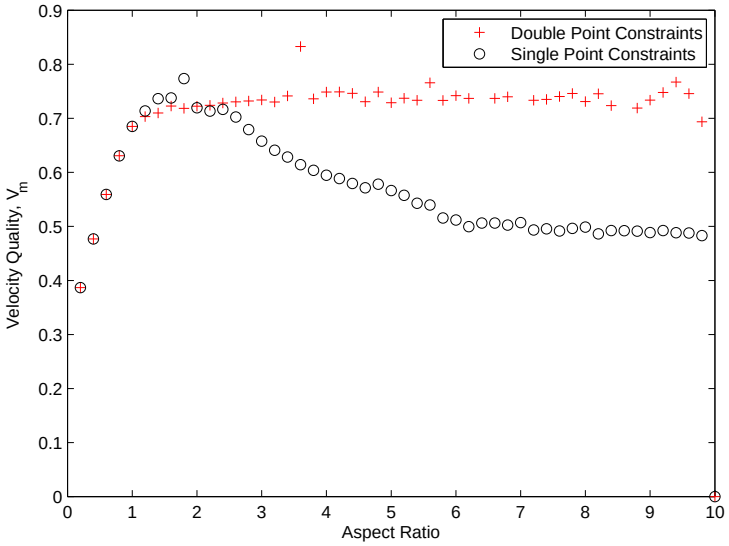


Figure 3.15: Objective Function Results for $\mu = 0.7$ with Varying Constraints for Rectangular Peg. The effect of the coefficient of friction is significant.

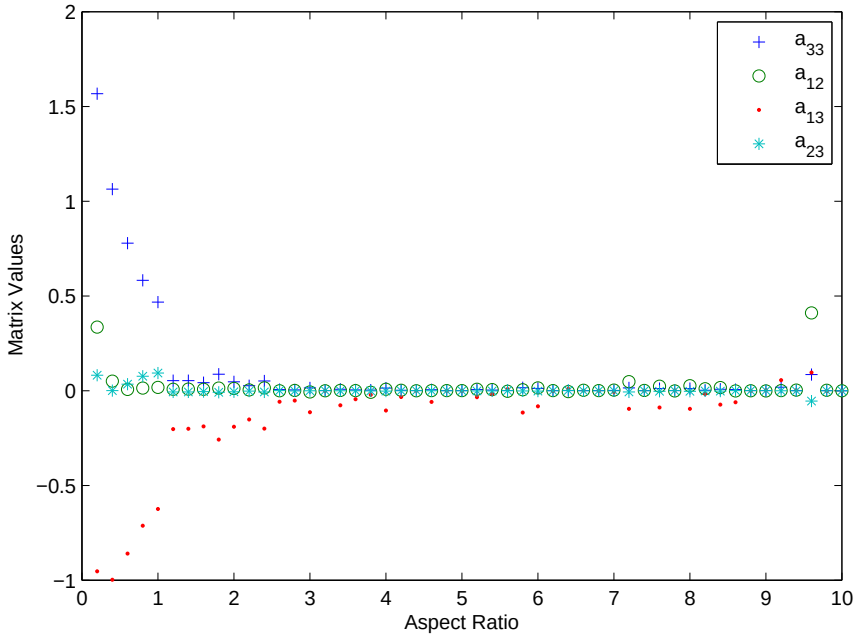


Figure 3.16: Admittance Characteristics for Rectangular Assembly Using a Maximized Friction Optimization for Rectangular Peg.

3.3 Comparison between Optimization Routines

It is important to compare the quality and behavior of the different matrices for situations that have a different coefficient of friction, meaning how does the matrix that was optimized for a certain friction coefficient behave for a smaller friction coefficient. Applying a smaller coefficient optimized friction to a larger one is inadvisable as it does not necessarily satisfy the error reduction conditions. Figure 3.17 presents the resulting worst case value of the velocity metric for the different aspect ratios investigated, using different coefficients of friction.

As expected the optimization routine based on the velocity metric generates motion for the extremals that is considered higher quality. This is especially true when the selected admittance was obtained for a friction coefficient which is close to that for which the admittance is being evaluated. However, it be seen that the performance of the velocity metric optimization is better than the friction based optimization for all cases considered.

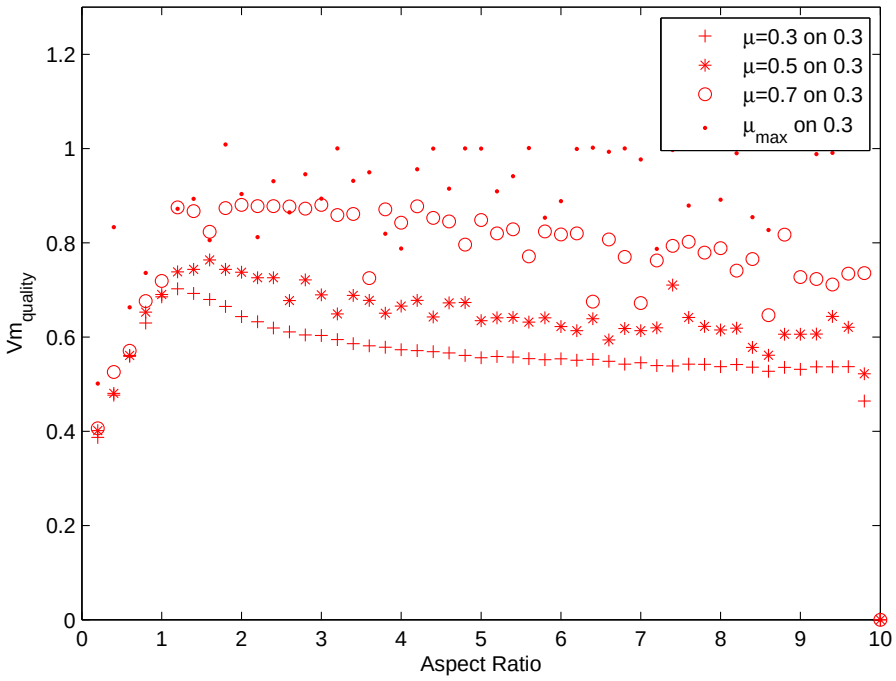


Figure 3.17: Resulting Motion Quality for Rectangular Assembly for $\mu = 0.3$. The velocity metric outperforms the friction based optimization.

3.4 Discussion of Results

Observing the different admittance matrix values obtained for different geometries it can be seen that the admittances demonstrate two distinct behaviors, one before an aspect ratio of 1.2 is reached and another after. For assembly tasks that have an aspect ratio smaller than 1.2, the behavior will be quite similar regardless of the friction coefficient being used. For all these geometries two point contact cannot exist and hence they possess a larger space of admittances that satisfy the error reducing conditions.

As found by Wiemer [25], for most geometries the two point contacts are the ones that most constrain the optimization. Because of this, the space of acceptable admittances is larger for shorter parts than for longer ones. Once the two point contact constraints are applied, the optimization is driven mostly by the identification of acceptable solutions than admittances which result in close to optimal motion. This is most apparent when optimizing for larger coefficients of friction.

The comparisons indicate that the velocity metric performs better than the maximum friction optimization, which is especially apparent when evaluating its performance at the coefficient of friction for which it was selected. However, results also showed that if the target and evaluated coefficients of friction are close together its performance is better than that of the maximum friction optimization.

The results obtained by the velocity metric optimization all result in high quality error reducing motion. The friction based approach, does not accomplish this, sometimes selecting admittances that almost violate the constraints (including error reduction). This is why the motion quality of its results is in most situations close to 1, resulting in motion that is barely moving towards the proper position. It is expected that the matrices selected by the velocity metric procedure will be more robust and desirable for assembly tasks with coefficients that are less than 0.7.

There is still a possibility that for a given configuration the result of the velocity metric could perform worse than that of the friction based approach, as the optimization only improves a selected set of configurations. However the velocity will still be error-reducing, and always result in successful assembly even if it is accomplished in a longer time frame due to constraint satisfaction. Furthermore, as will be shown in Chapter 6,

there is significant numerical evidence to demonstrate that the quality of intermediate motions is better than of the considered extremals.

3.5 Summary of Chapter 3

In this chapter the results for a rectangular peg assembly were discussed. It is demonstrated that the use of a velocity metric as the objective of a minimax optimization results in a satisfactory compliant system that both has a relatively high quality motion at the extremals and error reduction motion for all possible configurations.

The resulting admittance matrix performs better than the one obtained by the previous friction based program. The statement holds true for all the coefficients investigated demonstrating the viability of the procedure for admittance selection.

When examining the admittance, the resulting motion for the extremals is closer to the intended nominal velocity using the velocity metric than the friction approach providing a manner by which the direction and magnitude of the resulting motion can be controlled.

The following chapters will present results for assembly tasks involving a triangular and stake shaped peg. It will be shown that the velocity metric based optimization is successful in generating force-guided admittance with optimal motion for those geometries.

4 TRIANGULAR PEG ASSEMBLY

To evaluate a velocity metric based optimization resulting in optimal motion, results for different geometries are needed. This chapter presents the optimal admittance results for a triangular peg assembly obtained when using the resulting motion quality as the objective function of a minimax optimization. The peg assembly consists of a triangular movable part held by a manipulator being inserted into a fixed chamfered part with a triangular hole. The results were obtained for a range of triangles of different aspect ratios. Relationships between the part geometry and optimal admittance are identified. Results also present improvement over the previously used friction based optimization by obtaining higher quality motion as defined by the velocity metric.

First the variation in part geometry considered is described. Next, results for the optimizations are presented and trends are investigated. These trends include those of the admittance components, the resulting quality of the best worst case motion and location of the compliance center for the admittances selected. Finally, optimal admittance behavior obtained using this approach is compared to the results obtained by the maximized friction approach to determine its effectiveness.

4.1 Assembly Description

This section identifies part geometry and part contacts considered in generating the optimal admittance for a triangular peg assembly. Related items include successful assembly conditions and contributing contact states.

Figure 4.1 shows the dimensions of the part being investigated. The variation in geometry used is the aspect ratio defined as L/W, L is normalized with respect to W, to obtain a clear picture of the behavior of the admittance and its components without having to consider units.

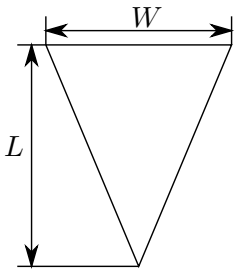


Figure 4.1: Triangular Peg Dimensions. The length of the part is normalized with respect to the width of the peg.

4.1.1 Contact State Enumeration

Figure 4.2, presents the corresponding edge and vertex numbers for both the fixed and movable parts.

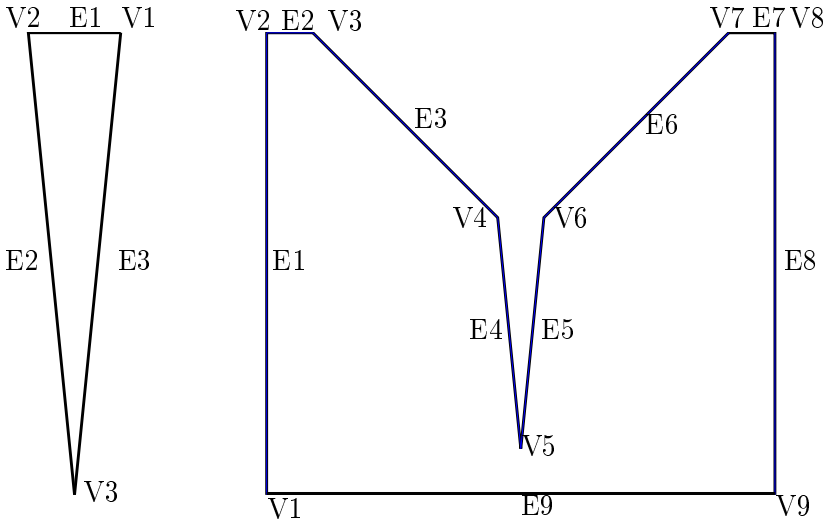


Figure 4.2: Feature Enumeration for Triangular Assembly.

In order to account for the misalignment present on the robot positioning the following bounds where selected:

- $\pm X_{RB} = [-1.87, 1.87]$
- $\pm Y_{RB} = [0, 24.5]$
- $\pm \Theta_{RB} = [-\frac{\pi}{36}, \frac{\pi}{36}]$

The X_{RB} and Y_{RB} values are dimensionless and the Θ_{RB} is expressed in radians. These error boundaries are selected as conservative estimates of the misalignment experienced by the robot performing the assembly, it is not expected that any robot will possess positioning error of similar magnitude.

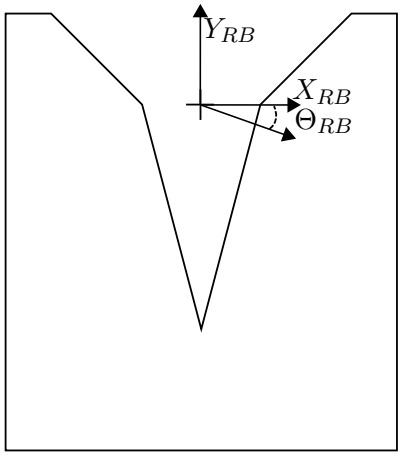


Figure 4.3: Frame Used as Basis to Describe Misalignment Bounds in a Triangular Peg Assembly.

Due to the conformable nature of the assembled parts it is unnecessary to include clearance in the fixed part, as opposed to the rectangular peg assembly.

A configuration is defined as successful assembly if the location of the top of the peg is within the selected bound:

- $y_{succesful} = \{0, 0.01\}$

It is necessary to select this bound in order to adjust the process for difficult configurations, for example situations where the ideal motion corresponds to complete horizontal movement, or where the peg is situated at the bottom of the hole.

For a triangular peg assembly the following contact states are not considered successful assembly and contribute to the optimization process and constrain generation.

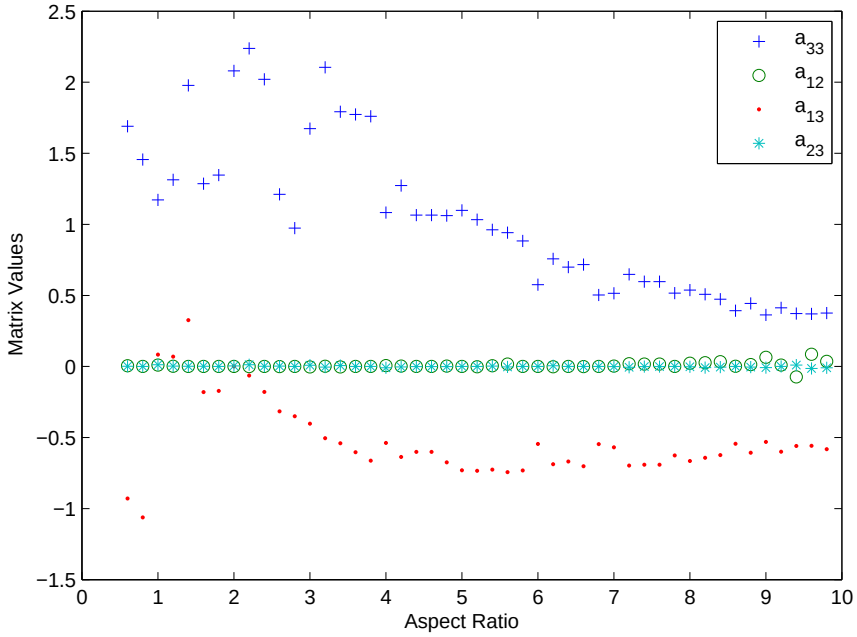


Figure 4.4: Admittance Characteristics for Triangular Assembly Using a Coefficient of Friction of 0.3.

- V4-E2
- V6-E3
- E3-V3
- E4-V3
- E5-V3
- E6-V4
- V4-E2 E5-V3
- V6-E3 E4-V3

4.2 Results

The following results are obtained for the values of the admittance components for different coefficients of friction as the aspect ratio of the part is increased. The components presented are a_{33}, a_{12}, a_{13} and a_{23}, represent the mapping between contact forces/torque into motions in a given direction as presented in Chapter 1.

Once again the values of a_{22} (mapping of force in the y direction to motion in the same direction) are not presented as they do not follow an identifiable pattern, with the quality of the resulting motion seemingly independent of its value. The nonlinear nature

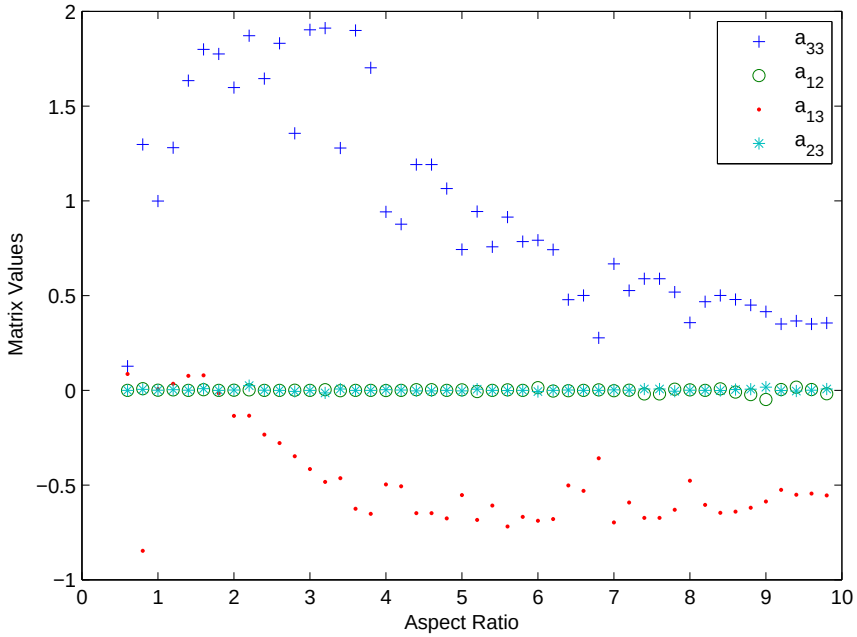


Figure 4.5: Admittance Characteristics for Triangular Assembly Using a Coefficient of Friction of 0.5.

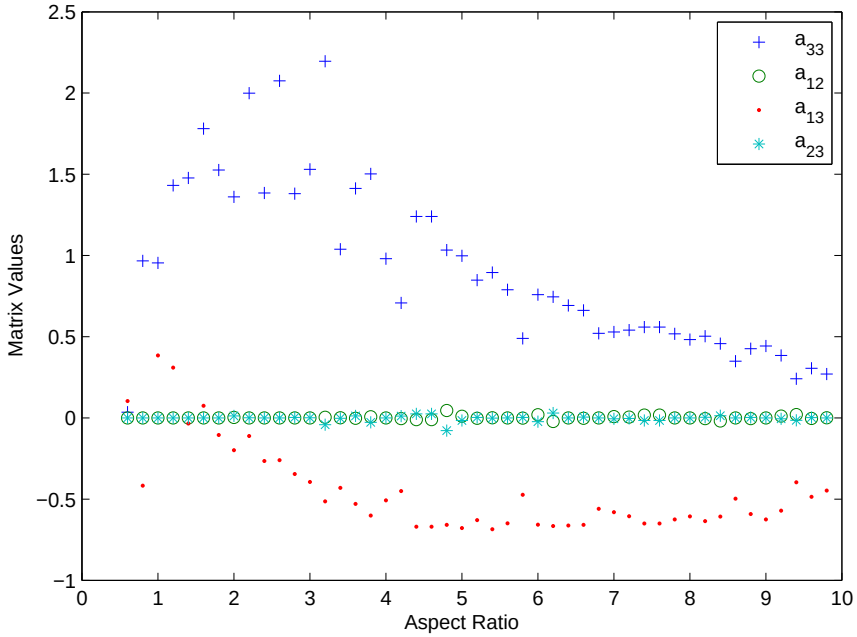


Figure 4.6: Admittance Characteristics for Triangular Assembly Using a Coefficient of Friction of 0.7.

of the problem and the use of a random optimization approach lead to some noise being present in the results.

The results indicate that the change in the aspect ratio affects the value of the matrix components. However, unlike the rectangular peg assembly the value of the friction coefficient does not to play a significant role in matrix component selection. As the aspect ratio increases the components of the admittance plateaus at a certain value regardless of the static coefficient of friction being used. The a_{12} and a_{23} components remain close to zero for all aspect ratios. The a_{33} component remains at a value of 0.4, and finally a_{13} reaches -0.5, while being positive for small aspect ratios. There is significant variation in the components which can be attributed to several admittances resulting in the same behavior for the worst case scenario, and the space of possible admittances remaining rather large, but still being reduced as the aspect ratio increases.

The increase of the coefficient of friction reduces the space of possible values for the admittance components, however as opposed to the rectangular peg the friction coefficient doesn't affect objective function results nor the admittance component selection. This means that while the conditions become harder to satisfy for a longer range of friction coefficients, the reduction of the space of possible admittances is not significant enough to eliminate the admittances that achieve the quality of motion which is being limited by the geometry of the part.

4.2.1 a_{22} Component

As stated before the value of a_{22} does not seem to follow a discernible pattern. The value of V_m seems to be largely unaffected by the value of a_{22} as shown by Figure 4.7. As in the previous section it appears its main role is in generating admittance matrices that are positive definite required so it can be achieved by a passive mechanism.

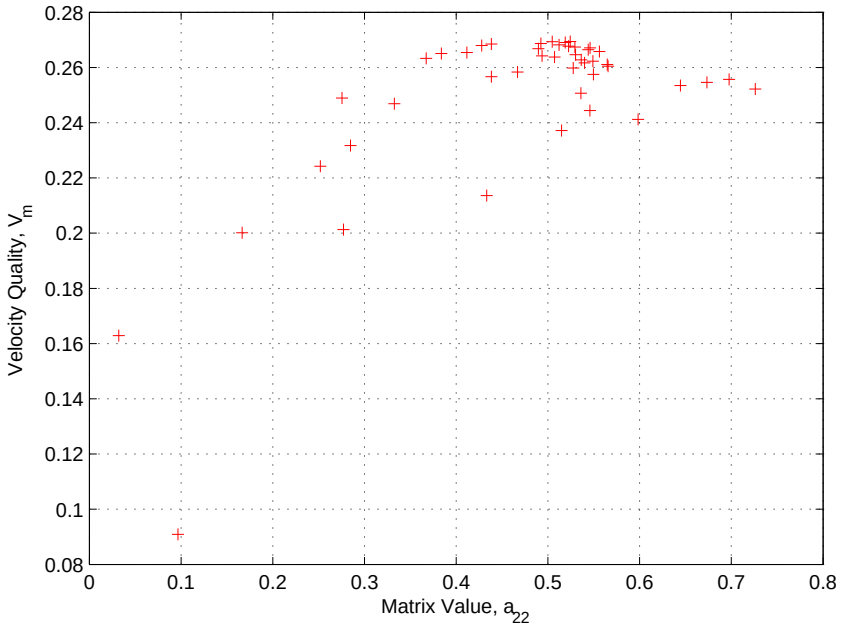


Figure 4.7: Relation between a_{22} and V_m for Triangular Peg for a Friction Coefficient of 0.3. The change in the value of a_{22} does not seem to have an effect on the quality of the resulting motion.

4.2.2 a_{12}, a_{23} Components

From Figure 4.8 it can be observed that the values for a_{12} and a_{23} behave similarly to those obtained for the rectangular peg optimization. Both components approaching small values close to zero with an increase in the aspect ratio. As pointed out by Wiemer [25] the value of these components needs to be small in order to maintain symmetry for the motions. Due to equation 1.2 it becomes necessary that these values are close to zero to guarantee that changing from one contact state to its mirror does not result in motion occurring in a non-error reducing direction.

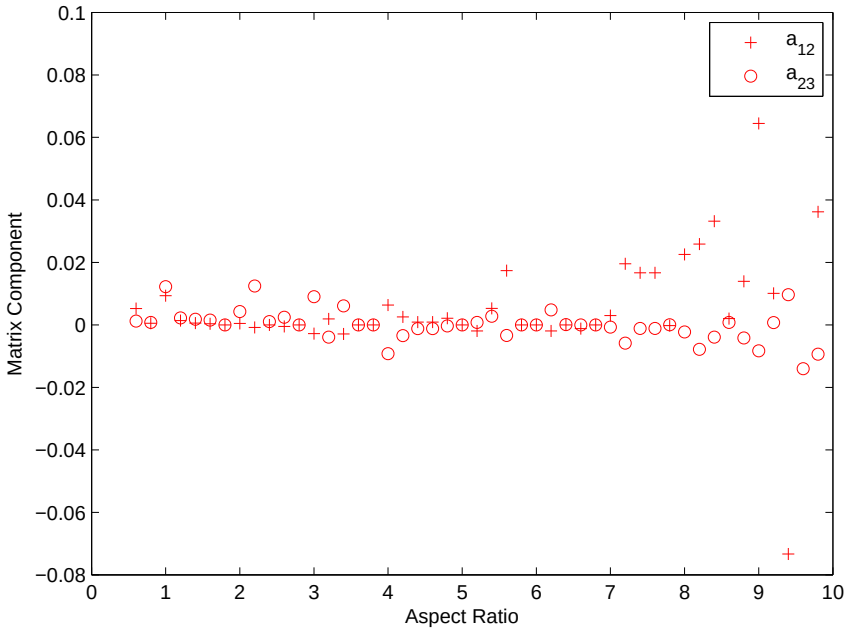


Figure 4.8: Resulting a_{12}, a_{23} Components for Varying Aspect Ratio for Triangular Assembly. The values of the results approach small due to part symmetry.

4.2.3 a_{13}, a_{33} Components

These ratio of these two components represents the y location of the compliant center (point of force/moment decoupling), y_{cc}. The location of this y_{cc} is not unique and several combinations of these values result in a similar location, which leads to a solution space with multiple local minima. Figure 4.9 shows that as the peg gets longer y_{cc} moves downward. Unlike the RCC method the compliant center is not located at the bottom of the peg, but close to a fixed point in the movable part. The increase of the friction coefficient does not have a significant impact on the location of the center of compliance. This can once again be explained by the friction coefficient not significantly reducing the space of possible admittances. It can be seen that there is a small discontinuity in the obtained values at an aspect ratio less than 1, that can be explained by the inclusion of two point contact into the optimization routine.

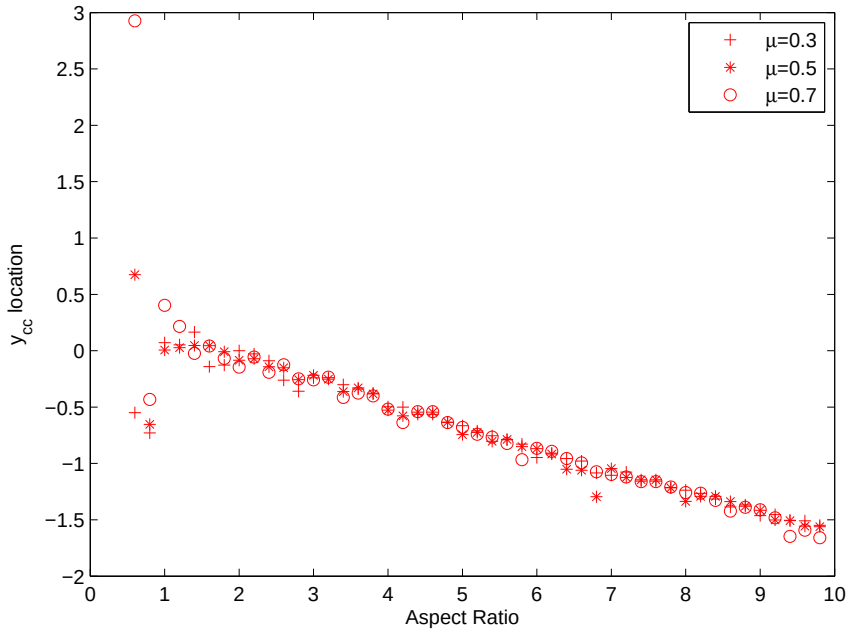


Figure 4.9: Resulting y_{cc} Location for Triangular Peg Assemblies. The location of the compliant center moves downward from the top of the peg as aspect ratio increases.

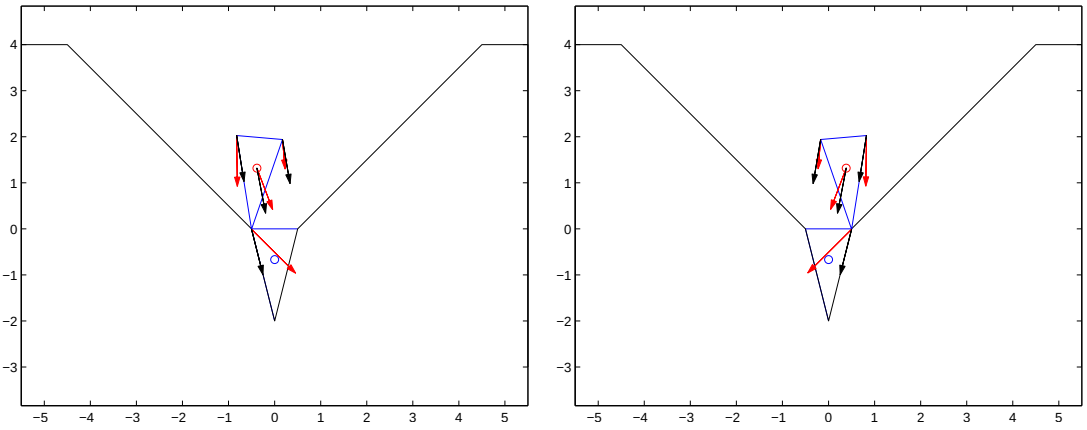
4.2.4 Dominating Contact States

The configurations shown in Figures 4.10 represent the contacts which are considered as having the worst motion quality for the triangular peg assembly. This means that the quality of motion of these extremals determines the overall performance of the matrix, and is the largest of the evaluated set of configurations. For the triangular case these extremals are the E3-V3 and E6-V3 contact states.

These configurations correspond to E-V contacts where the peg is located at the bottom of the chamfer of the fixed part and are about to transition into another contact state. This high deviation from the ideal motion can be explained by being constrained by the edge of the chamfer, and the ideal motion being almost directly down into the hole, resulting in a large discrepancy between them. The geometry then becomes the limiting factor of how close the two motions can be to each other.

4.2.5 V_m Values for Optimization

Figure 4.11 presents the motion quality for the worst-case scenario for triangles of different aspect ratios. There is a clear correlation between the V_m value and the aspect



(a) E3-V3 (b) E6-V3

Figure 4.10: Dominating Contact States for Triangular Peg Assembly. These contact state configurations correspond to transition into a different contact state.

ratio of the part. Unlike the previous rectangular peg results the change in coefficient of friction does have an effect on the objective function results. All optimization, regardless of coefficient of friction, tend towards a value 0.27.

The reason for values plateauing at a certain value can be explained by observing the motion for the dominating extremal at different coefficients of friction and selected aspect ratios. As the part gets longer each subsequent increase in length becomes insignificant; hence, the an increase of one unit has less impact on the results of the optimization.

Unlike the rectangular peg assembly, the admittance selection seems to only be driven by the geometry of the parts being assembled and not the coefficient of friction of the task. For the motions presented in Figure 4.12, the motion of the center of gravity and the contact point is of similar magnitude but that of the top corners is different. Taking the average of the discrepancy of all motions shown corroborates the results obtained by the optimization. The shorter peg has a V_m of 0.34 while the large peg has one of 0.41. This means that the friction does not reduce the space of possibles admittances that satisfy the error-reducing conditions to the point where the admittance resulting in close to ideal motion is no longer available.

4.2.6 Friction Coefficient

The effect of a change in friction is corroborated when observing the effect of the inclusion of two point constraints into the optimization. Wiemer [25] showed that these

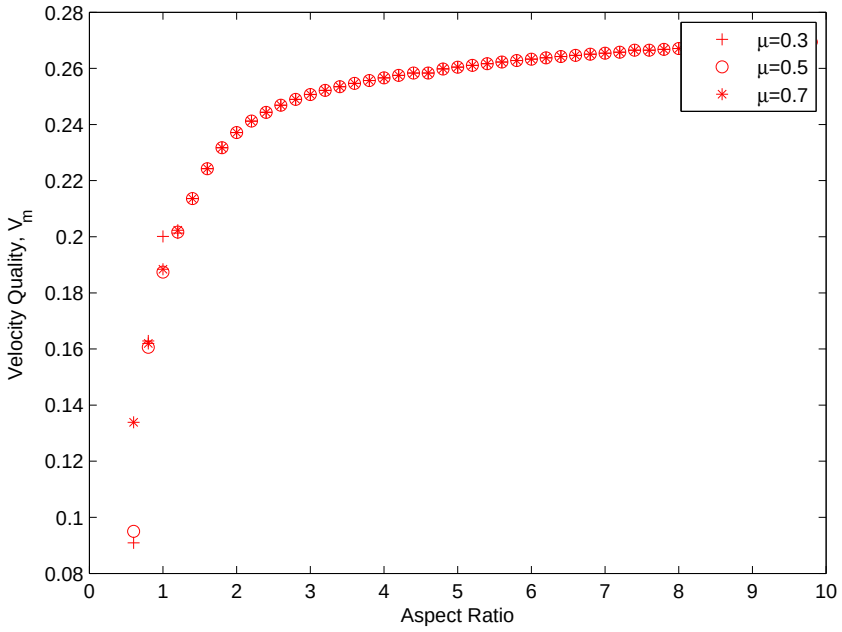
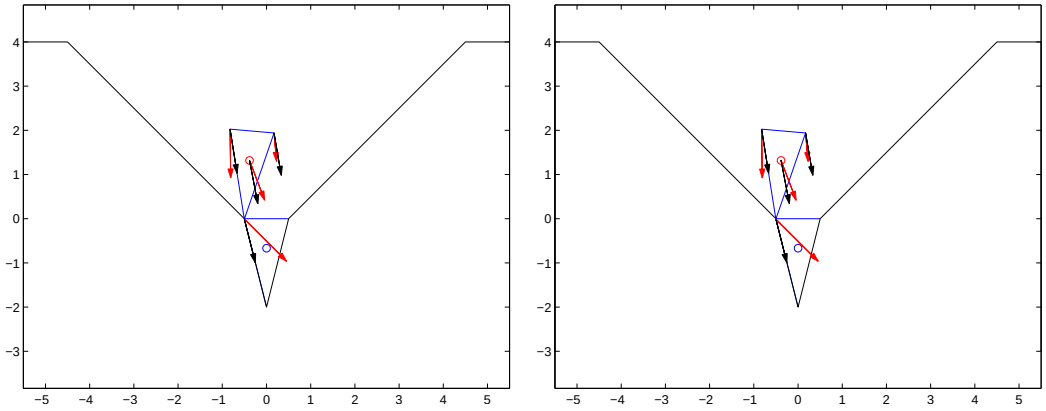


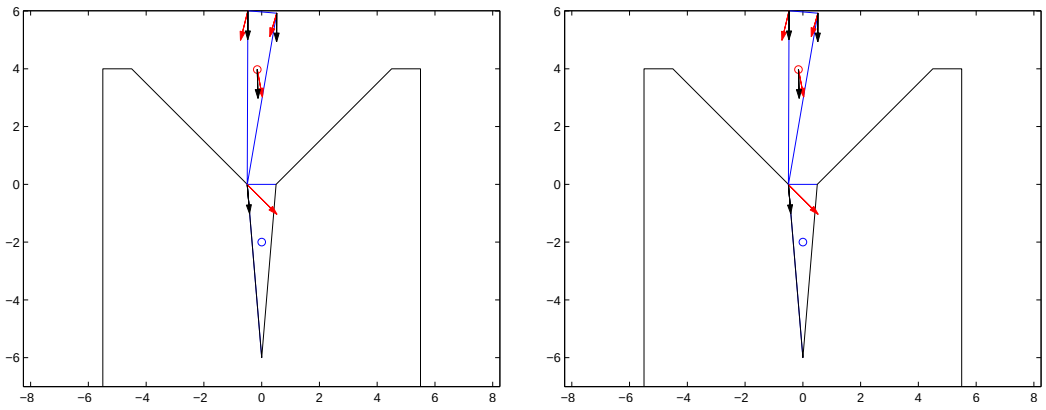
Figure 4.11: Velocity Quality Results for Triangular Peg Assembly. The change in the coefficient of friction causes does not affect performance.

conditions impose the conditions that are the hardest to satisfy for an assembly task. Figures 4.13 and 4.14, present the results obtained for an optimization where the constraints arising for two point contacts are not present. In both of these figures the two point constraints seem not to have an impact on the results pointing, once again, to the friction not having a significant impact on the obtained results. However it is important to mention that while it does not affect the quality of the resulting motion, it does have an impact on the magnitude of the contact forces being experienced in the system.

Comparing the velocity metric results to the friction based results shown in Figure 4.15, also corroborates these findings. If the coefficient of friction significantly reduced the space of possible admittances, the values for the velocity optimized admittance with a 0.7 coefficient of friction shown in Figure 4.6 would resemble those in Figure 4.15. Significant solution space reduction is not expected until coefficients of friction greater than 0.8 are reached.



(a) $L/W = 2.0$ $\mu = 0.3$ (b) $L/W = 2.0$ $\mu = 0.7$



(c) $L/W = 6.0$ $\mu = 0.3$ (d) $L/W = 6.0$ $\mu = 0.7$

Figure 4.12: Resulting Velocities for Dominating Contact State in a Triangular Assembly. Change in friction coefficients do not alter the behavior of the results.

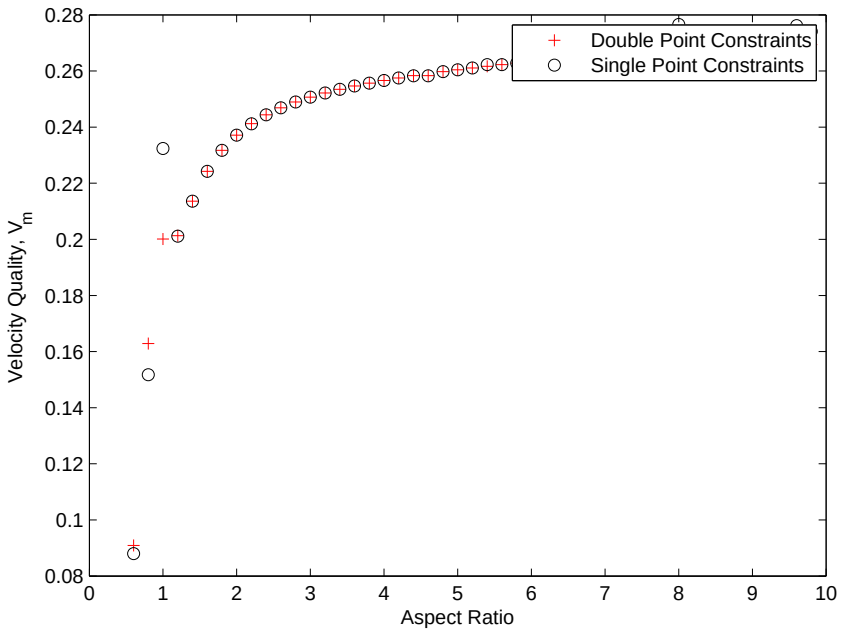


Figure 4.13: Motion Quality (V_m) Results for $\mu = 0.3$ with Varying Constraints for Triangular Peg Assembly. The effect of the coefficient of friction is not significant.

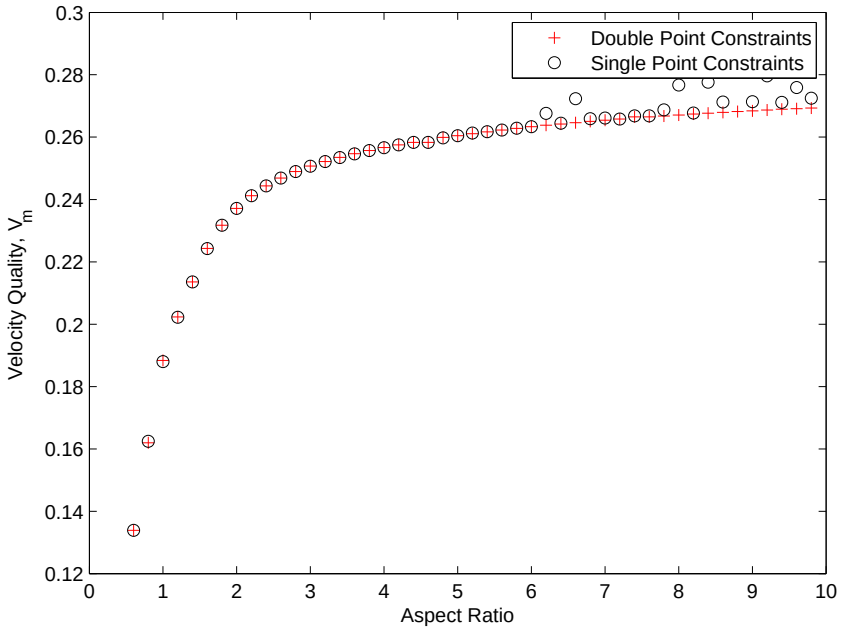


Figure 4.14: Motion Quality (V_m) Results for $\mu = 0.7$ with Varying Constraints for Triangular Peg Assembly. The effect of the coefficient of friction is no significant.

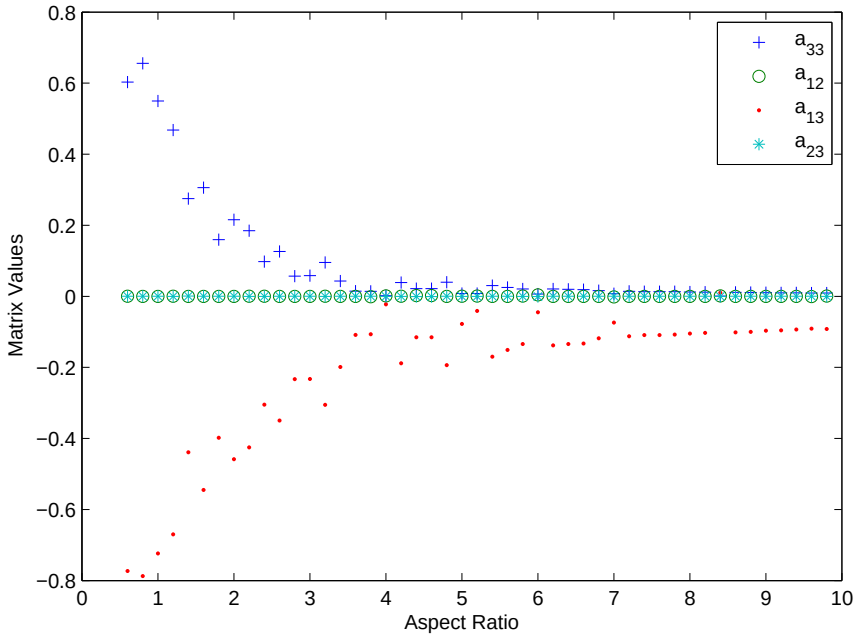


Figure 4.15: Admittance Characteristics for Triangular Assembly Using a Maximized Friction Optimization.

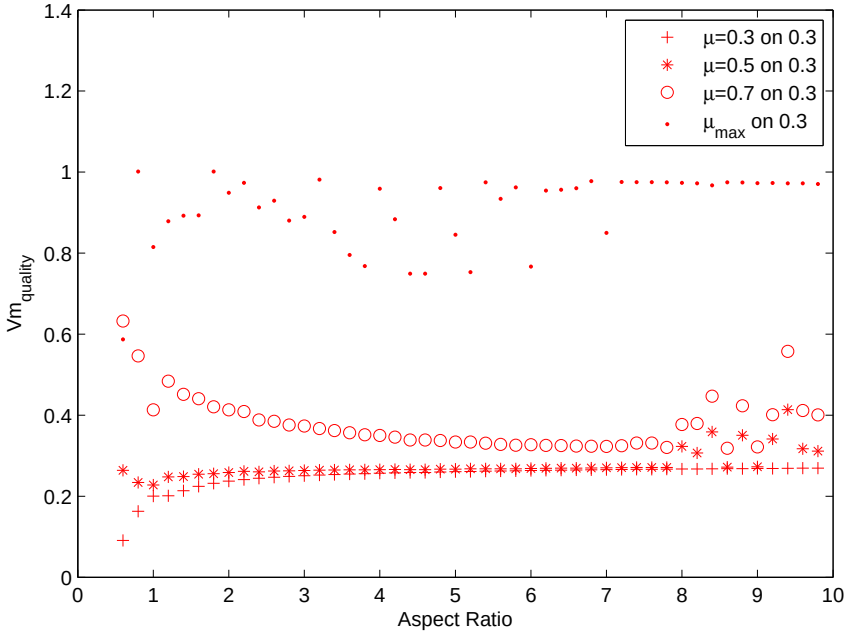


Figure 4.16: Resulting Motion Quality for Triangular Peg Assembly for $\mu = 0.3$. The velocity metric outperforms the friction based optimization.

4.3 Comparison between Optimization Routines

To determine the effectiveness of the velocity metric based admittance selection procedure a comparison to the previously used optimization is needed. Figure 4.16 presents the resulting motion quality for a selected coefficient of friction when the admittance used for error corrective motion was selected using a different coefficient of friction. As it was expected the optimization routine based on ideal velocity generates admittances resulting in motion for the extremal configurations that is of higher quality than the admittance generated by the friction based selection strategy.

These results demonstrate that the procedure suggested in this project is capable of identifying better performing admittances resulting in close to optimal behavior for the triangular peg assembly, than the previously used method of maximized friction.

4.4 Discussion of Results

The results presented in this section indicate that for a triangular peg assembly, the aspect ratio of the assembled part has the most impact on the selection of the admittances components and the motion quality achieved by them. Changes in the friction present in the assembly task does not seem to have significant impact on the performance of the system or admittance component value selection.

The comparisons between selection strategies indicate that the admittance matrices obtained by the velocity matrix, especially for lower coefficients of friction, perform better than the maximized friction approach by obtaining higher quality motion for the extremals considered. Furthermore it has a significant advantage over that approach as its resulting motions will be close in magnitude towards the proper assembly position, this is not true for the friction based approach, which sometimes yields solutions that almost violate the constraints (including error reduction). Because of this it is expected that these matrices will be more robust and desirable for assembly tasks with friction coefficients less than 0.7.

There is still a possibility that for a given configuration the result of the velocity metric could perform worse than that of the friction based approach. However, the velocity will still be error-reducing, and will always result in successful assembly even if it is accomplished in a longer time frame due to constrain satisfaction. Furthermore, investigation presented in Chapter 6 provide evidence suggesting that the quality of the motion of intermediate configurations are bounded by the extremals selected.

4.5 Summary of Chapter 4

In this chapter the results for a triangular peg assembly where discussed. It is demonstrated that the use of a velocity metric as the objective of a constrained minimax optimization results in a satisfactory compliant system that both has a relatively high quality motion at the extremals and error reduction motion for all possible configurations.

The resulting admittance matrix performs better than that obtained the previous friction based program. This holds true for coefficients of friction for which admittance behavior was designed as well as friction coefficient lower than its target.

When examining the admittance, the resulting motion for the extremal configurations is closer to the intended nominal velocity when using the velocity metric than when the maximized friction approach is used. This provides a manner by which the direction and magnitude of corrective motion of the parts can be controlled. The results obtained for a triangular peg differ from those of a rectangular assembly as they are purely driven by the geometry of the parts being assembled and not the friction coefficient.

The following chapter presents results for a stake shaped peg. It will be shown that the velocity metric based optimization successfully generates force-guidance with close to optimal motion for that geometry.

5 STAKE PEG ASSEMBLY

To evaluate the velocity metric based admittance selection process, the results for different geometries are needed. Previous chapters presented results for both rectangular and triangular shaped peg assembly tasks. This chapter presents the optimal admittance results for a stake peg assembly obtained using the corrective motion quality as the objective function. The peg assembly consists of a stake shaped peg held by a manipulator being inserted into a fixed chamfered stake shaped hole. The results were obtained for a range of different aspect ratios. Relationships between the part geometry and optimal admittance are identified. Results also demonstrate improvement over the previously used friction based optimization by obtaining admittances with higher quality corrective motion.

First the variation in part geometry is described. Next, results for the velocity metric based optimizations are presented and trends are investigated. These trends include the behavior the admittance components, the resulting quality of the best worst case motion and the location of the compliance center for the optimal admittances. Finally, the admittance behavior obtained using this approach is compared to the results for the maximized friction selection strategy to determine the effectiveness of such approach.

5.1 Assembly Description

This section identifies part geometry and part contacts considered in generating the optimal admittance for a stake peg assembly. Related items include the part clearance, successful assembly conditions, and contributing contact states.

Figure 5.1 shows the dimensions of the geometry investigated. The variation in geometry used is the aspect ratio defined as L/W, and l/W. L and l are normalized with respect to W, to obtain a clear picture of the behavior of the admittance without having to consider units. The aspect ratio of l/W was fixed to always be 3 for all geometries investigated. This was done in order to compare the results to those previously obtained in [25].

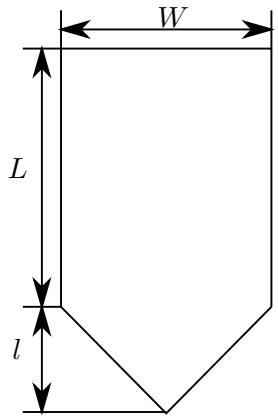


Figure 5.1: Stake Peg Dimensions. Aspect ratio L/W is normalized with respect to the width of the peg.

5.1.1 Contact State Enumeration

Figure 5.2, presents the corresponding edge and vertex numbers for both the fixed and movable part.

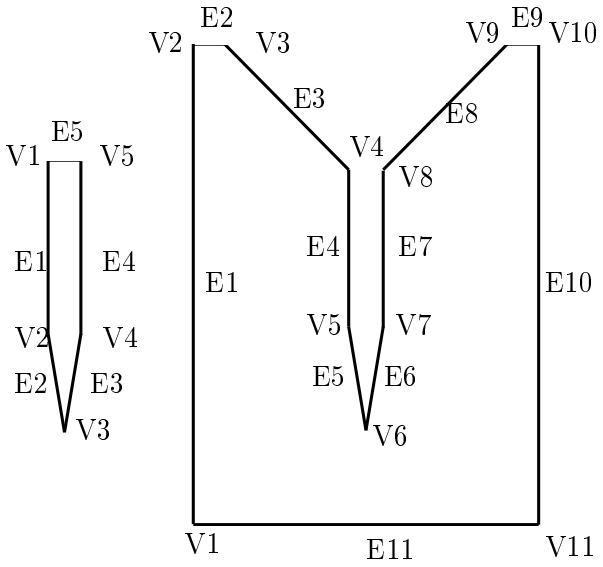


Figure 5.2: Feature Enumeration for Stake Assembly.

In order to account for the misalignment present on the robot positioning the following bounds where selected:

- $\pm X_{RB} = [-1.87, 1.87]$
- $\pm Y_{RB} = [0, 24.5]$
- $\pm \Theta_{RB} = [-\frac{\pi}{36}, \frac{\pi}{36}]$

where X_{RB} and Y_{RB} are unit less, and Θ_{RB} is expressed in radians . These error bounds are selected as conservative estimates of the misalignment experienced by the robot performing the assembly. It is expected that any current industrial manipulator will possess error positioning much lower that the selected bounds.

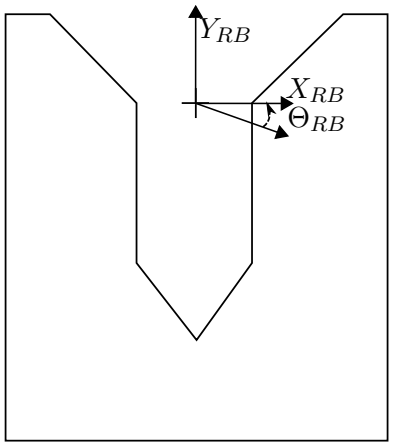


Figure 5.3: Frame Used as Basis to Describe Misalignment Bounds in Stake Peg Assembly.

The selected clearance is only applied on the x direction of the rectangular shaped area of the hole and is selected as as 0.10 units.

A configuration is defined as successful assembly if the location of the top of the peg is within the selected bound:

- $y_{succesful} = \{0, 0.2\}$

It is necessary to select this bound in order to adjust the process for difficult configurations; for example situations where the ideal motion corresponds to complete horizontal movement, or where the peg is situated at the bottom of the hole and is misaligned by the clearance.

For a stake peg assembly the following contact states are not considered successful assembly and contribute to the optimization process and constraint generation.

- V4-E1
- V4-E2
- V8-E3
- V8-E4
- E3-V3
- E4-V2
- E5-V2
- E5-V3
- E6-V3
- E6-V4
- E7-V4
- E8-V3
- V4-E1 E5-V2
- V4-E1 E6-V3
- V8-E4 E5-V3
- V8-E4 E6-V4

Most of the contacts states are similar to those encountered for the triangular peg assembly and the rectangular peg assembly. However there is a significant increase in the number of the contact states that have an effect on the optimization routine. Since each contact state is the further decomposed, this results in a significantly large increase in the number of constraints being applied on the optimization (close to 2000). This can lead to the optimization routines not being capable of converging to a single value and taking a longer time to compute.

5.2 Results

The following results are obtained for the values of the admittance components for different coefficients of friction as the aspect ratio of the part is increased. These components include a_{33}, a_{12}, a_{13} and a_{23}. These components represent the mapping of forces/-torque into a given direction as explained in Chapter 1. These represent the values of the admittance components that were identified as those resulting the best worst-case scenario for motion quality of the extremals evaluated.

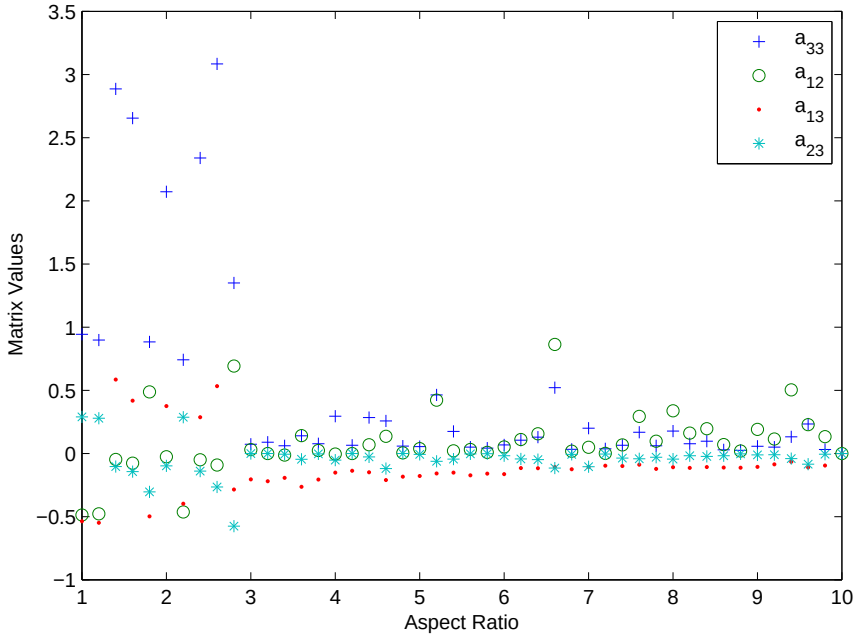


Figure 5.4: Admittance Characteristics for Stake Peg Assembly Using a Coefficient of Friction of 0.3.

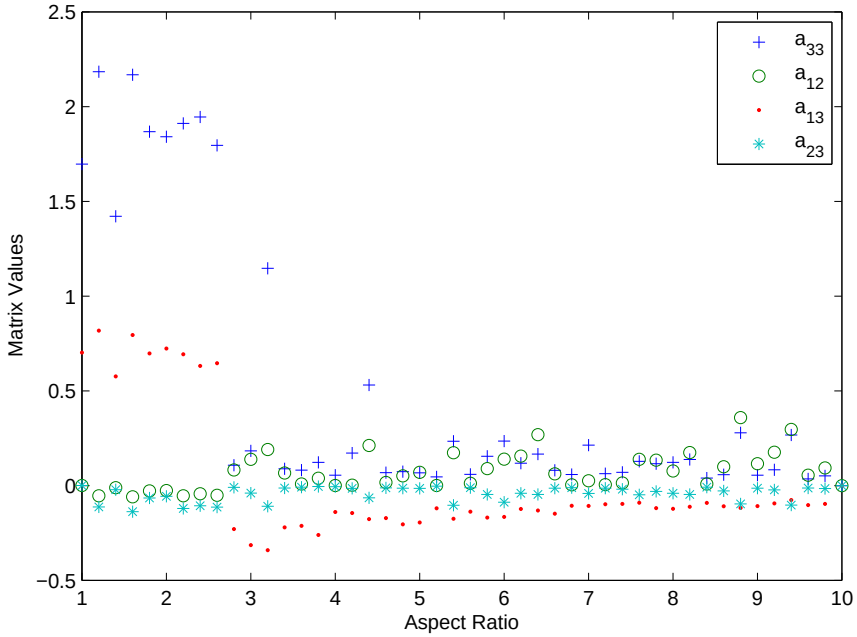


Figure 5.5: Admittance Characteristics for Stake Peg Assembly Using a Coefficient of Friction of 0.5.

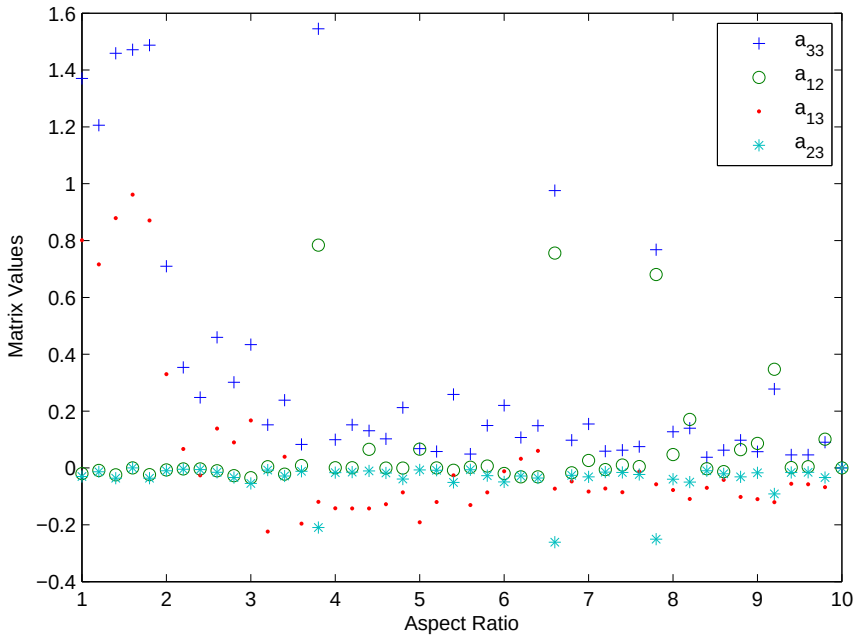


Figure 5.6: Admittance Characteristics for Stake Peg Assembly Using a Coefficient of Friction of 0.7.

As with the results obtained for the rectangular and triangle peg assemblies, the values of a_{22} are not presented as they do not follow an identifiable pattern.

For the stake geometry, the patterns for components selection are harder to identify than in previous cases. This can be attributed to the significant increase in the number of contacts states affecting the optimization routine. It is important to note that due to the non linear nature of the optimization the results are quite noisy. This can be solved by running subsequent optimizations on the selected results and varying the population used for the algorithm.

Change in aspect ratio does have a significant impact on the obtained results as seen in 5.4, 5.5, and 5.6, which present the values of the admittance components with an increase in aspect ratio. The components of the admittance tend to converge to certain values as was the case with the previous geometries. With a_{12} and a_{23} seemingly converging to small values close to zero. a_{33} remains positive approaching 0.1 and a_{13} remaining negative for all aspect ratios approaching -0.1. As with previous geometries, as the aspect ratio the space of admittances satisfying the error reducing conditions becomes smaller. Because of the reduction in possible solutions, the quality of the motions also becomes worst as aspect ratio increases, as those providing the best motion are no longer available.

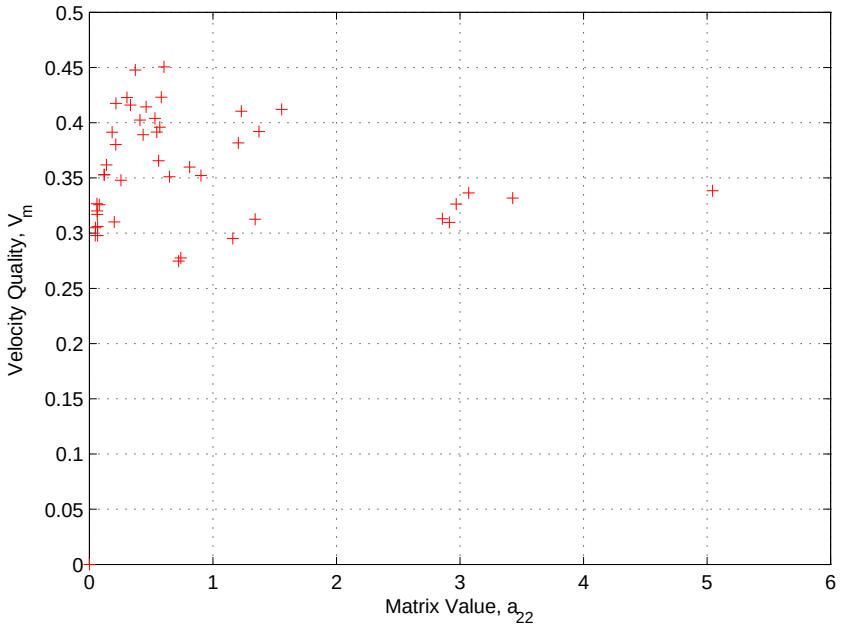


Figure 5.7: Relation between a_{22} and V_m for Stake Peg. The change in the value of a_{22} does not seem to have an effect on the quality of the resulting motion for a Friction Coefficient of 0.3.

There is also a significant discontinuity occurring at an aspect ratio of 3. This can be explained by noticing that at this point further two point contacts become available, increasing the difficulty of identifying an error-reducing admittance.

The increase in the static coefficient of friction of the assembly task, does not seem to have a significant impact on the values of the admittance components selected. This can be explained by observing that the behavior of the assembly is mostly dictated by the triangular bottom mirroring the results for the triangular peg, admittance components being affected by geometry more than by friction coefficients.

5.2.1 a_{22} Component

As stated the value of a_{22} (the mapping of forces in the y direction to motions along that same direction) does not seem to follow a discernible pattern. The value of V_m seems to be largely unaffected by the value of a_{22} as shown by Figure 5.7. As found in [25] a_{22}'s main role is in generating admittance matrices that are positive definite, which can be realized with a passive mechanism.

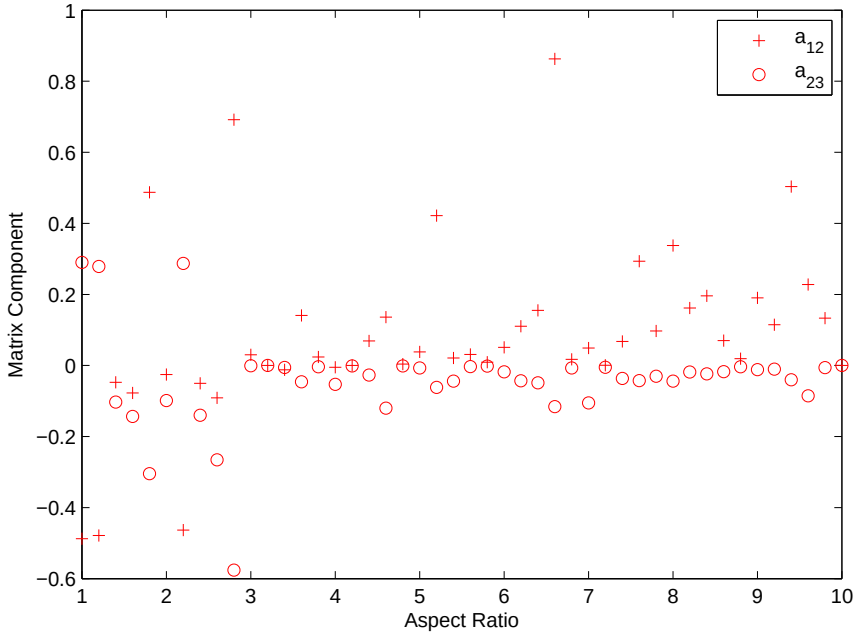


Figure 5.8: Resulting a_{12}, a_{23} Components for Varying Aspect Ratio for Stake Peg Assembly. The values of the results approach zero due to part symmetry for some of them but present significant variation.

5.2.2 a_{12}, a_{23} Components

Figure 5.8 presents the value of the a_{12} and a_{23} components as a function of aspect ratio. It can be observed that the value for a_{12} and a_{23} does not behave exactly like those in the previous section. While the values of a_{12} and a_{23} do approach zero for a significant amount of aspect ratios, several other cases present values much larger than anticipated. Once again this is a result of the increase in the difficulty of the optimization routine for an extremely constrained task which contains several local minima.

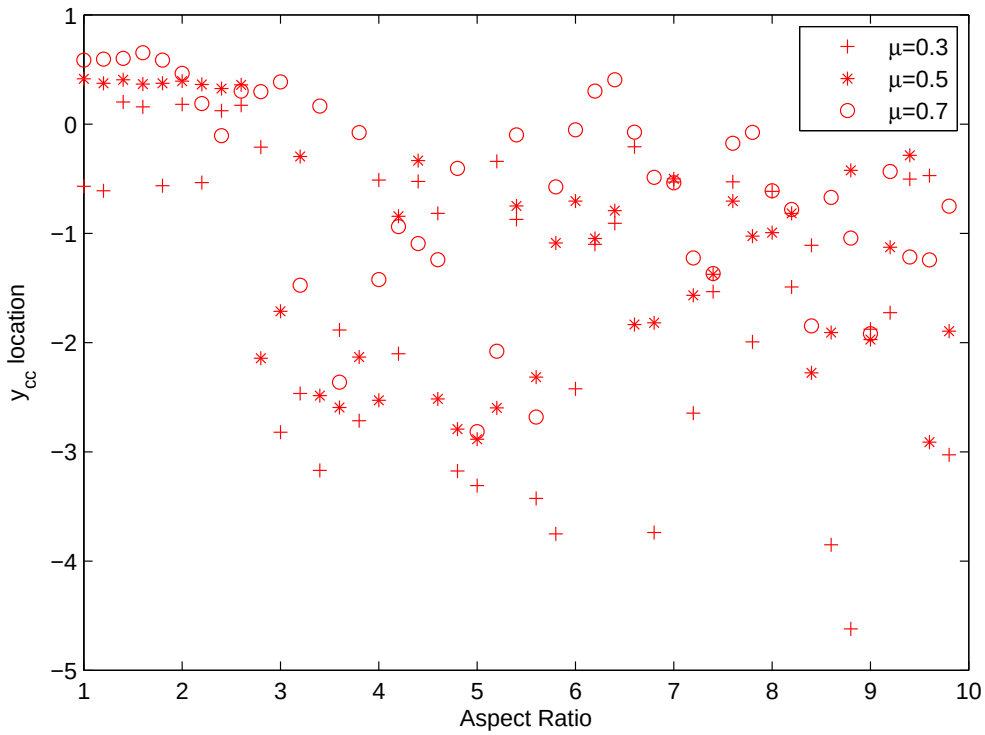
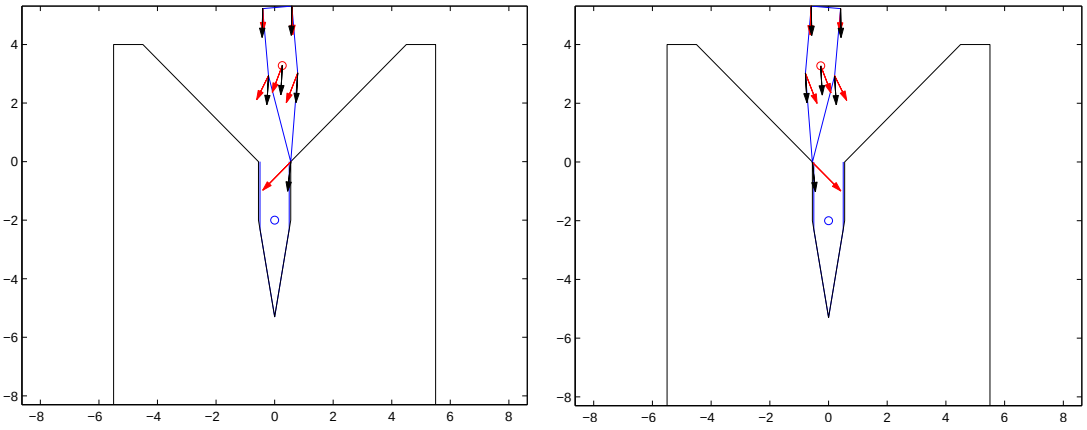


Figure 5.9: Resulting y_{cc} Location for Stake Peg Assemblies. The location of the compliant center seems to be arbitrary.

5.2.3 a_{13}, a_{33} Components

The ratio of a_{13} and a_{33} determines the location of the compliant center along the y direction, expressed as y_{cc}. Several values of a_{13} and a_{33} result in the same y_{cc}, hence it is not unique resulting in a solution space with multiple local minima. Figure 5.9 presents the location of the compliant center in the y direction as the aspect ratio of the geometry is increased. In previous sections the location of the y_{cc} moved downward with relation to the change in aspect ratio, this pattern is not present for the stake peg assembly. While the location of y_cc does move downward it does not follow a discernible pattern. The discontinuity that exists at an aspect ratio of 3 cause by the inclusion of new two point contacts has a significant effect. The investigation in [25] for stake shaped pegs also presented similarly varying compliant center locations.



(a) E8-V3 (b) E3-V3

Figure 5.10: Dominating Contact States for Stake Peg Assembly. These contact state configurations correspond to transition into a different contact state.

5.2.4 Dominating Contact States

The following contact states are identified as those whose extremal configurations are considered the worst-case scenario for the Stake Peg Assembly. These contact state are the E3-V3 and E8-V3 contact states.

These contact states refer to configurations shown in Figure 5.10 where the peg is located at the bottom of the chamfer on the fixed part and is about to transition into another contact state. This can be explained by the motion of the contact point being constrained to the direction of the edge of the chamfer, and the ideal motion being almost directly down into the hole, hence limiting how close the the resulting and ideal motions can be to each other.

5.2.5 V_m Values for Optimization

Figure 5.11 presents the obtained motion quality for the worst-case scenario for different aspect ratios and friction coefficients. The results for the value of the objective function V_m seem to follow a particular pattern, albeit a noisy one. The solutions behave similarly to that identified for a triangular peg, seemingly limited by the part geometry. It seems that the coefficient of friction does not significantly affect the results, which is consistent with the results for a triangular peg. However unlike the triangular peg, there does seem to be slightly better motion quality at lower coefficients of friction. For aspect

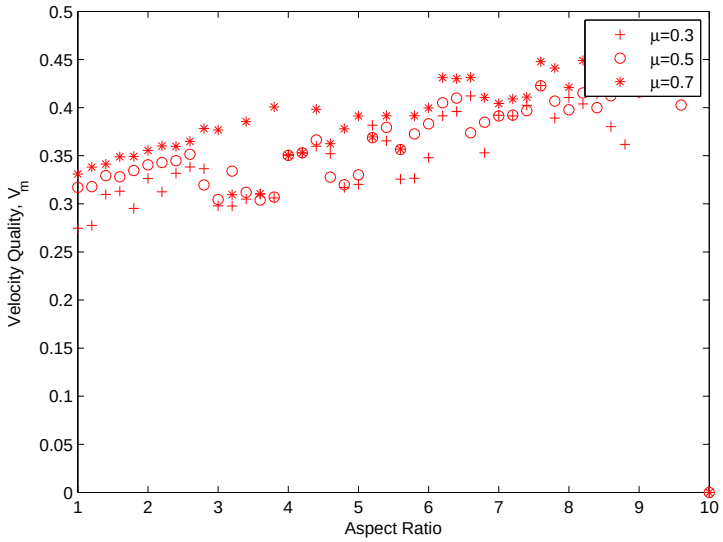
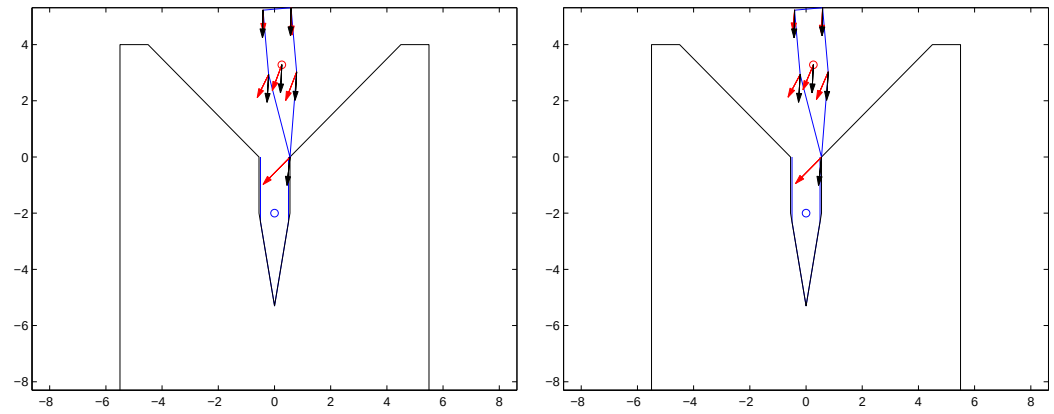


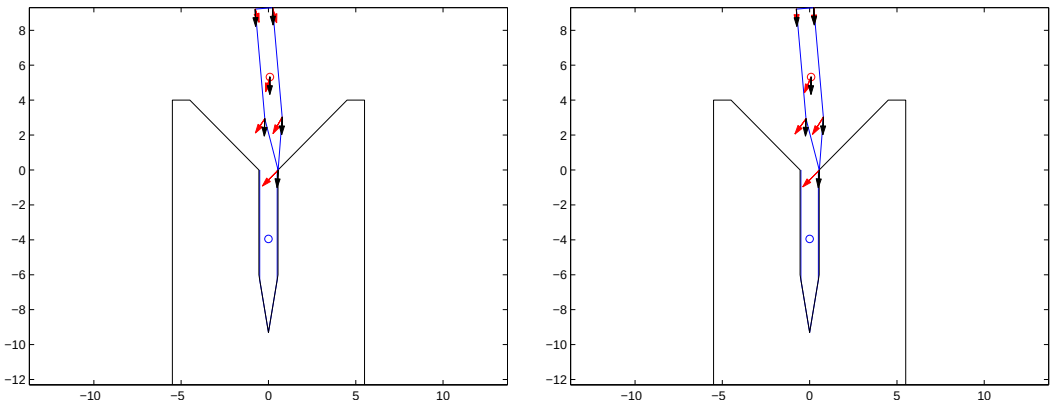
Figure 5.11: Velocity Quality Results for Stake Peg Assembly. The change in the coefficient of friction does not seem to have significant impact on the results.

ratios lower than 3, the behavior is consistent and corresponds to the optimization being easier to perform due to the lower number of contact states evaluated. However as more contact states become possible, the optimization becomes numerically more difficult to perform resulting in the noise present. However it is important to note that the motions still possess satisfactory motion quality.

Looking at the resulting motions shown in Figure 5.12 for the worst-case scenario configurations the motion of the configuration is still close to ideal, and would be considered satisfactory, regardless of the noise present in the optimization. As the part length increases, the motion quality becomes less than for the shorter aspect ratios, however the change is not significant.



(a) $L/W = 2.0$ $\mu = 0.3$ (b) $L/W = 2.0$ $\mu = 0.7$



(c) $L/W = 6.0$ $\mu = 0.3$ (d) $L/W = 6.0$ $\mu = 0.7$

Figure 5.12: Resulting Velocities for Dominating Contact State Configuration in Stake Peg Assembly. Friction coefficients do not alter the behavior of the results significantly.

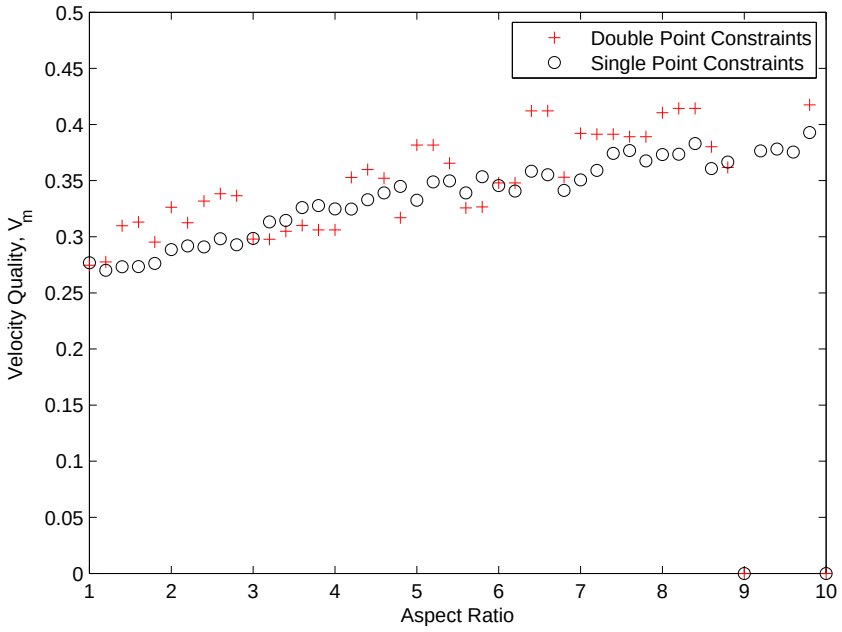


Figure 5.13: Objective Function Results for $\mu = 0.3$ with Varying Constraints. The effect of the inclusion of two point constraints is not significant.

5.2.6 Friction Coefficient

Figures 5.13 and 5.14 show the obtained results for the quality of motion with and without two point constraints, demonstrating that two point contacts do not have a significant effect on the obtained results. This is important as two point constraints are the ones that most affect the optimization routine. Since removing them does not have a significant effect, it can be determined that the optimization routine is being limited by the geometry of the assembled part more than by the static friction.

Figure 5.15 presents the resulting admittance components for the maximized friction components. Comparing them to those of the velocity metric for a high coefficient of friction such as shown in Figure 5.6, it can be observed that while the lower aspect ratio results differ, for longer aspect ratios the results are closer together, due to the space of available admittances satisfying the error-reducing constraints being reduced.

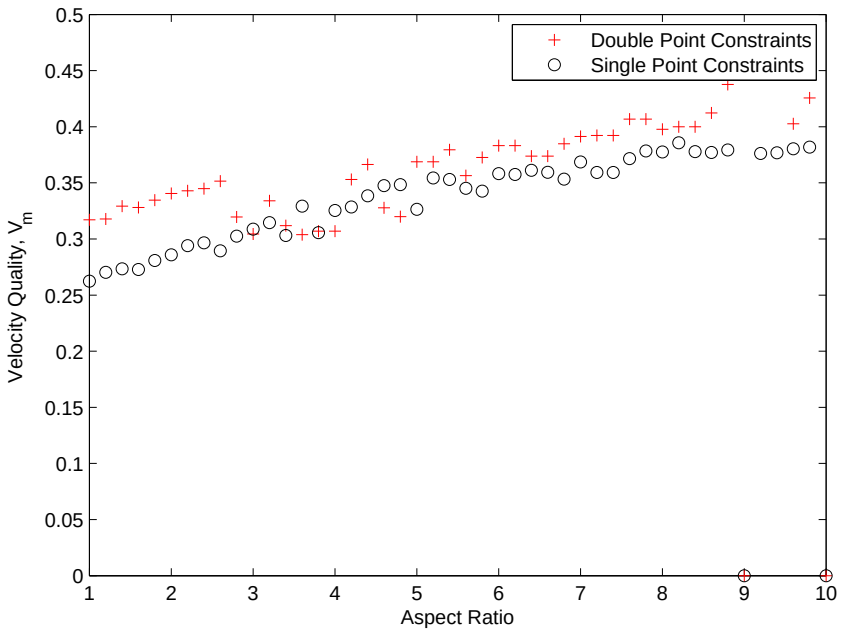


Figure 5.14: Objective Function Results for $\mu = 0.5$ with Varying Constraints for Stake Peg. The effect of the inclusion of two point constraints is not significant.

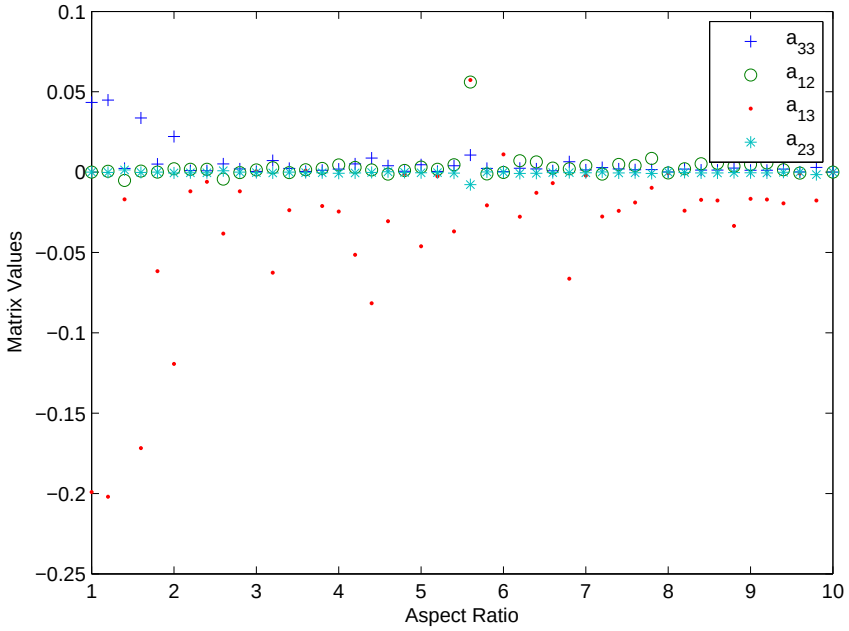


Figure 5.15: Admittance Characteristics for Triangular Assembly Using a Maximized Friction Optimization.

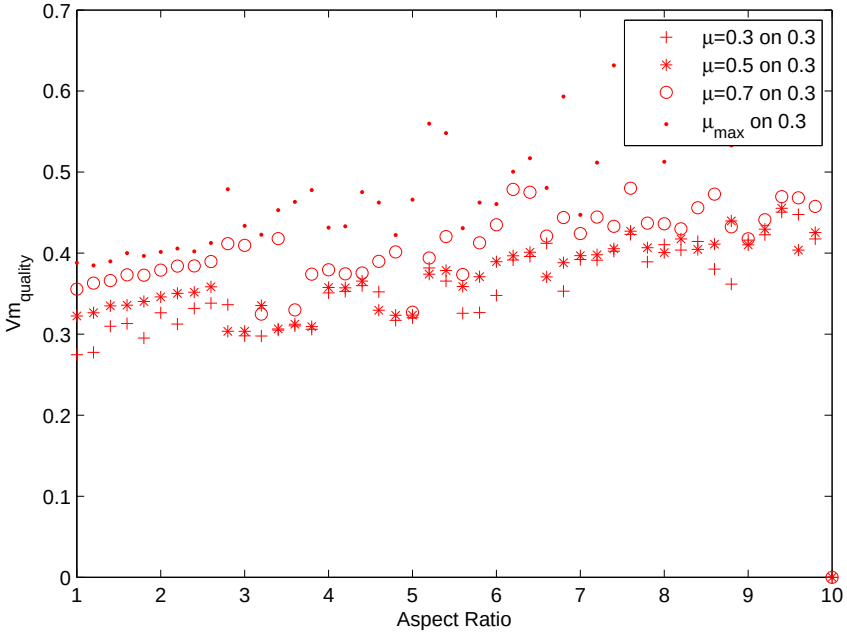


Figure 5.16: Resulting Motion Quality for Stake Peg Assembly for $\mu = 0.3$. The velocity metric outperforms the friction based optimization.

5.3 Comparison between Optimization Routines

Figure 5.16 presents a comparison of the performance of different admittances, which were optimized for a certain coefficient of friction, applied to assembly tasks with a different coefficient of friction.

As it was expected, the optimization routine based on ideal velocity generates motion for the extremals that is considered higher quality when the selected coefficient is close to that for which the admittance was obtained, as observed by 0.5 optimized used on an assembly with $\mu = 0.3$. Applying the optimized results to other lower friction coefficients obtains better results than those of the maximum friction optimization all cases investigated. This provides evidence suggesting the use of the velocity metric as opposed to the maximized friction as the objective function of the optimization. Once again the discontinuity at an aspect ratio of 3 can be explained by the inclusion of two point contact states that greatly increase the complexity of the optimization routine resulting in noisier results.

5.4 Discussion of Results

From the comparisons presented it becomes obvious that the velocity metric performs better when applied to different friction values, especially when optimized for a given friction value, than the maximum friction based optimization. However as opposed to the previously investigated geometries, there is significant noise in the data presented. The increase in the number of contact states and configurations greatly increases the number of constraints needing to be satisfied. In most optimizations the constraint closer to be violated is caused by two point contact states that occur inside the hole.

The optimization is, regardless of the noise present, successful in identifying admittance behavior resulting in high quality motion for its extremals. It is also, as will be shown in Chapter 6, capable of generating high quality motion for intermediate configurations not contained in the set of optimized extremals.

However there is still a possibility that for a given configuration the result of the velocity metric could perform worse than that of the friction based approach. This can be undesirable, however the velocity will still be error reducing, and will always result in successful assembly even if it accomplished in a longer time frame due to constrain satisfaction.

5.5 Summary of Chapter 5

In this chapter the results for a stake peg assembly where discussed. It is demonstrated that the use of a velocity metric as the objective of a minimax optimization results in a satisfactory compliant system that both has a relatively high quality motion at the extremals and error reduction motion for all possible configurations.

The resulting admittance matrix performs better than the previous friction based program. This holds assembly tasks with with a coefficient of friction for which the admittance was designed for, as well as for lower coefficients of friction.

When examining the admittance, the resulting motion for the extremals is closer to the intended nominal velocity using the velocity metric than the friction approach. As a result, the admittance presents a more controllable motion, which possess a motion closer than that intended. This finding allows admittance generated by this procedure

to be applied to assembly task in an industrial setting with more ease. However there seems to be more noise present in the optimization of stake peg assemblies, and further investigation should be performed.

The following chapter presents further investigation into the results obtained for the quality of motion for configurations not contained in the optimized set.

6 MOTION QUALITY FOR INTERMEDIATE CONFIGURATIONS

As shown in the previous chapters the velocity metric based admittance selection procedure is capable of generating admittance behavior resulting in close to optimal motion for a selected set of extremals. This section will present results obtained from an additional numerical investigation investigations into the quality of the resulting motion for configurations not present in the optimized set. The numerical investigation was performed for all the geometries considered in previous sections (rectangular, triangular and stake pegs), providing evidence to support the use of the velocity metric based admittance selection procedure.

6.1 Motion Quality for Selected Configurations

This section presents the resulting motion quality for configurations not contained within the set of configurations for which the constrained minimax optimization was performed. This is done to support the use of the developed velocity metric based process for admittance selection. The first figures present the motion quality results for intermediate configurations bounded for the extremals of the contact state identified as dominating in Chapters 3, 4, and 5 [1]. These configurations are those which the movable part is located at the bottom of the chamfer of the fixed part and has rotated pasts its proper assembled position. The resulting motion quality of another contact state located inside the hole of the fixtured part will also be presented. The first section presents the results obtained for the rectangular peg assembly.

6.1.1 Rectangular Peg Assembly Motion Quality

Figures 6.1 and 6.4 presents the obtained results obtained for the quality of motion of an intermediate configuration whose motion is the result of the application of the planar control law 1.2 using an admittance generated from the proposed velocity metric

[1] All investigations use a friction coefficient of 0.3 and an aspect ratio of 6.

optimization routine. Figure 6.1 presents the obtained results for intermediate configurations of what is considered the dominating extremal for a rectangular peg assembly task (E7-V4). The z axis presents the obtained quality of motion, while the x and y axis present the values of the two variables used to determine the configuration of the contact state. Figure 6.2 presents a graphical representation of the corners of x and y space for Figure 6.1. The corners are the extremals of the contact state considered.

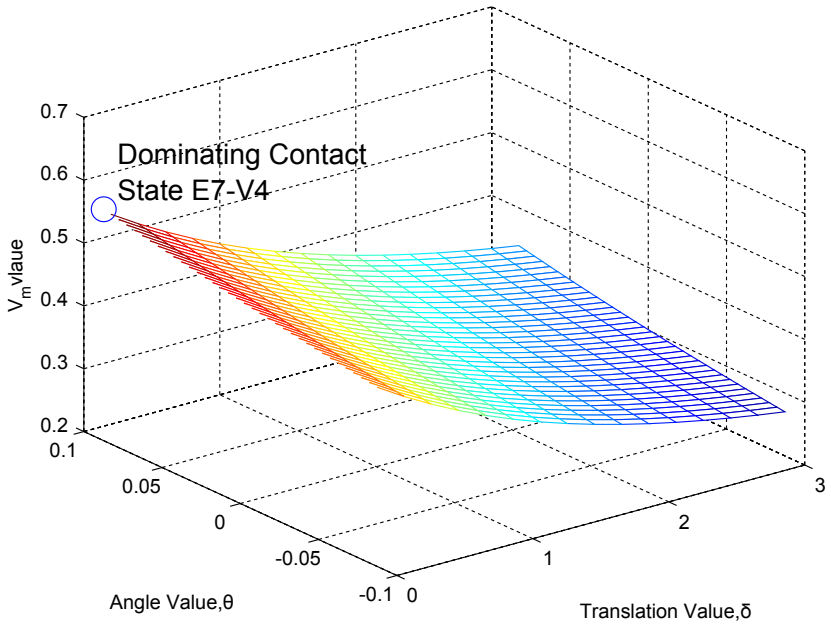


Figure 6.1: Surface Plot for Rectangular Peg Dominating Contact State E7-V4. The maximum value for V_m is obtained at the dominating (sampled) configuration. Extremal configurations correspond to corners of the rectangular surface.

As can be seen the intermediate configurations, meaning those not located on the corners of the area are found to have higher quality than what is identified as the worst-case scenario (at δ_{min}, θ_{max}). This points to the value of the quality of motion for an assembly task being bounded by the value identified as the worst extremal.

This can be further corroborated by looking at another contact state, in this case one located within the hole of the fixtured part (E6-V4). The results are presented in Figure 6.4, as with other figures in this chapter negative values are assigned to configurations that result in penetration of the parts. The maximum value obtained for the configurations in this extremal is considerable smaller than the value identified by the optimization routine for the worst-case scenario.

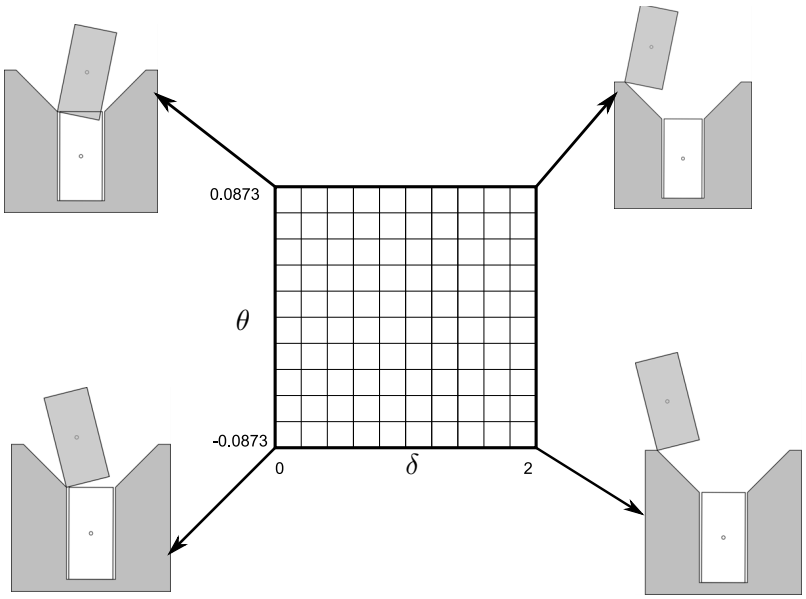


Figure 6.2: Representation of Configurations Located on the Corners of the Evaluated Configuration Space for Rectangular Pegs. Each configuration corresponds to an extremal of the contact state E7-V4.

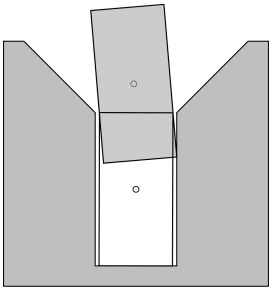


Figure 6.3: Graphical Representation of Contact State E6-V4.

The following sections present similar results for both the triangular and stake peg assemblies.

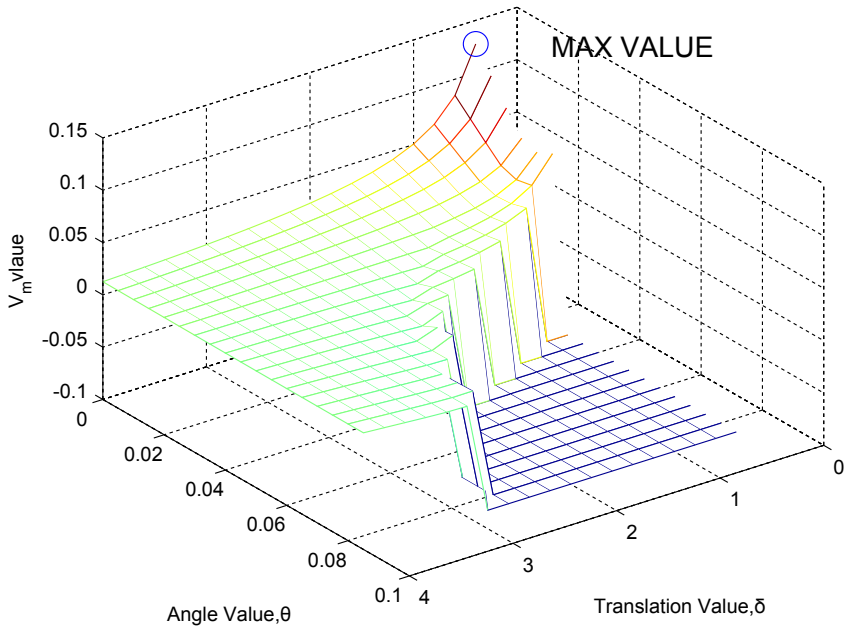


Figure 6.4: Surface Plot for Rectangular Peg Contact State E6-V4. The maximum value for V_m is not higher than that of the dominating (sampled) configuration.

6.1.2 Triangular Peg Assembly Motion Quality

As done in the previous section, Figures 6.5 and 6.8 present the obtained results for the quality of motion of an intermediate configuration whose motion is the result of the application of the planar control law 1.2 using an admittance generated from the proposed velocity metric optimization routine. Figure 6.5 presents the obtained results for intermediate configurations of what is considered the dominating extremal for a triangular peg assembly task (E6-V3). The extremals of the space presented are shown in Figure 6.6.

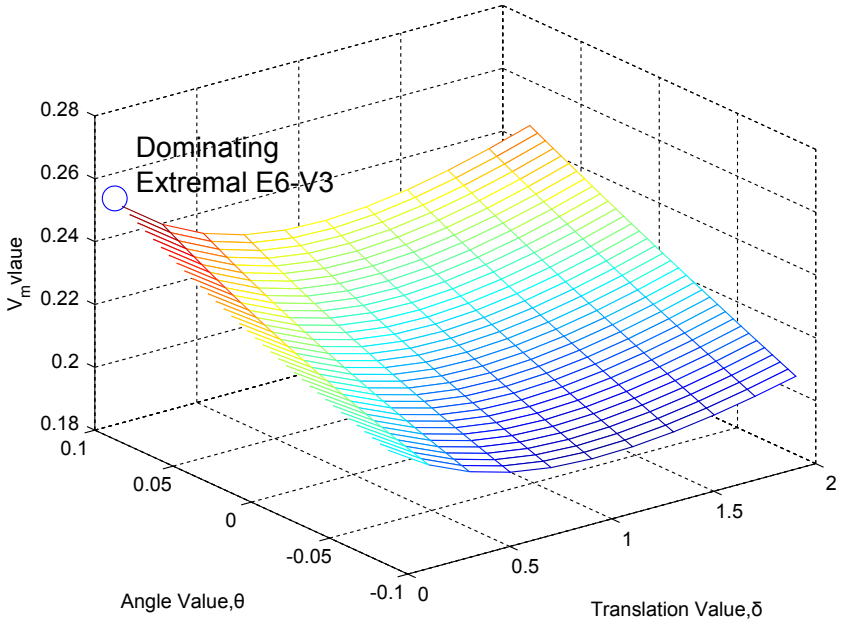


Figure 6.5: Surface Plot for Triangular Peg Dominating Contact State E6-V3. The maximum value for V_m is obtained at the dominating (sampled) configuration. Extremal configurations correspond to corners of the rectangular surface.

As with the results for the rectangular peg assembly the intermediate configurations, found to have higher quality motions than the configurations identified as the worst-case scenario (at δ_{min}, θ_{max}). As with the rectangular peg assembly investigation this point to the validity of using the velocity metric based process for admittance selection.

Looking at a contact state (E5-V3) located inside the hole of the fixed part, similar results are obtained. As seen in Figure 6.8The maximum value for the motion discrepancy is still considerably lower than tat identified for the worst-case scenario. The negative values for the V_m value correspond to configurations resulting in part penetration, the

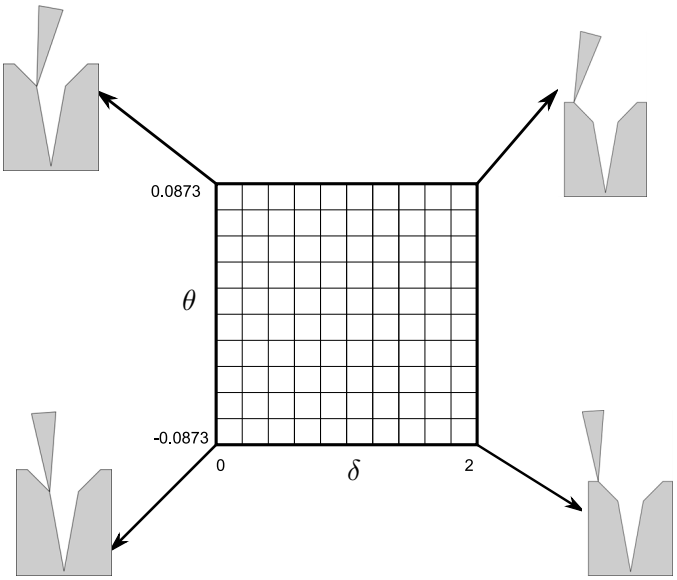


Figure 6.6: Representation of Configurations Located on the Corners of the Evaluated Configuration Space for Triangular Pegs. Each configuration corresponds to an extremal of the contact state E6-V3.

"spikes" seen in the plot correspond to values close to the limit for identifying successful assembly.

The following section presents the same investigation for the stake peg assembly.

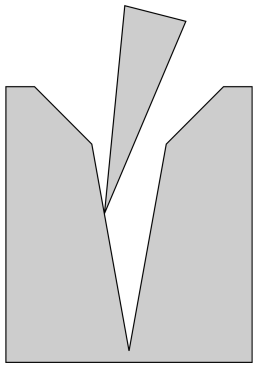


Figure 6.7: Graphical Representation of Contact State E6-V4.

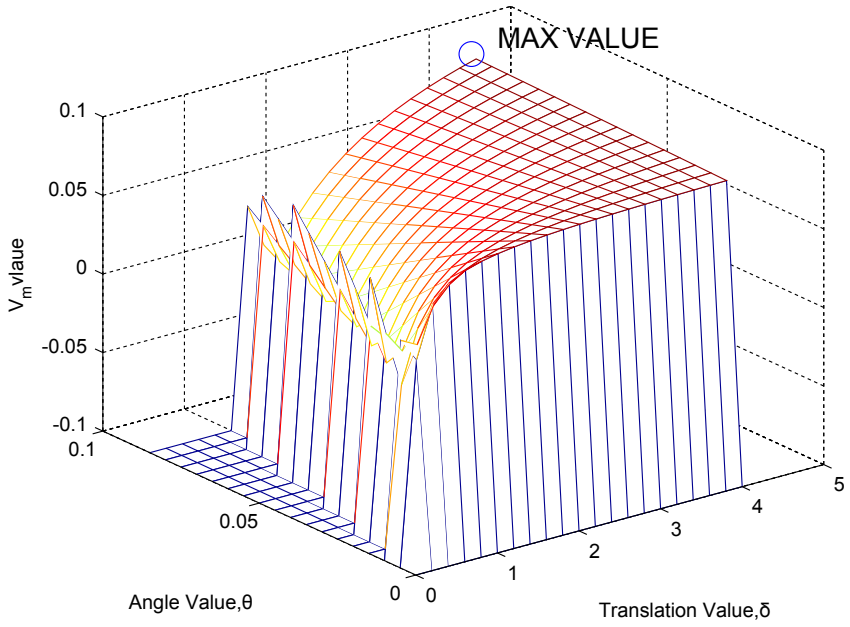


Figure 6.8: Surface Plot for Triangular Peg Contact State E5-V3. The maximum value for V_m is not higher than that of the Dominating Configuration.

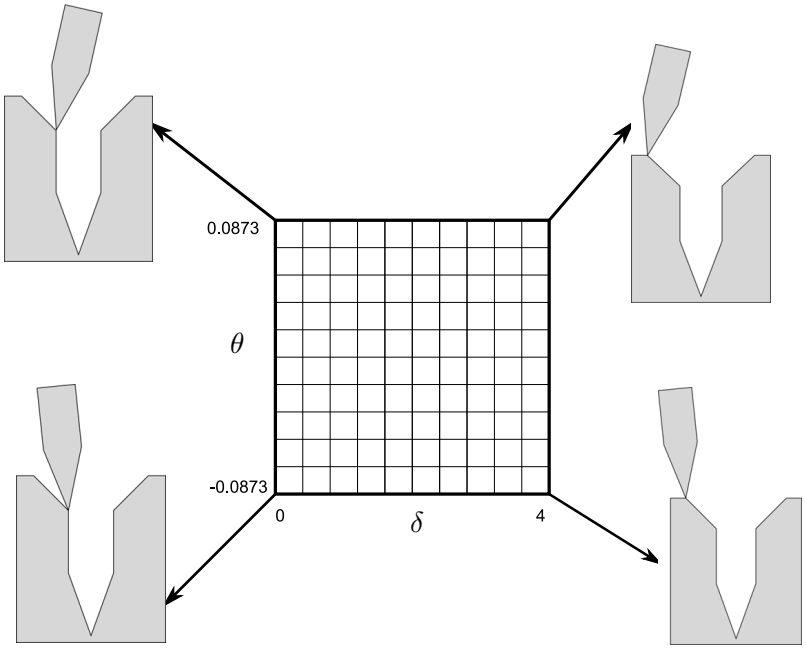


Figure 6.9: Representation of Configurations Located on the Corners of the Evaluated Configuration Space for Stake Pegs. Each configuration corresponds to an extremal of the contact state E8-V3.

6.1.3 Stake Peg Assembly Motion Quality

Figures 6.10 and 6.12 present the obtained results for the quality of motion of an intermediate configuration for the stake peg assembly. Figure 6.10 presents the obtained results for intermediate configurations of what is considered the dominating contact state for a stake peg assembly task (E8-V3). The extremals of the space presented are shown in Figure 6.9.

The intermediate configurations of the contact state are found to have higher quality motions than the configuration identified as the worst-case scenario (at δ_{min}, θ_{max}). As with the rectangular peg assembly investigation this points to the validity of using the velocity metric based process for admittance selection. This result is especially important for the stake peg assembly, whose optimized result contained significant amount of noise. The resulting admittances, even though though noisy, still result in high quality motion not only for the evaluated configurations but also for intermediate ones.

Looking at a contact state (E4-V2) located inside the hole of the fixed part, similar results are obtained. As seen in Figure 6.12 the maximum value for the motion discrepancy is still considerably lower than tat identified for the worst-case scenario. The negative values for the V_m value correspond to configurations resulting in part penetra-

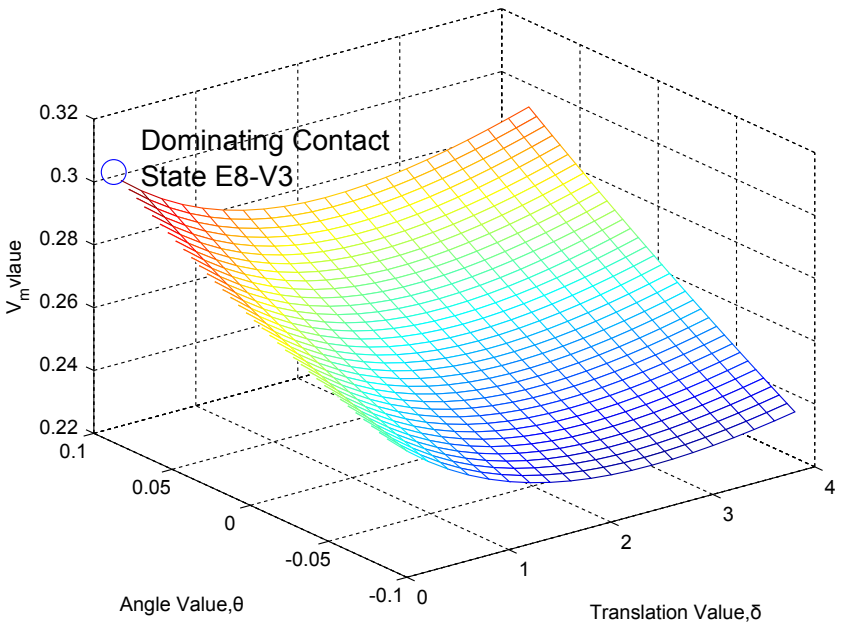


Figure 6.10: Surface Plot for Stake Peg Dominating Contact State E8-V3. The maximum value for V_m is obtained at the dominating (sampled) configuration. Extremal configurations correspond to corners of the rectangular surface.

tion. Once again this demonstrates the validity of the use of the velocity metric based admittance selection procedure.

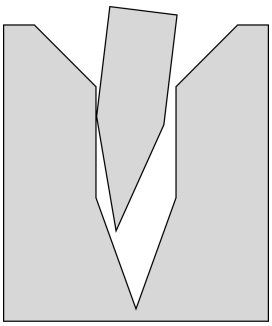


Figure 6.11: Graphical Representation of Contact State E4-V2.

The results obtained for rectangular, triangular, and rectangular peg assemblies are satisfactory. However, it is important to note that even though these results provide significant support for the statement that if the admittance is optimized for the extremals in the configuration the quality of the motion of all intermediate configurations is bounded, it does not guarantee that this will be the case. This is only a numerical investigation and analytic constraints need to be developed.

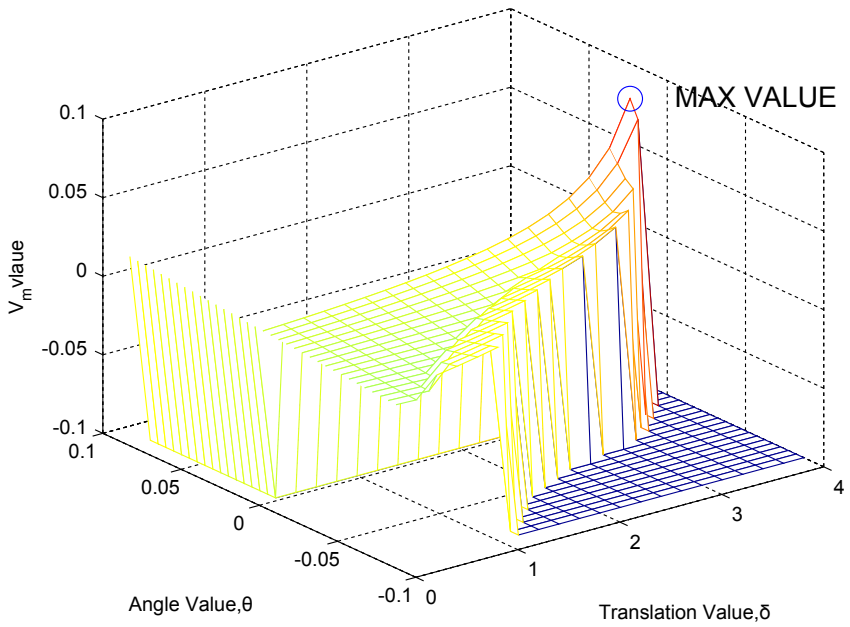


Figure 6.12: Surface Plot for Triangular Peg Contact State E4-V2. The maximum value for V_m is not higher than that of the dominating (sampled) configuration.

6.2 Summary of Chapter 6

This chapter presented additional numerical investigation supporting the use of a velocity metric based optimization for admittance selection. It showed significant evidence demonstrating that the quality of configurations not evaluated in the optimization routine better than the extremals evaluated. For the cases investigated the maximum discrepancy from the ideal motion is obtained at the dominating extremal.

The following section will highlight the contribution of the project as well as suggesting future work to be undertaken to improve upon the procedure for identifying compliant behavior resulting in close to optimal motion.

7 CONTRIBUTION AND FUTURE WORK

There are currently significant difficulties in the implementation of automatic assembly systems due to the lack of precise relative positioning in conventional robotic systems. Suggested solutions are either expensive and complex (active compliance and vision systems) or are only limited to certain geometries (RCC). Previous work done at Marquette University generated a process by which passive compliant systems could be designed yet the quality of the motions obtained was not optimal.

The contribution of this project was generating a process by which passive compliant systems can be designed for a variety of geometries which result in close to optimal motion. This allows for compliant systems to be designed which are optimal for a specified geometry, friction coefficient and target translational velocity magnitude. This presents an improvement over other admittance design processes which do not take into account the quality of the motion of the resulting corrective motion. This will allow for the design of relatively inexpensive passive compliant mechanisms for use in an industrial setting. Furthermore the tool created for the selection of the admittance behavior provides a solid foundation for further investigation into the qualities of ideal admittance matrices. The program handling the admittance selection has also been simplified enough that it could be used by individuals that are not necessarily experts in force-guided assembly. The project also showed that the use of a velocity based metric is useful for motion comparison tasks.

Further work on this area should be directed towards identifying the cause for the variation in the results for the stake shaped peg assembly. It is believed that the cause for these variations is the increase in the number of constraints for the optimization cause by an increase in the number of contact states identified. It is also believed that the manner by which contact states are decomposed for extremal identifications needs to be modified. Investigation into the cause for this discrepancy can provide further insight for more complex parts. More complex geometries should also be investigated to test the robustness of the process. Development of a process for designing compliant behavior for concave and three dimensional assemblies still remains to be done. Finally,

the development of conditions to limit the magnitude of the contact forces for all possible configurations is necessary to guarantee that the assembled parts will not be damaged during assembly the process.

BIBLIOGRAPHY

[1] Sigitas Kilikevičius Bakšys, Bronius and Andrejus Chadarovičius. Experimental investigation of vibratory assembly with passive compliance. *Mechanika*, 17(6):608–614, 2011.

[2] C. Y. Lin Chang, R. J. and P. S. Lin. Visual-based automation of peg-in-hole microassembly process. *Journal of Manufacturing Science and Engineering*, 133, 2011.

[3] Mani Maran Ratnam Chin, Kong Suh and Rajeswari Mandava. Force-guided robot in automated assembly of mobile phone. *Assembly Automation*, 23(1):75–86, 2003.

[4] Namik Cibilak and Lipkin Harvey. Remote center of compliance reconsidered. *Proceedings of the 1996 ASME Design Engineering Technical Conference and Computers in Engineering*, 1996.

[5] Luis Criales. Development of a velocity metric for rigid-body planar motion. Master's thesis, Marquette University, 2009.

[6] Mark Cutosky and Paul Wright. Position sensing wrists for industrial manipulators. *12th International Symposium on Industrial Robots*, 1982.

[7] J. De Schutter and H. Van Brussel. Compliant robot motion ii. a control approach based on external control loops. *The International Jounal of Robotics Research*, 7(4).

[8] Ernest D. Fasse Goeree, Barry B. and Michael M. Marefat. Determining feasible contact states of pairs of spatial polyhedra. *Proceedings of the 2000 IEEE International conference on Robotics and Automation*, 2:1396–1401, 2000.

[9] K. Maycock Haskiya, W. and J. A. G. Knight. A passive compliant wrist for chamferless peg-in-hole assembly operation from vertical and horizontal directions. *Proceedings of the Insitution of Mechanical Engineers*, 212(6):473–478, 1998.

[10] S. Huang and J. Schimmels. Admittance selection for force-guided assembly of polygonal parts despite friction. *IEEE Transactions on Robotics*, 40.

[11] S. Majumder Jain, R. K. and A. Dutta. Scara based peg-in-hole assembly using compliant ipmc micro gripper. *Robotics and Autonomous Systems*, 61:297–311, 2013.

[12] H. Kazerooni. Direct-drive active compliant end effector (active rcc). *IEEE Transactions on Robotics and Automation*, 4(3).

[13] Sangchul Won Lee, Sangcheol and Seungjoon Choi. Development of a new variable remote center compliance for assembly robots. *Advanced Robotics*, 14(3):241–255, 2000.

[14] Jing Xiao Joris De Schutter Herman Bruyninckx Meeussen, Wim and Ernesto Staffetti. Automatic verification of contact states taking into account manipulator constraints. *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, 4:3583–3588, 2004.

[15] Richard Murray, Zexiang Li, and Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[16] Chong Jin Ong and Elmer G. Gilbert. Growth distances: New measures for object separation and penetration. *IEEE Transactions on Robotics and Automation*, 12(6):888–903, 1996.

[17] Feng Pan and Joseph Schimmels. Robust procedures for obtaining assembly contact state extremal configurations. *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, 2004*, 1, 2004.

[18] Feng Pan and Joseph M. Schimmels. Efficient contact state graph generation for assembly applications. *Proceedings of the 2004 IEEE International conference on Robotics and Automation*, 2:2592–2598, 2003.

[19] Michael A Peshkin. Programmed compliance for error-corrective manipulation. *IEEE Transactions on Robotics and Automation*, 6(4):473–482, 1990.

[20] Bharath Ram Shetty and Marcelo H Ang. Active compliance control of a puma 560 robot. *Proceedings of the 1996 IEEE International Conference on robotics and Automation, 1996*, 4:3720–3725, 1996.

[21] Colin R. Reeves and Jonathan E. Rowe. *Genetic alorithms-principles and perspectives: a guide to GA theory.* Springer, 2002.

[22] R.H. Sturges and Schitt Laowattana. Passive assembly of non-axisymmetric rigid parts. *Proceedings of the 1994 IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, 2:1218–1255, 1994.

[23] Hong Qiao Zhicai Ou Su, Jianhua and Yuren Zhang. Sensor-less insertion strategy for an eccentric peg in a hole of the crankshaft and bearing assembly. *Assembly Automation*, 32(1):86–99, 2012.

[24] Daniel E Whitney. Quasi static assembly of compliantly supported rigid parts. *ASME Journal of Dynamic Systems, Measurements, and Control*, 104(1):65–77, 1982.

[25] Steven C. Wiemer and Joseph M. Schimmels. Optimal admittance characteristics for planar force-assembly of convex polygonal parts. *2012 IEEE International Conference on Robotics and Automation (ICRA).*

[26] R.F. Wolffenbuttel, K.M. Mahmoud, and P.P.L Regtien. Multiaxis compliant capacitice wrist sensor for use in automated assembly with industrial robots. *Instrumentation and Measurement Technology Conference, 1990*, pages 54–59, 1990.

[27] J. Xiao. Contact constraint analysis and determination of geometrically valid contact formations from possible contact primitives. *IEEE Transactions on Robotics and Automation*, 13(3):456–466, 1997.

[28] Yangsheng Xu and Richard Paul. A robot compliant wrist system for automated assembly. *IEEE International Conference on Robotics and Automation*, pages 1750–1755, 1992.

[29] Bum-Jae You, Young Seok Oh, and Zeungnam Bien. A vision system for an automatic assembly machine of electronic components. *IEEE Transactions on Industrial Electronics*, 37(5).

A OPTIMIZATION RESULTS FOR RECTANGULAR PEG ASSEMBLY

This section presents the obtained data for the selection of an admittance providing passive force-guidance with optimal motion for rectangular peg assemblies.

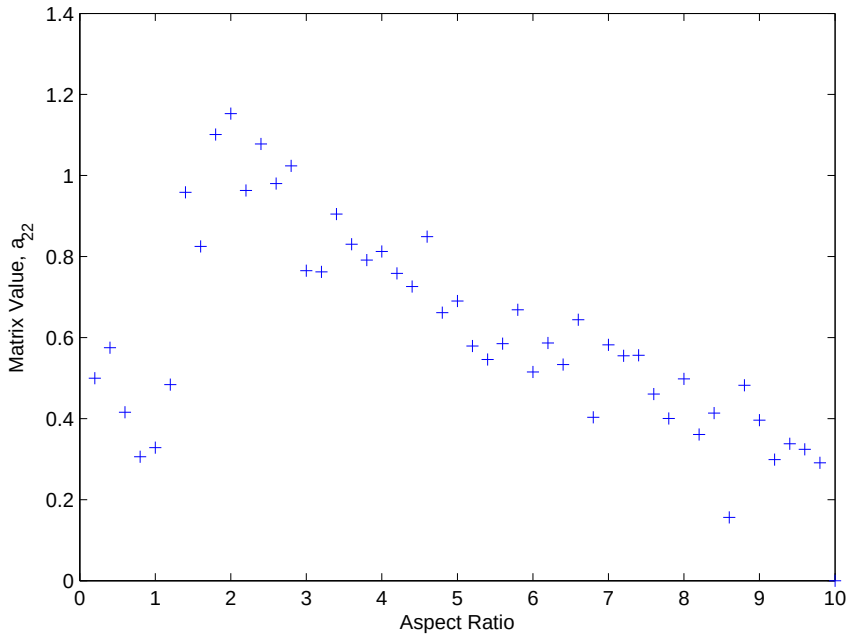


Figure A.1: Resulting a_{22} Values for Varying Aspect Ratio and $\mu = 0.3$ Rectangular Peg. No pattern is apparent.

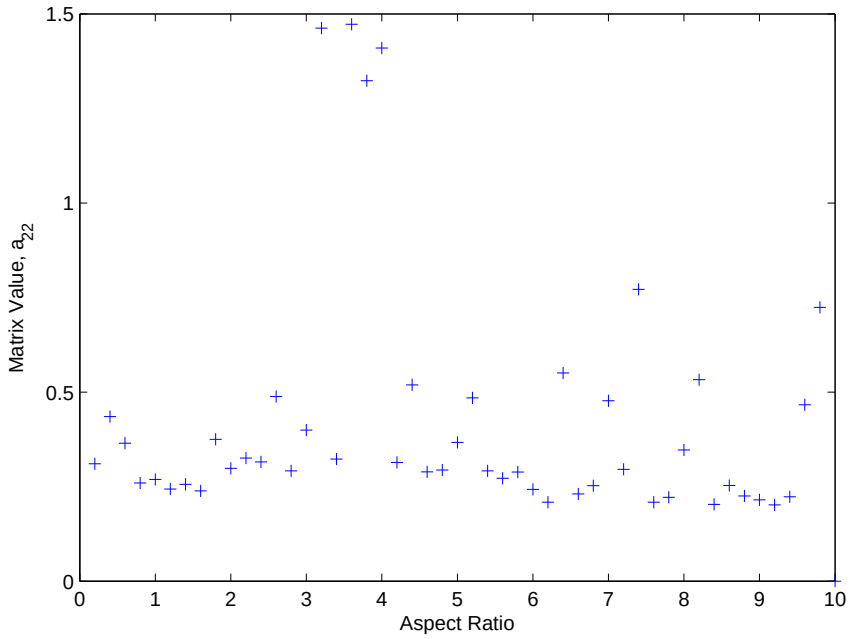


Figure A.2: Resulting a_{22} Values for Varying Aspect Ratio and $\mu = 0.5$ for Rectangular Peg. No pattern is apparent.

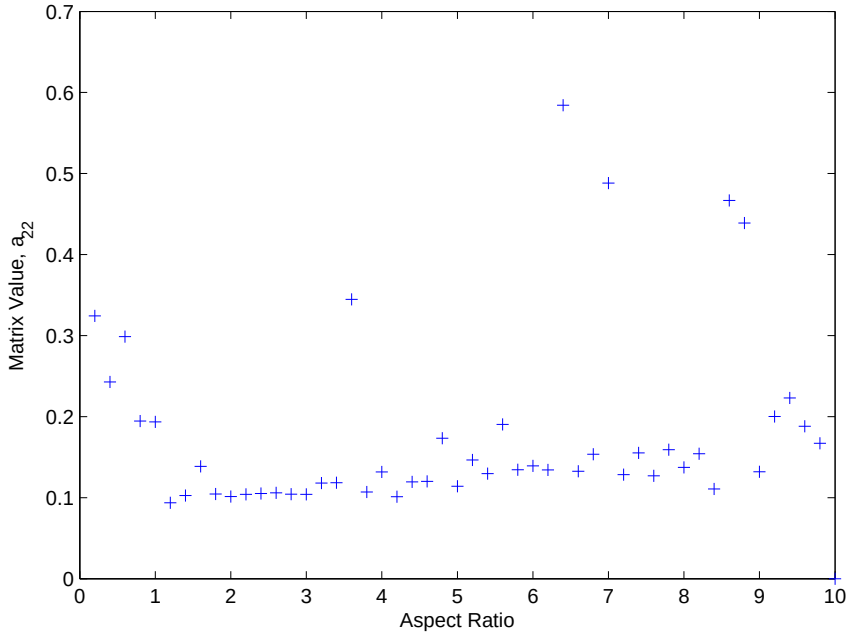


Figure A.3: Resulting a_{22} Values for Varying Aspect Ratio and $\mu = 0.7$ for Rectangular Peg. No pattern in apparent.

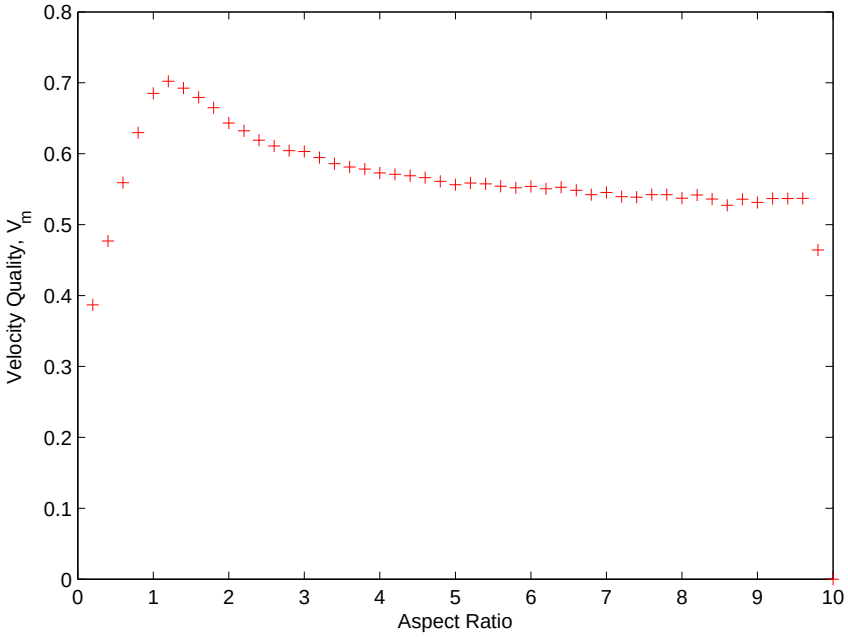


Figure A.4: Resulting V_m Values for Varying Aspect Ratio and $\mu = 0.3$ for Rectangular Peg. Values tend towards 0.55 for long peg assemblies.

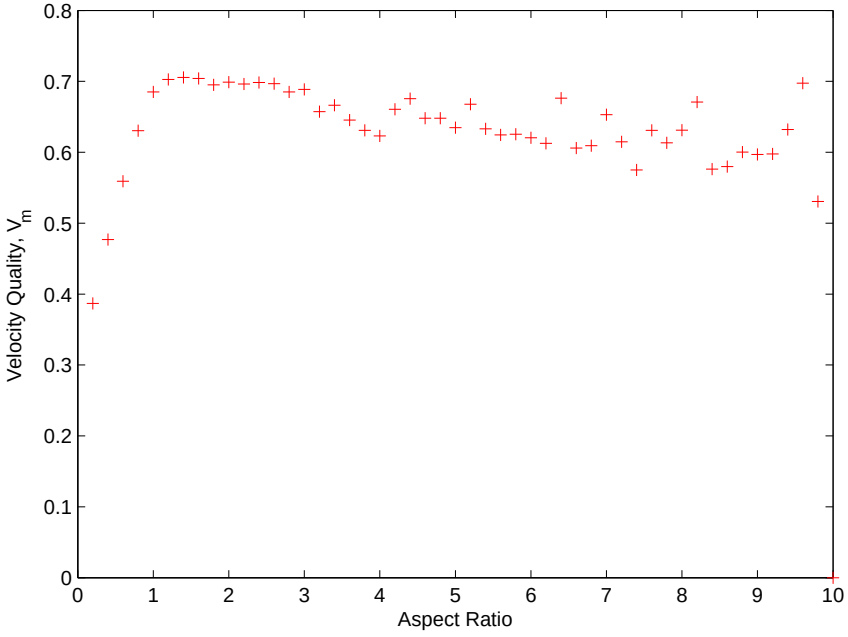


Figure A.5: Resulting V_m Values for Varying Aspect Ratio and $\mu = 0.5$ for Rectangular Peg. Values tend towards 0.65 for long peg assemblies.

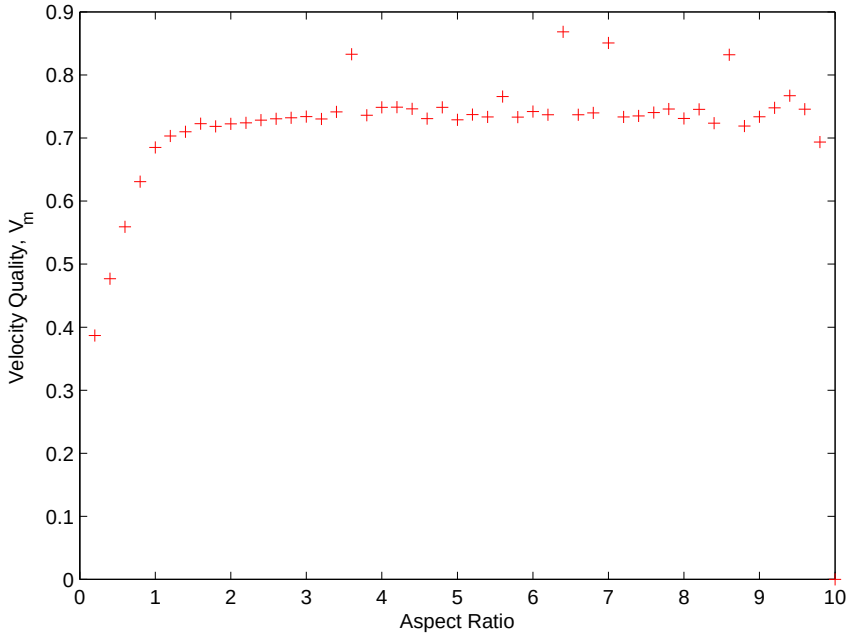


Figure A.6: Resulting V_m Values for Varying Aspect Ratio and $\mu = 0.7$ for Rectangular Peg. Values tend towards 0.75 for long peg assemblies.

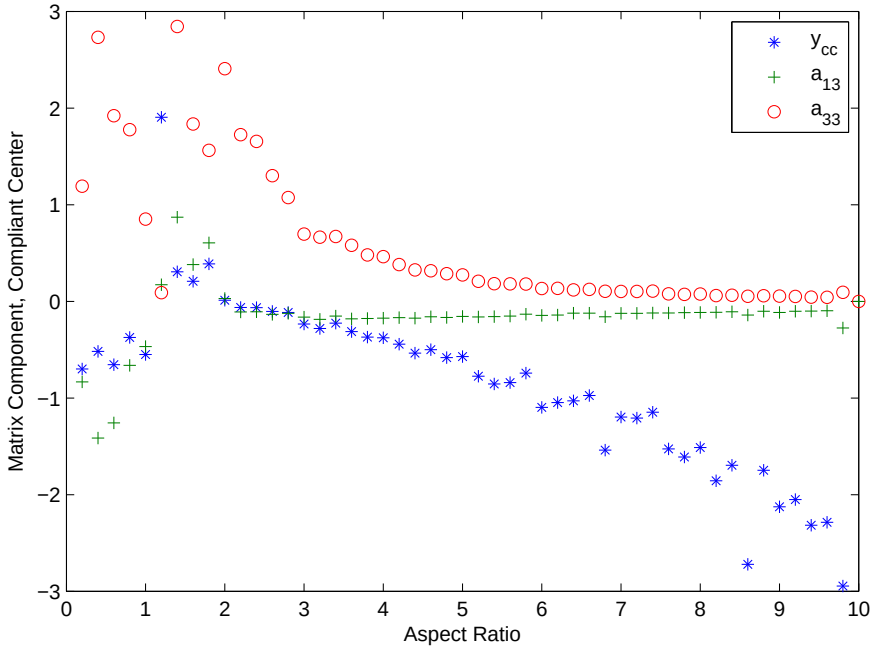


Figure A.7: Resulting y_{cc} Location for Varying Aspect Ratio and $\mu = 0.3$ for Rectangular Peg. Downward trend is observed.

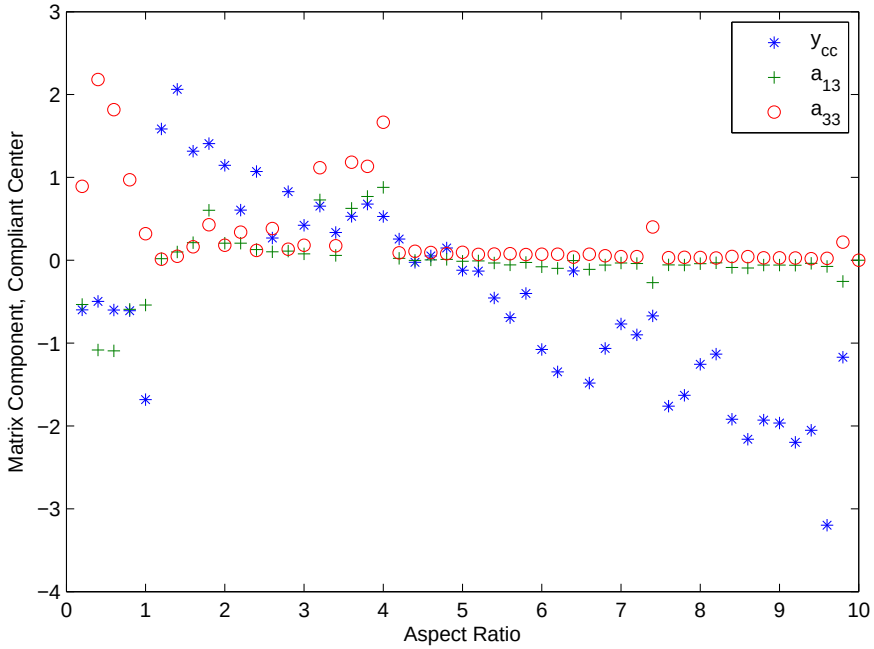


Figure A.8: Resulting y_{cc} Location Values for Varying Aspect Ratio and $\mu = 0.5$ for Rectangular Peg. Downward trend is observed.

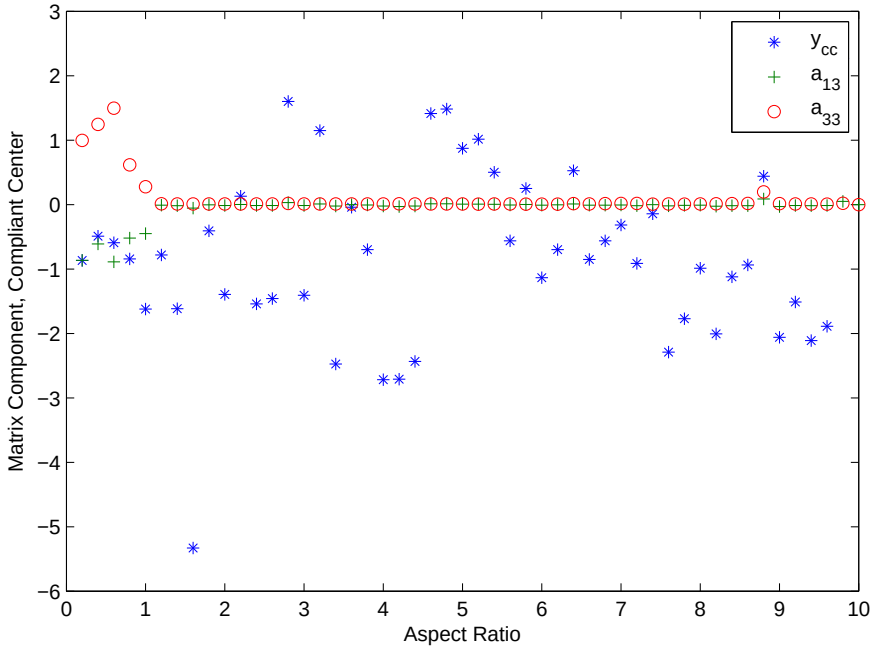


Figure A.9: Resulting y_{cc} Location for Varying Aspect Ratio and $\mu = 0.7$ for Rectangular Peg. Downward trend is not observed.

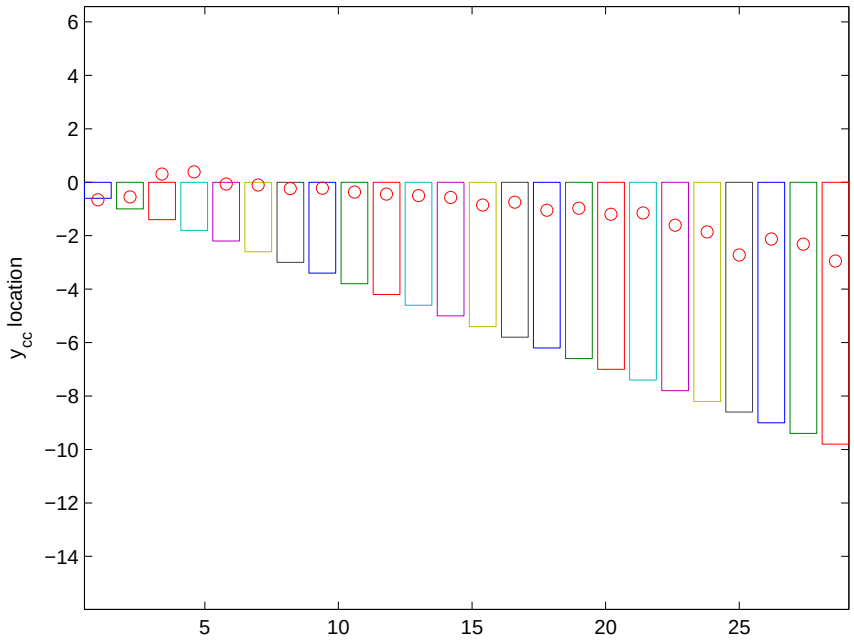


Figure A.10: Resulting y_{cc} Location in Relation to Peg for $\mu = 0.3$ for Rectangular Peg.

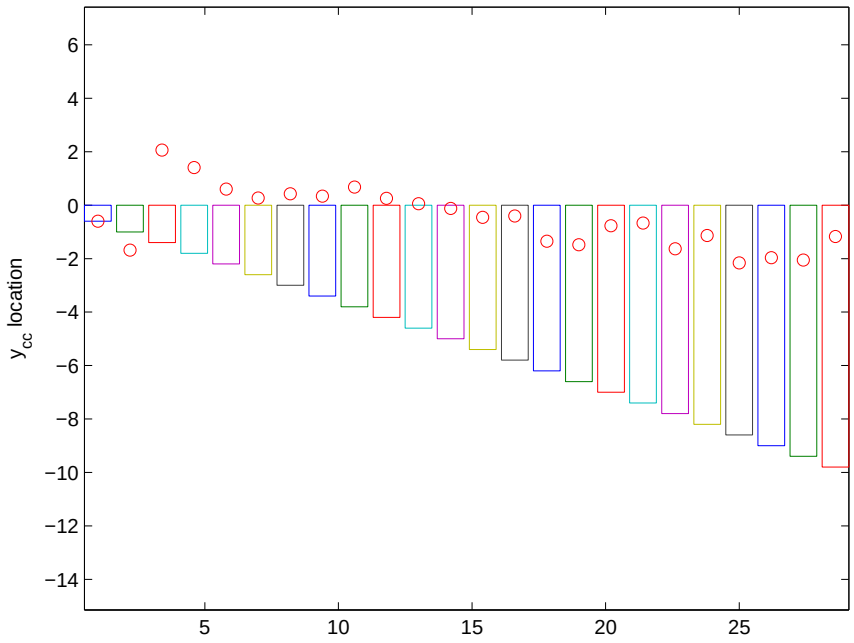


Figure A.11: Resulting y_{cc} Location in Relation to Peg for $\mu = 0.5$ for Rectangular Peg.

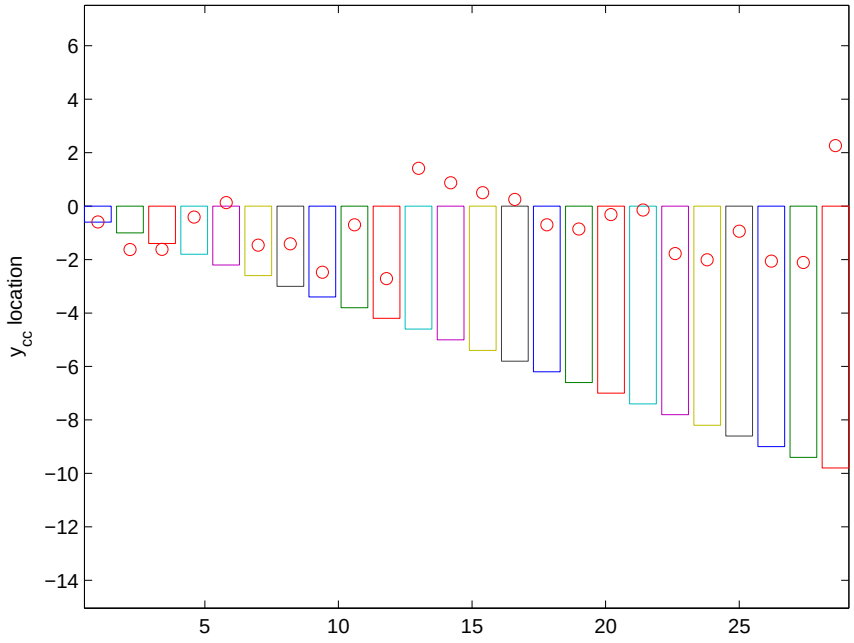


Figure A.12: Resulting y_{cc} Location in Relation to Peg for $\mu = 0.7$ for Rectangular Peg.

B OPTIMIZATION RESULTS FOR TRIANGULAR PEG ASSEMBLY

This section presents the obtained data for the selection of an admittance providing passive force-guidance with optimal motion for triangular peg assemblies.

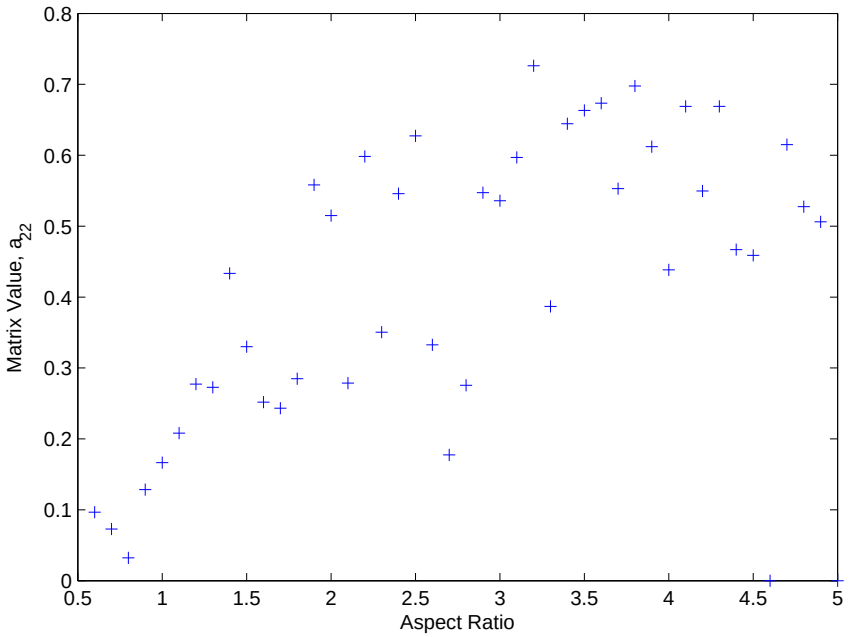


Figure B.1: Resulting a_{22} Values for Varying Aspect Ratio and $\mu = 0.3$ for Triangular Peg No pattern is apparent.

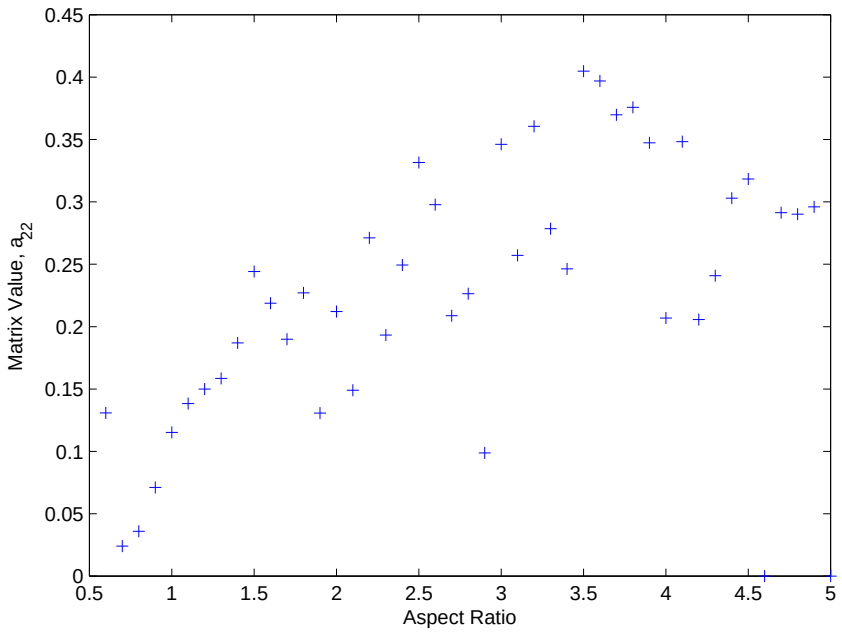


Figure B.2: Resulting a_{22} Values for Varying Aspect Ratio and $\mu = 0.5$ for Triangular Peg No pattern is apparent.

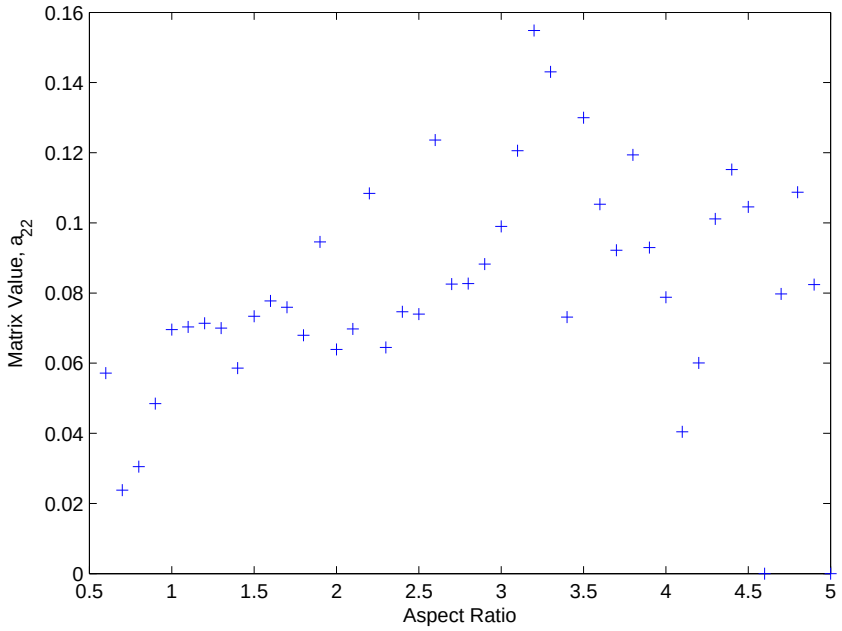


Figure B.3: Resulting a_{22} Values for Varying Aspect Ratio and $\mu = 0.7$ for Triangular Peg No pattern in apparent.

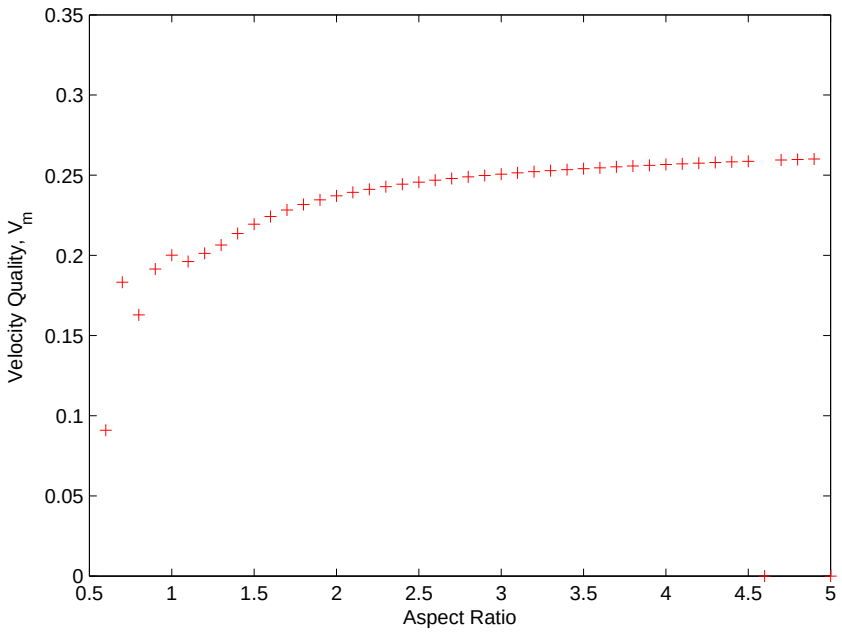


Figure B.4: Resulting V_m Values for Varying Aspect Ratio and $\mu = 0.3$ for Triangular Peg Values tend towards 0.55 for long peg assemblies.

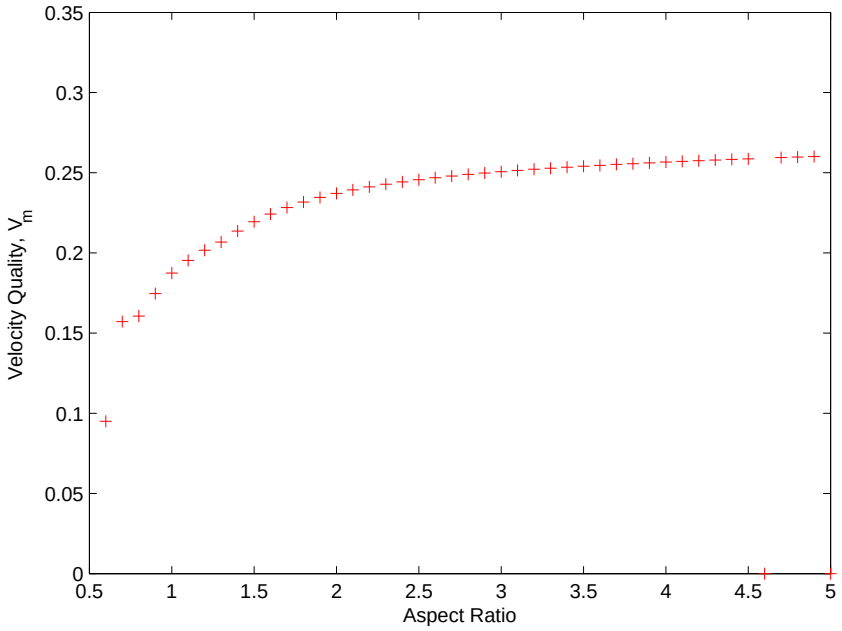


Figure B.5: Resulting V_m Values for Varying Aspect Ratio and $\mu = 0.5$ for Triangular Peg Values tend towards 0.65 for long peg assemblies.

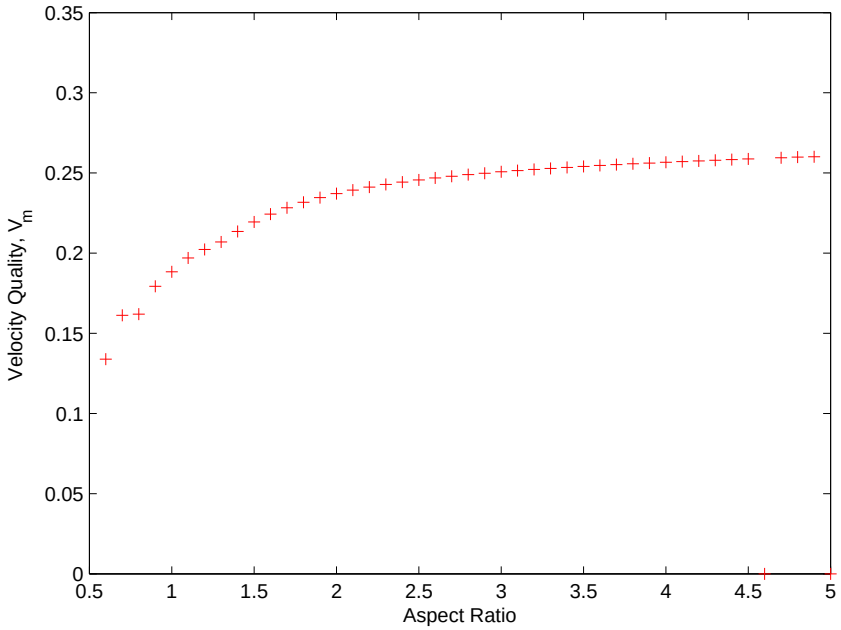


Figure B.6: Resulting V_m Values for Varying Aspect Ratio and $\mu = 0.7$ for Triangular Peg Values tend towards 0.75 for long peg assemblies.

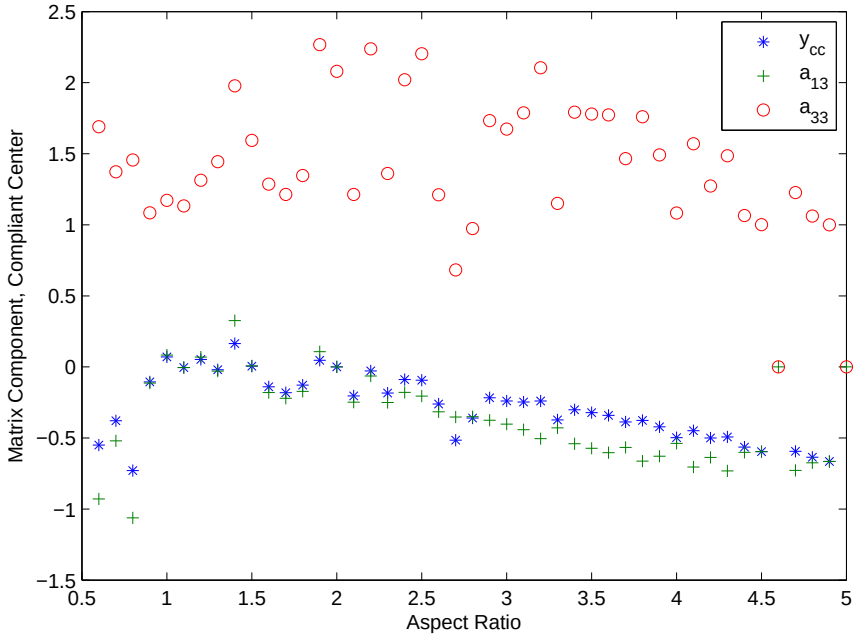


Figure B.7: Resulting y_{cc} Location for Varying Aspect Ratio and $\mu = 0.3$ for Triangular Peg Downward trend is observed.

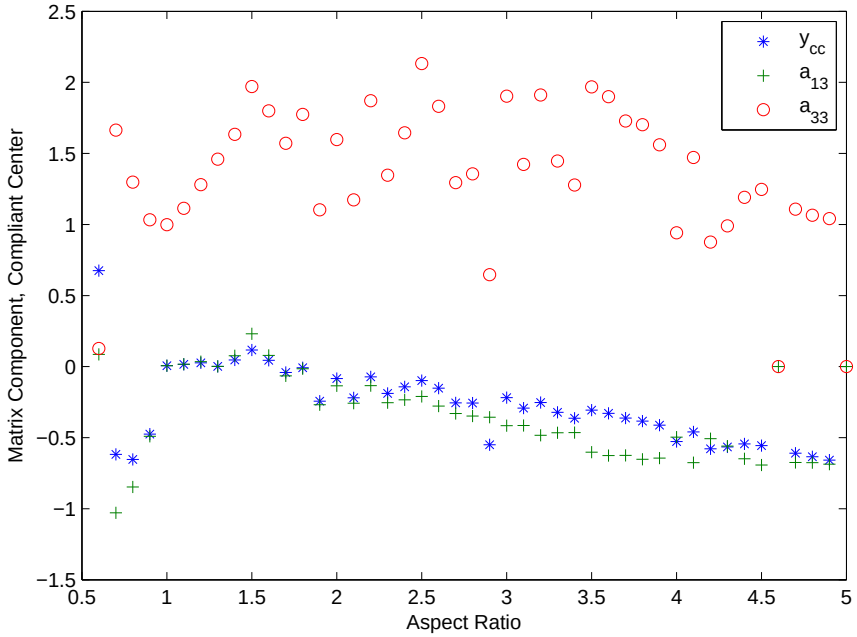


Figure B.8: Resulting y_{cc} Location Values for Varying Aspect Ratio and $\mu = 0.5$ for Triangular Peg Downward trend is observed.

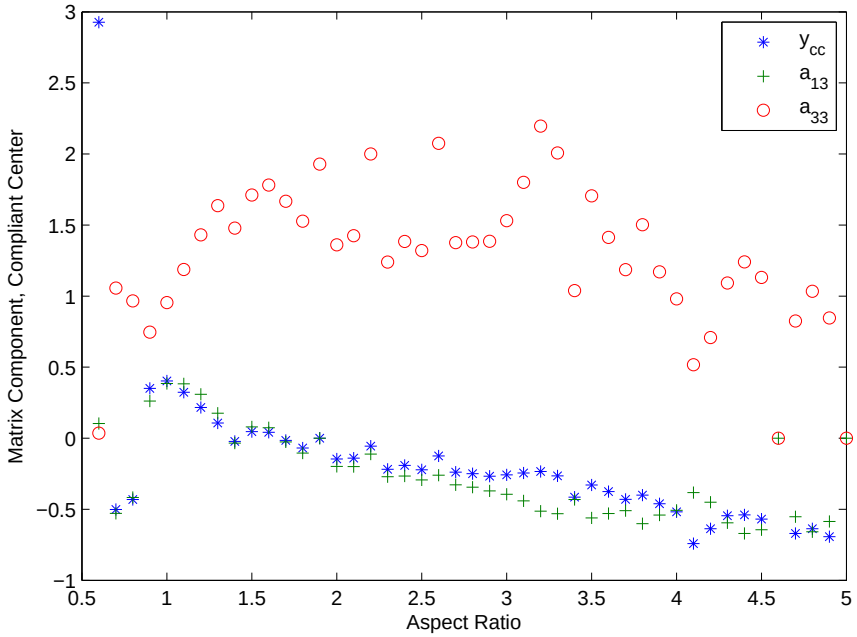


Figure B.9: Resulting y_{cc} Location for Varying Aspect Ratio and $\mu = 0.7$ for Triangular Peg Downward trend is not observed.

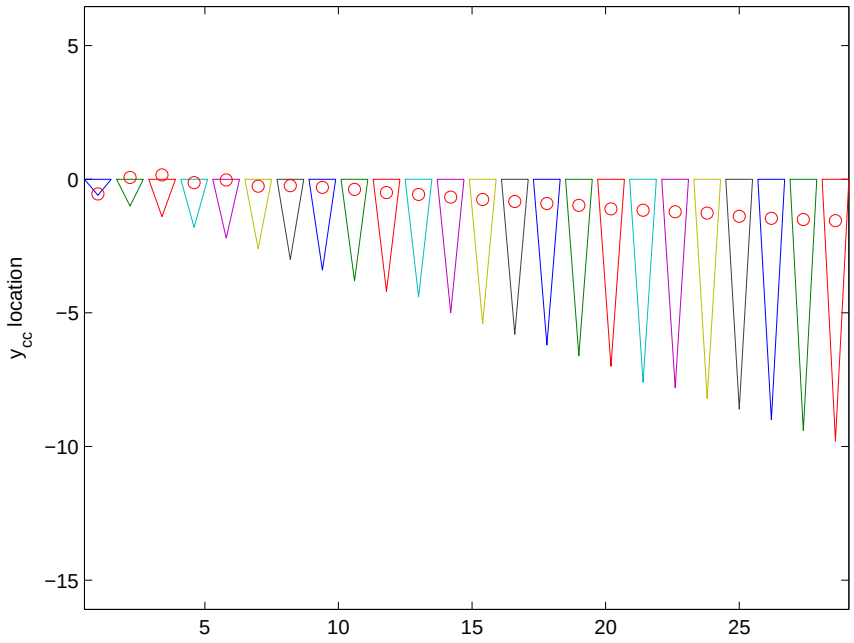


Figure B.10: Resulting y_{cc} Location in Relation to Peg for $\mu = 0.3$ for Triangular Peg

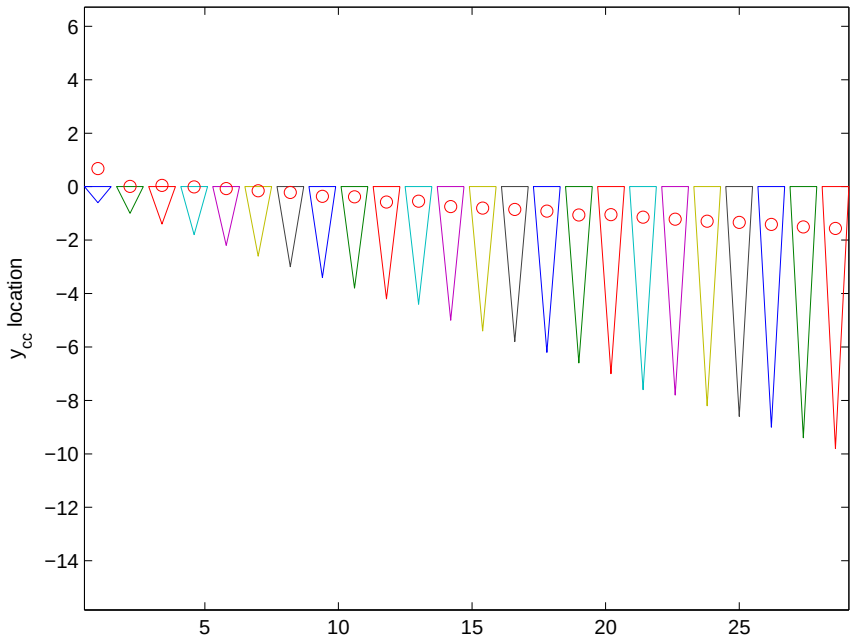


Figure B.11: Resulting y_{cc} Location in Relation to Peg for $\mu = 0.5$ for Triangular Peg

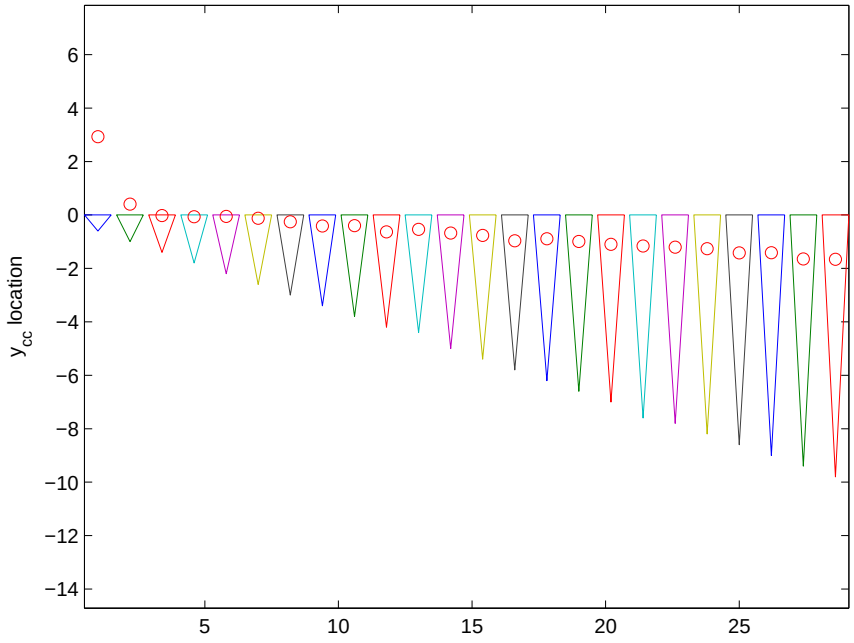


Figure B.12: Resulting y_{cc} Location in Relation to Peg for $\mu = 0.7$ for Triangular Peg

C OPTIMIZATION RESULTS FOR STAKE PEG ASSEMBLY

This section presents the obtained data for the selection of an admittance providing passive force-guidance with optimal motion for stake peg assemblies.

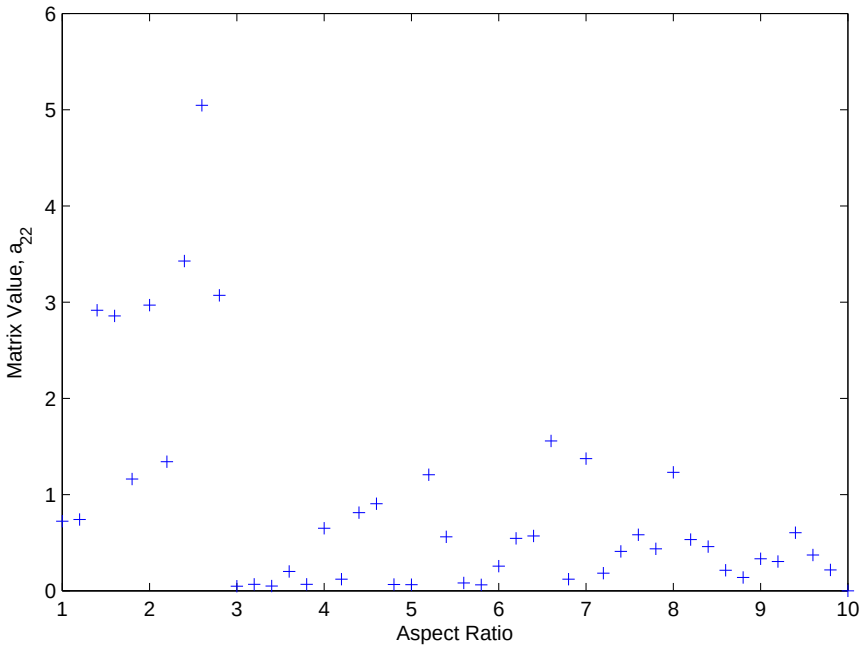


Figure C.1: Resulting a_{22} Values for Varying Aspect Ratio and $\mu = 0.3$ for Stake Peg. No pattern is apparent.

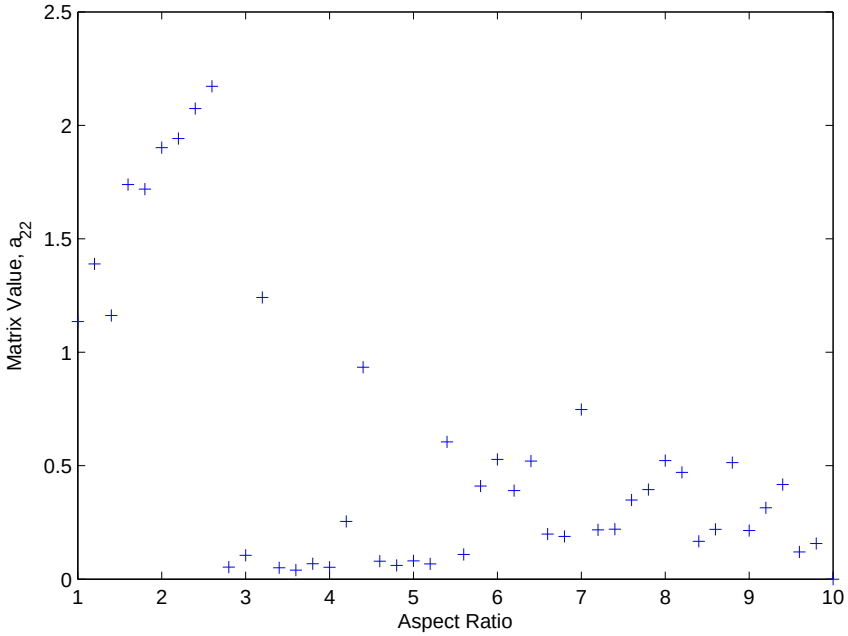


Figure C.2: Resulting a_{22} Values for Varying Aspect Ratio and $\mu = 0.5$ for Stake Peg. No pattern is apparent.

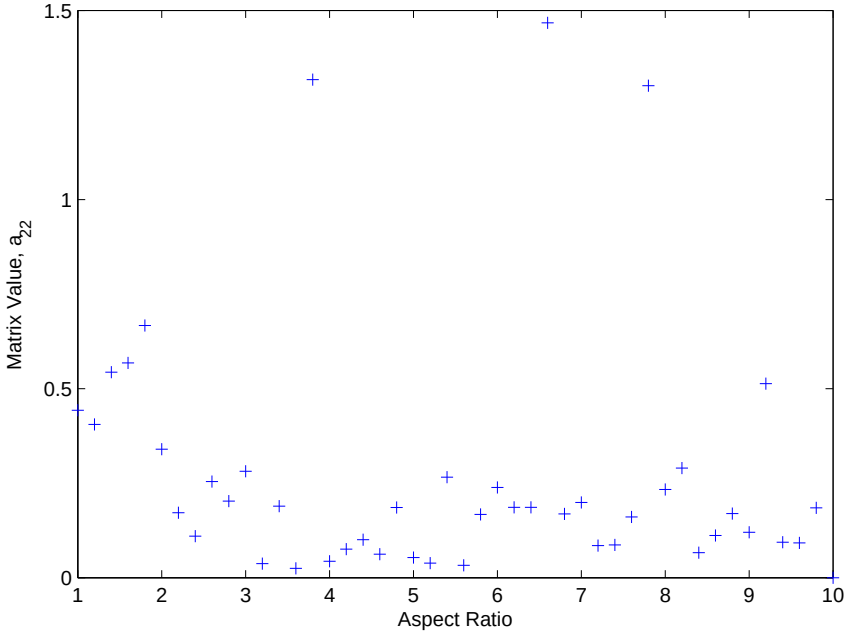


Figure C.3: Resulting a_{22} Values for Varying Aspect Ratio and $\mu = 0.7$ for Stake Peg. No pattern in apparent.

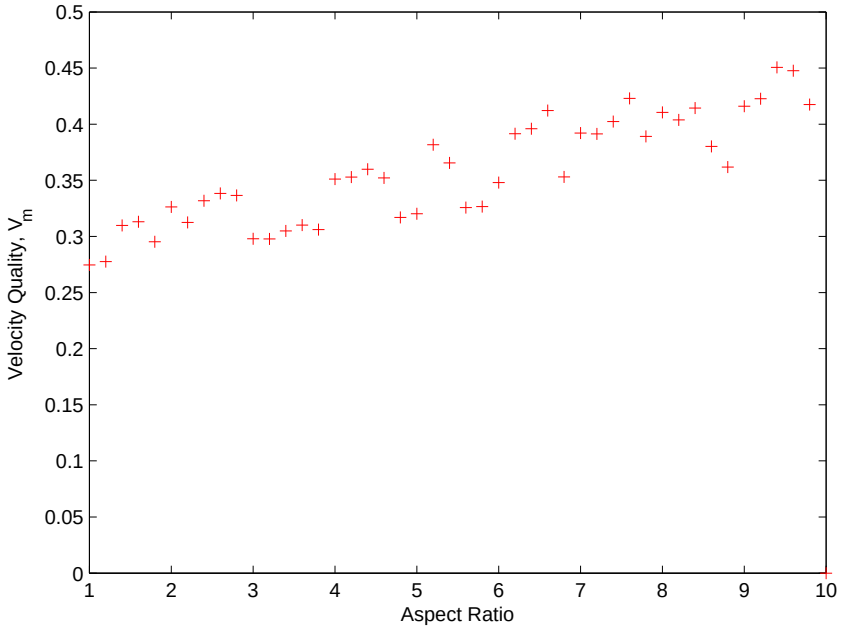


Figure C.4: Resulting V_m Values for Varying Aspect Ratio and $\mu = 0.3$ for Stake Peg.

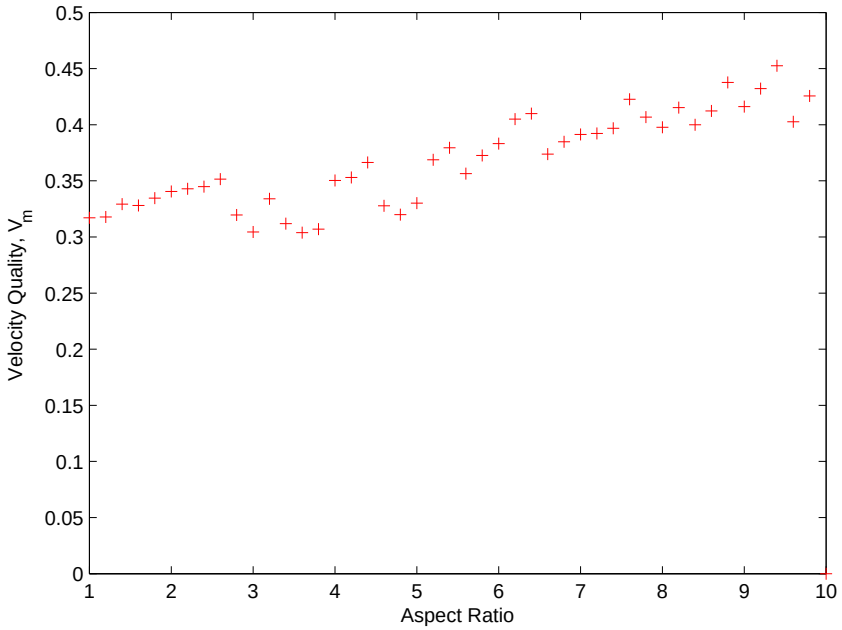


Figure C.5: Resulting V_m Values for Varying Aspect Ratio and $\mu = 0.5$ for Stake Peg.

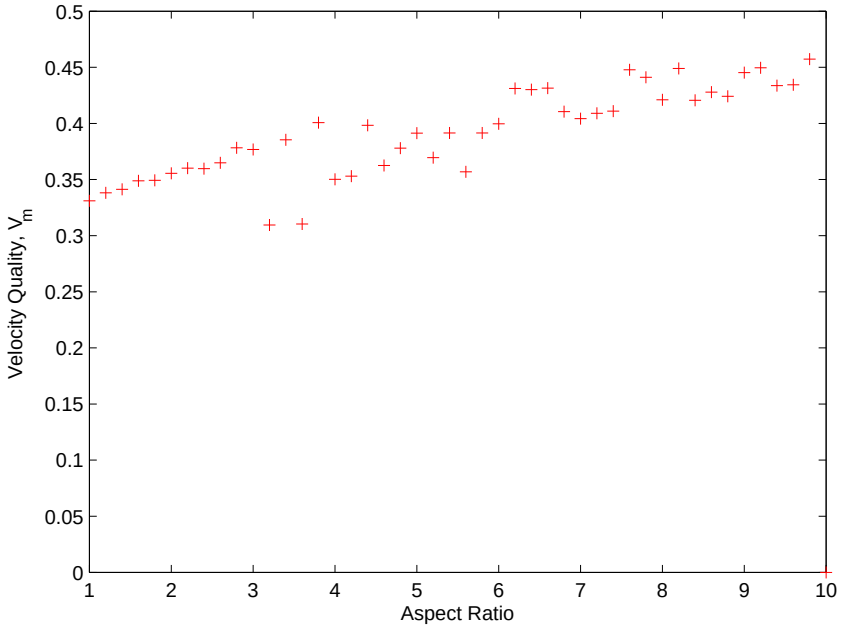


Figure C.6: Resulting V_m Values for Varying Aspect Ratio and $\mu = 0.7$ for Stake Peg.

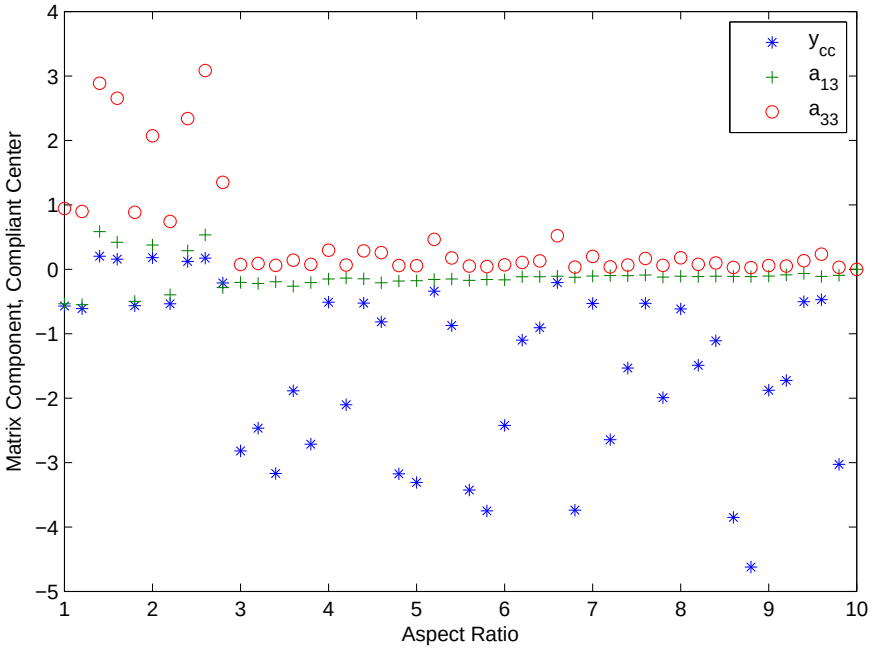


Figure C.7: Resulting y_{cc} Location for Varying Aspect Ratio and $\mu = 0.3$ for Stake Peg. Downward trend is not observed.

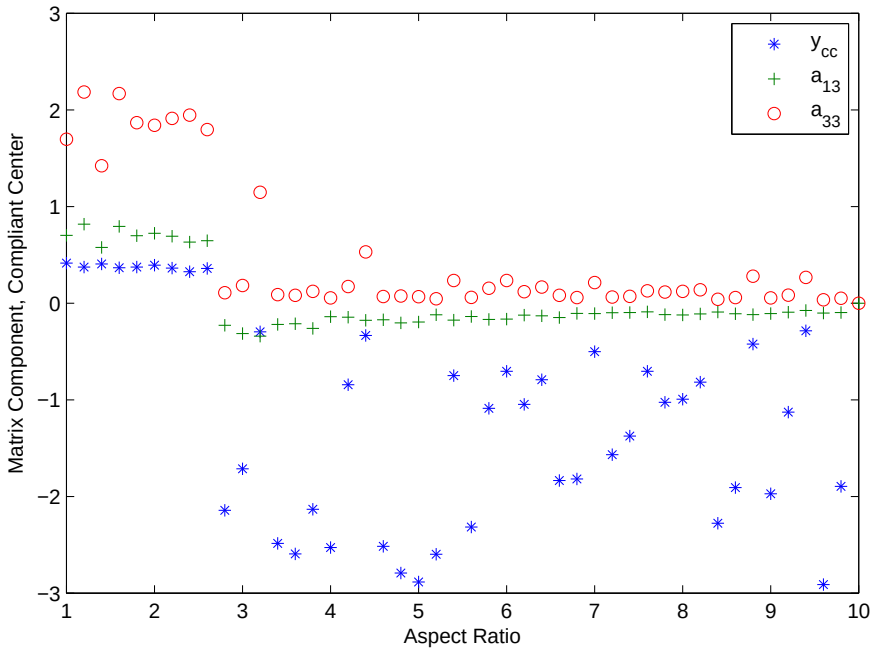


Figure C.8: Resulting y_{cc} Location Values for Varying Aspect Ratio and $\mu = 0.5$ for Stake Peg. Downward trend is not observed.

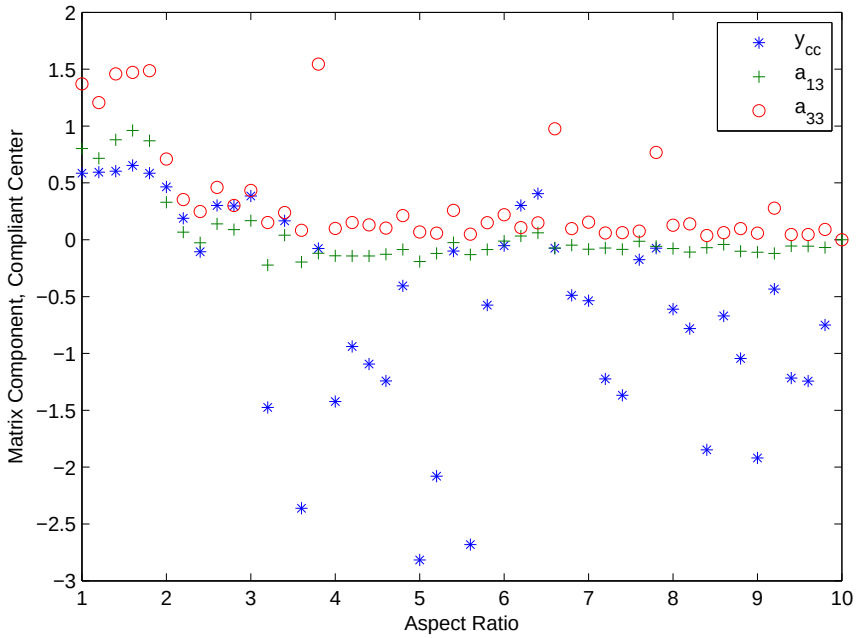


Figure C.9: Resulting y_{cc} Location for Varying Aspect Ratio and $\mu = 0.7$ for Stake Peg. Downward trend is not observed.

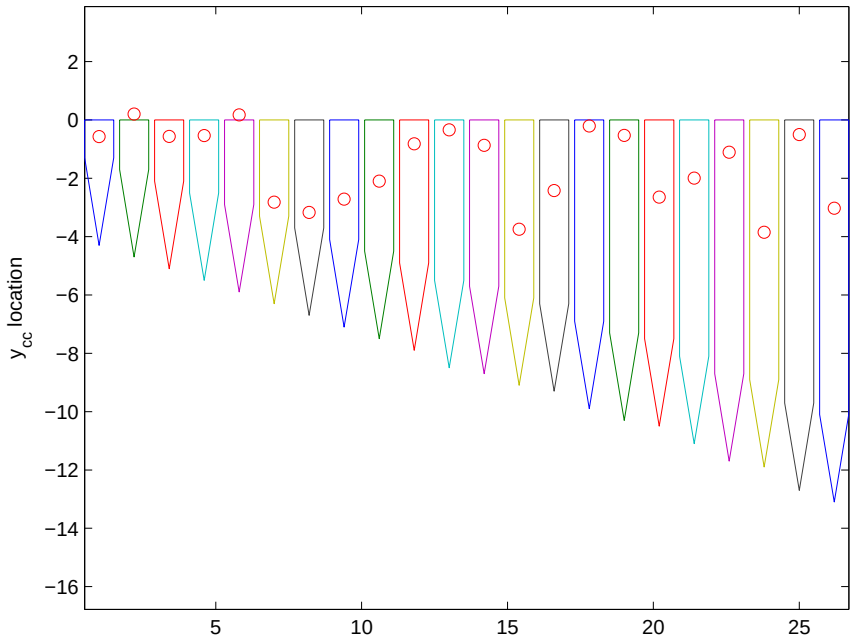


Figure C.10: Resulting y_{cc} Location in Relation to Peg for $\mu = 0.3$ for Stake Peg.

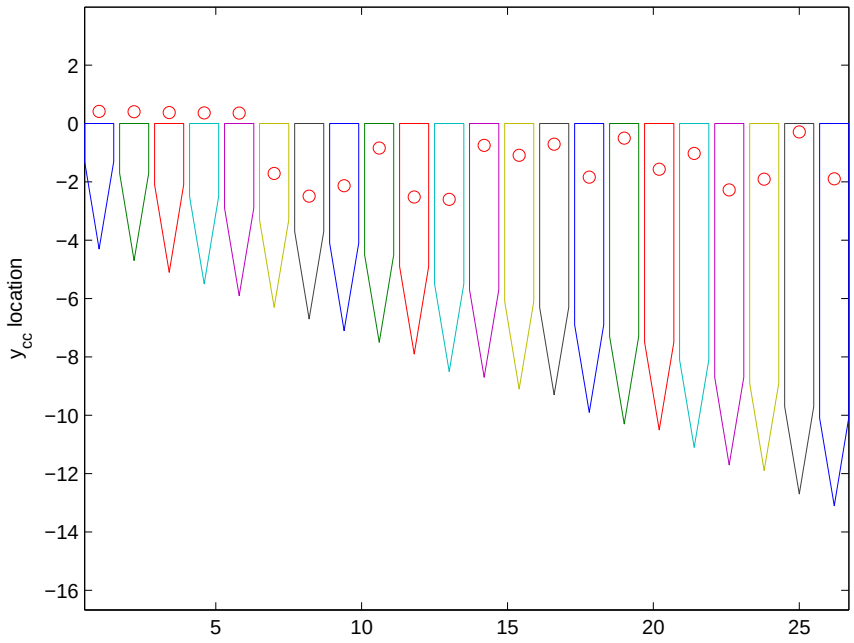


Figure C.11: Resulting y_{cc} Location in Relation to Peg for $\mu = 0.5$ for Stake Peg.

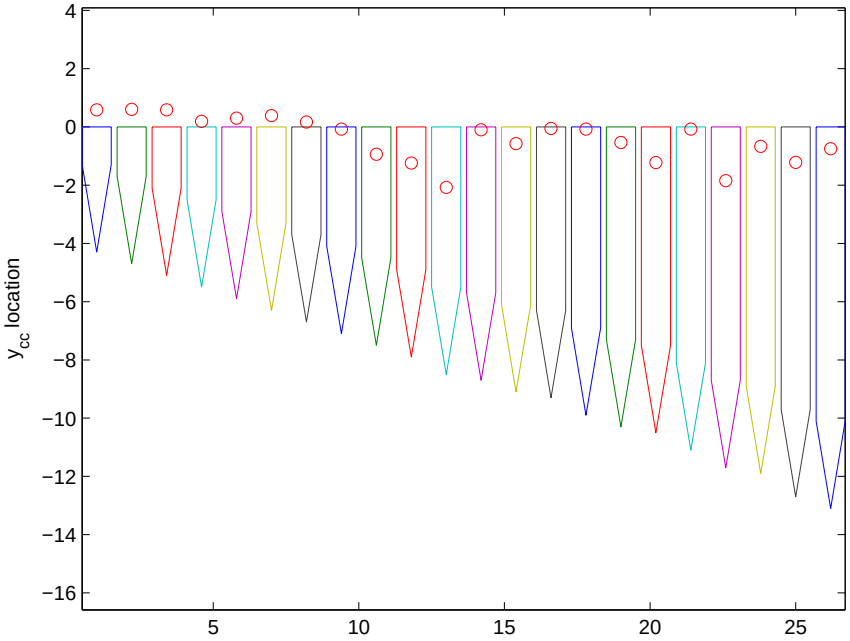


Figure C.12: Resulting y_{cc} Location in Relation to Peg for $\mu = 0.7$ for Stake Peg.

D MATLAB CODE VALIDATION

This chapter presents data supporting the results obtained by the new MATLAB program. The development of the MATLAB code to identify the best admittance matrix was based on work previously done by Weimer [25]. Wiemer developed a program that used a genetic algorithm to identify contact state extremals. This program also identified the best admittance guaranteeing successful assembly with the highest coefficient of friction.

Issues arose with said program. It was originally written in C++ and had become deprecated, not complying with modern C compilers. Furthermore it did not take advantage of modern multi-core microprocessors. Finally, the code being written in C++ did not posses high readability, especially when being used by mechanical engineers not used to large C++ files.

In order to fix these issues the program was rewritten in MATLAB, this resulted in a program is easier to understand and can take advantage of parallel processing while using MATALB's robust genetic and optimization routines. This also makes the program easier to build on and to understand.

Figure D.1 presents the process the program uses to obtain the highest admittance. The first part of the process requires the identification of the possible contact states that can occur within the given bounds of misalignment of the robot. The program first identifies one point contacts and then two point contacts using the possible simpler contact states. Second, the program identifies the extremals of the identified contact states. These extremals generate a rectangular space ($\pm\Delta\delta$ and $\pm\Delta\theta$) which contains all the possible configurations that occur within each contact states. Finally these extremals are used to generate a set of constraints that guarantee the assembled part will be successfully assembled and moving toward the appropriate position. These constraints are used for a constrained optimization whose objective function is the maximum friction coefficient.

The following figures present the results for rectangular, triangular and stake shaped pegs in force guided assembly tasks. This process was used to guarantee that the program

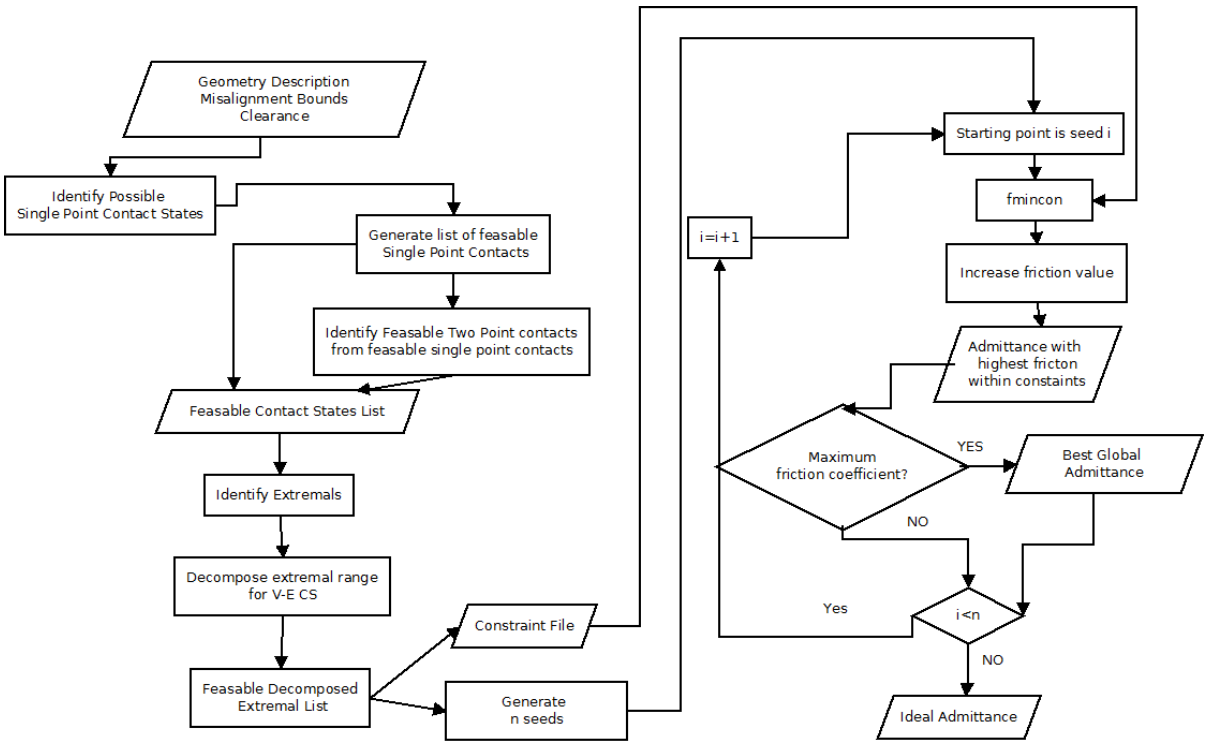


Figure D.1: MATLAB flowchart for Friction Based Optimization.

behaved in the same manner as its C++ equivalent (which no longer worked). The program was shown to be successful in obtaining the same results as those obtained by Wiemer. Given these results, the base of the MATLAB program was taken to be robust and used for the velocity based optimization.

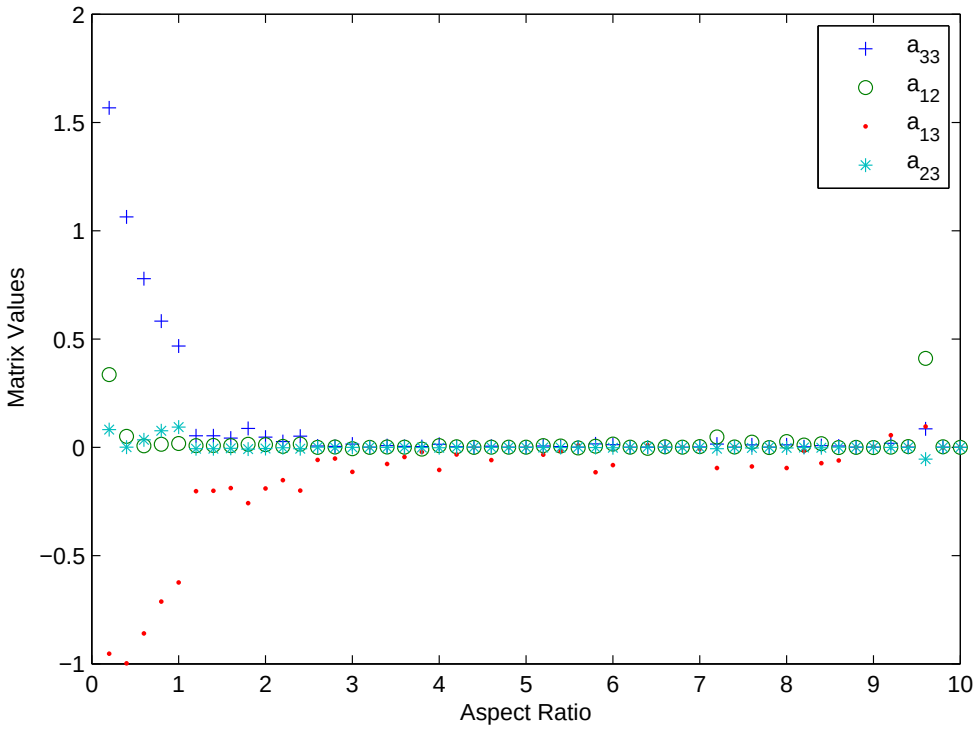


Figure D.2: Admittance Characteristics for Rectangular Peg using Friction Optimization.

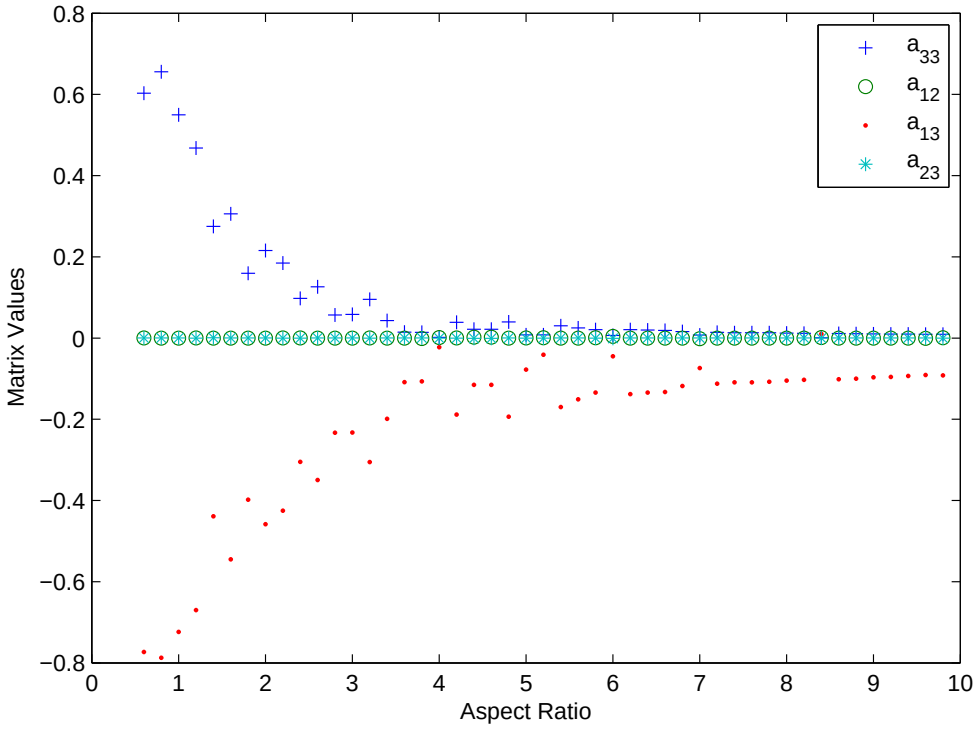


Figure D.3: Admittance Characteristics for Triangular Peg using Friction Optimization.

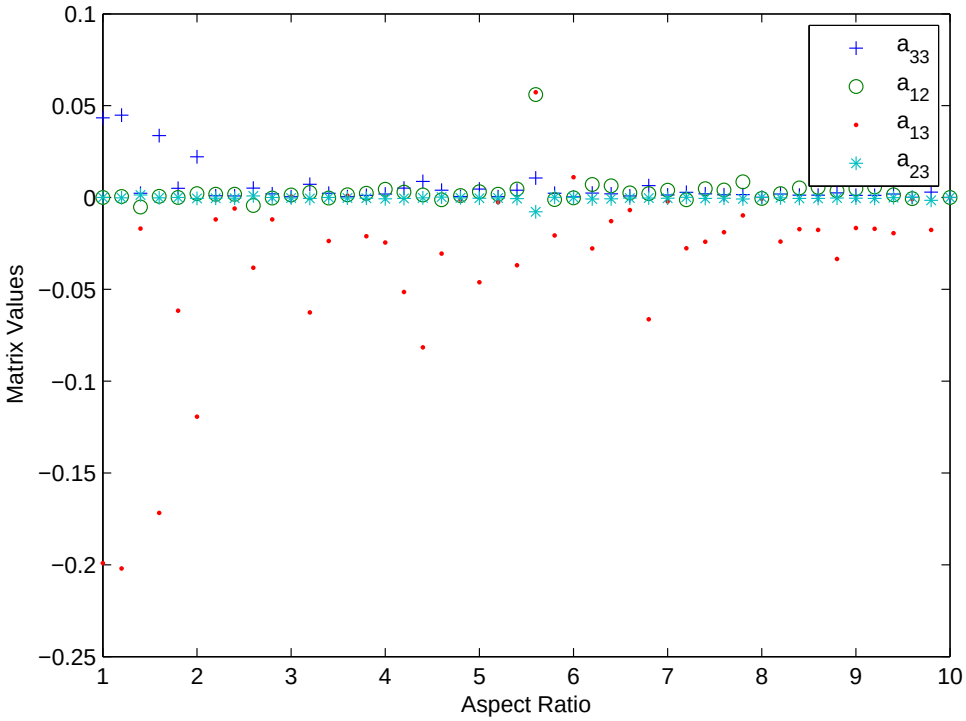


Figure D.4: Admittance Characteristics for Stake Peg using Friction Optimization.

E GENETIC ALGORITHMS

This chapter is meant to introduce the basic concepts of genetic algorithms as well as its implementation in the research project. First, a brief introduction to genetic algorithms (GA's) will be presented. Next, the basic concepts will be introduced. Finally, a discussion of the MATLAB implementation in the project is discussed.

E.1 Introduction

Genetic Algorithms have become in the last couple of years an increasing popular tool for solving complex optimization problems. This type of algorithm is based on evolutionary processes. The desired optimization is reached by simulating an evolutionary process on a starting population. This is accomplished by simulating a "real world" evolutionary process, complete with recombination, mutation and selection. The GA then allows certain individuals (solutions) to die or reproduce depending on their fitness (a value associated with the objective of the optimization) then allows these individuals to reproduce. GA's posses an advantage over other non-heuristic methods, they can cover a wider range of starting solutions, and they can also deal with non-linear problems with relative ease [21]. However there are some limitations to genetic algorithms. One of most important is that it does not have a clear end of the function, the result being the relatively better solution not the best solution.

E.2 Basic Concepts

As mentioned before most of the theory behind GAs is the attempt to replicate the evolutionary process of natural selection in the "real world". As a results, most of the fundamental principles mirror those found in evolution.

E.2.1 Individual

An individual is thought of as a group of parameters from which the objective function value is calculated. Each individual can be generated manually or in most cases, selected from an *inital population* which covers the range of the possible values for each variable. A diverse population allows for coverage of a relatively large search space. Once each individual is selected it is transformed into a chromosome. This *chromosome* is then converted into a *phenotype* from which genes are transformed to values for the diverse alleles, meaning gene types of the same family that result in different values. This function also serves to transform the values of the objective to only positive values, becoming what is known a fitness value.

This *fitness value* is a measure that is directly correlated to the objective function. It proves how "healthy" a given individual and determines its reproduction success rate, with the fitter individuals reproducing more successfully.

E.2.2 Crossover and Selection

As mentioned out of the initial population the fittest individuals are selected to reproduce at a higher rate. While there are several methods by which to accomplish this, they will not be discussed here. Sufficient to say that the healthier individuals reproduce the most.

A new generation is created using crossover meaning that the selected individuals have their chromosomes combined to create new individuals. This allows for the creation of individuals whose individual allele values contributed to the success towards the optimization. In most modern approaches, *elitisim* is used, a process by which the most succesful individuals from one generation is carried over to the new generation.

The process is repeated until a certain number of generations are achieved, the change in the best individuals fitness is less than a given value or the change in fitness becomes stalled.

E.2.3 Mutation

In order to maintain a diverse population and prevent the optimization from converging to local minima, a mutation is used. This is a relatively simple process, by which some of the individuals have the values of their alleles changed, resulting in new individuals which might perform better than the offspring of fit individuals.

Diverse methods exist for how to accomplish both selection, crossover and mutation, however the discussion of that theory is much more complex and hence not done here.

E.3 Implementation

When using GA's for the optimizations in the research project, MATLAB was chosen due to its widespread adoption and robust routines. This GA is part of the global optimization toolbox for which a license is required. Other alternatives include MIT C++ GAlib which was used in the previous program and University of Sheffield's Genetic Algorithms Toolbox which used in alternative versions of the code for which a global optimization toolbox is not available.

In order to set up GA in MATLAB first an objective function needs to be created (for our application the fitness function and the objective function are the same). The function prototype is as follows.

```
function (y)=ftn_fcn(x,a,b)
...
...
y= end value;
end
```

The optimization is then started by running the command

```
X=ga(@(x)ftn_fcn(x,a,b),nvars,A,b,Aeq,beq,LB,UB,nonlcon,options)
```

This function returns a vector `X`, with the optimized results. The other parameters are defined as follows:

- `nvars`, which defines the number of decision variables which create the chromosome. Meaning how many parameters contribute to the optimization.

- `A,b` define a set of linear constraints of the form $Ax < b$. Where A is a matrix, and b a vector. These where not used in most cases and can be entered as [].
- `Aeq, beq`, same as `A,b`, defining equality constraints. Substituted by [] in most situations.
- `LB,UB` define the upper and lower bounds of the variables used in the optimization.
- `nonlcon` defines the non linear constraints and is generally and exterior function in the case of this project these constraints are the ones that guarantee the succesful assembly of the parts.
- `options` MATLAB structure which defines the optimization parameters such as crossover, mutation, selection and initial population.

The options structure needs to be modified for the given application for the project the following structure was used.

```
options=gaoptimset('PopInitRange',[lb; ub],'CrossoverFraction',0.6,
'EliteCount',3,'Display','final','FitnessLimit',thresh,
'FitnessScalingFcn',@fitscalingrank,'Generations',300,
'PopulationSize',300,'StallGenLimit',50,'TolFun',1e-12,
'UseParallel','always');
```

Each of these parameters tailors the optimization to work efficiently for the given task of identifying extremals. It is important to note that most of these parameters require a certain amount of trial and error.

- `PopInitRange` this matrix defines the range of the initial population over which an even distribution of individuals will be generate. If this parameters is not defined MATLAB generates an initial population that fills a 1 by 1 area, not covering the entire space.
- `CrossoverFraction` the percentage of the new generation which is generated using simple crossover, for this project 0.7 of the new population is the result of crossover.

- `EliteCount`, defines teh number of individuals which are guaranteed to survive for the next generation. For this case the 3 healthiest individuals of each generation are saved.
- `Display` decides how much information is presented to the user.
- `FitnessLimit` defines what value determines the value at which if a an individual achieves halts the algorithm and is presented as the best result.
- `FitnessScalingFcn` defines the scaling routine that transforms the fitness to make sure that significant difference exists between individuals, for example the difference between 20 to 30 as opposed to 1020 to 1030. An exterior function is selected in this case a simple ranking of individuals is performed, other options can be found in MATLAB's documentation.
- `Generations` defines the number of generations after that which the optimization is halted.
- `PopulationSize` defines the number of individuals for each generation
- `StallGenLimit` sets the number of generations after which if no progress is made the optimization halts.
- `TolFun` is the value that defines the minimum improvement that has to be made by the fitness of each best individual with regards to the previous generation. If this is not accomplished the optimization is assumed to have reached a minimum.
- `UseParallel` enables parallelization for the optimization.

Once all these parameters the optimization is started, results for this project on a quad core computer generally take about 1 minute.

F PARALLEL AND DISTRIBUTED COMPUTATION

The code used to perform the optimization was altered from the previously used to function both in parallel and distributed computing. Both of these concepts are related, they are computing techniques that can be used to take advantage of new multiprocessors architecture. Parallelization requires the code to be written in such manner that it can be computed by separate computer cores. Distributed computing requires the code to deployed on a separate computing grid, similar to batch processing, making it ideal for parameter sweep processes.

F.1 Parallel Computing

Previously all code had to be computed serially, meaning that all lines are evaluated one after another. This means that a multicore processor could not be utilized to its full potential as it becomes bottle necked at certain point, only once core is being used, reducing the computational speed. The process of parallellization is to identify opportunities on which computations can be computed at the same time. A clear example that benefits from parallelization is that of parameter speeds in design optimizations. Where the output of one function call does not depend on it being executed one after another. The execution of this parallel program however is still undertaken by the memory of a single computer, not several ones. The memory is split between the cores as necessary

F.1.1 Implementation

In the computer program design the process of parallelization was relatively straight forward, taking advantage of the different functions that MATLAB has available to achieve a given result. The first step in achieving this is to declare the number of workers (cores) which matlab is available to use, this is declared as follows.

```
matlabpool open local 4
```

This command tells MATLAB to open up 4 local cores, and use them for the calculations. Most optimization routines have parameters that can be set to allow them to run in parallel. These can be achieved as follows

```
optimset('UseParallel','always')
```

Which orders the optimization routine to try to run the code in parallel for its function evaluations. This increased the speed of the program significantly. Due to the significant speed change going from a compiled language (C++) to an interpreted one (MATLAB), the increase in speed for the code in a 4 core computer was not as significant as expected. However this does allow for the speed of the code to significantly increase following trends in increase of cores per processor in computers.

F.2 Distributed Computing

Distributed computing is in similar to parallel computing. For the purposes of this project it is defined as the generation of a code whose concurrent execution is performed on multiple nodes (processors) in a network. As opposed to parallel software these nodes do not share a common pool of memory and can function independently. These types of distributed networks can range from a few nodes, to thousands on bigger national grids.

F.2.1 Implementation

The program created to identify a suitable admittance matrix for a variety of geometries was perfectly suitable to have an implementation on a distributed network. This is accomplished by using the PERE grid at Marquette University a 1024 core network. This allows for obtaining admittance matrices for a large number of geometries in a short period of time. This is accomplished by making use of University of Wisconsin Madison's CONDOR software which handles the submission.

It is important to note that CONDOR and most distributed networks cannot process MATLAB programs natively, as this requires a high number of licenses to be purchases. A workaround for this issue is to use MATLAB's C++ compiler. This allows the creation of a executable that does not require an existing MATLAB license and can be run on a variety of platforms.

The process of submitting the program to the CONDOR system requires the folowing process

1. Login into the PERE cluster, using ssh (secure shell)

2. Loading of the appropriate module, which defines the correct paths for the MATLAB programs

   ```
   module load matlab/2011a
   ```

   other modules for different MATLAB distributions can be found using

   ```
   module load avail
   ```

3. Creation of a makefile as follows (*makefile*)

   ```
   main_code: main_code.m
   mcc -m -R -nodisplay -R -nojvm main_code.m
   clean:
   rm -f main_code.m
   rm -f *_main.c *_data.c *.prj readme.txt *.log run_*.sh
   ```

4. Creation of the executable with the `make` command

5. Creation of sh (BASH script) to be run by CONDOR (*main_ code.sh*)

   ```
   #!/bin/bash
   source /etc/bashrc && source /etc/profile
   # Source the modules script
   #source /cluster/Modules/3.2.7/init/bash
   # More Error Checking
   chmod +x main_code
   export LD_LIBRARY_PATH=/group/hpc-share/MATLAB...
   /R2011a/bin/glnxa64:$LD_LIBRARY_PATH
   module load matlab/2011a
   ./main_code $1
   ```

This code makes sure that the program can be executed, using *chmod* for privilege handling and defines the global path to the MATLAB compiler as well as removing unnecessary files.

6. Submission of the file to CONDOR, this requires a submission file to be created (*main_ code.sub*)

```
Universe = vanilla
Executable = main_code.sh
Arguments = $(PROCESS)
Output = main_code_$(PROCESS).out
Error = main_code_$(PROCESS).err
Log = main_code.log
initialdir = Results_Run
Requirements = ( OpSys == "LINUX" && Arch == "X86_64" )
transfer_input_files = ../main_code, ../X_INIT_NORM.mat...
 ../normal_b.csv, ../normal_a.csv, ../vertices_b.csv,...
  ../vertices_a.csv, ../mxlpsolve.mexa64, ../liblpsolve55.so
should_transfer_files = TRUE
when_to_transfer_output = ON_EXIT
Queue 100
```

This defines arguments the program takes, in this case aspect ratio. The requirements on the computers that run it (LINUX with with X86_64 architecture). The number of subjobs to submit (Queue), which correspond to Aspect Ratio in tenths. Transfers the necessary files, and tells the code to transfer the output files to the Results_Run directory. This code is then submitted to condor using

```
condor_submit main_code.sub
```

The status of the submitted jobs can be check by using

```
condor_q
```

The resulting files have the following extensions

- *.err* these files contain CONDOR error reports, if code is succesful they will be empty
- *.log* this file contains all the information of the overall CONDOR job
- *.out* contain the MATLAB console output of each individual submission inside a job
- *.mat* contain the variables used in the MATLAB program

The use of this distributed programming approach has significantly reduced the amount of time required per geometry family (rectangle, triangle, stake) from weeks to approximately 6 to 20 hours depending on geometry, with stake taking the longest.

G MAIN_CODE.M

This chapter presents the main function to perform an optimization, it is important to note that this function was used for batch processing in the distributed network so some of parameters have to be hard coded into it. Mostly functions dealing with object decomposition and vertex transformation. The hard coded parameters can be generated by some of the provided values.

```
function main_code(num_run,fric)
%% Admittance Matrix Generator for Force Guided Assembly Despite Friction
% _Fernando Rodriguez Anton_
%
% _Marquette University_
%
% This code generates an admittance Matrix for force guided assembly such
% that it follows the control law
%
% $$ v=v_o + AW|phi $$
%
% Where $v$ represents a twist (motion) in a 2D space with components
% $v_x$, $v_y$, $|theta$, $A$ represents the admittance matrix $W$
% represents a contact wrench and $|phi$ represents the magnitude of the
% contact.
friction=str2double(fric);
num_r=str2double(num_run);
str=datestr(clock);
tr=(clock);
cts=2*((num_r)/10);
File_name=sprintf('ACSG_%d_%d_%d_%.1f_F_%.1f.txt', tr(1),tr(2),tr(3),cts,
    friction);
disp(File_name)
S_name=sprintf('Var_F_%.1f_AR_%.1f_%d_%d_%d.mat',friction,cts, tr(1),tr(2),
    tr(3));
disp(S_name)
%% Constant Definitions
```

```
% The first step in the code is to define the variables that are necessary
% for the program to run. Most of the variables are user defined.
%
% *Robot Bound Definitions*
%
% We define the bounds of the error in the robot
% We define them as global variables since they are accessed by several
% functions therefore reducing the need to pass these values everywhere
% clearvars -except cts counter matrix_optimized File_name tr S_name, close
    all

counter=cts;
%%
%clc %clear and close all figures and variables
global INIT_MAX_THETA INIT_MIN_THETA INIT_MAX_X INIT_MAX_Y INIT_MIN_X
    INIT_MIN_Y PENALTYFACTOR
PENALTYFACTOR=75;
INIT_MAX_THETA=pi/36;
INIT_MIN_THETA=-pi/36;
INIT_MAX_X=1.87;
INIT_MAX_Y=24.5; % Can be scalled depending on part geometry
INIT_MIN_X=-1.87;
INIT_MIN_Y=0;
DP_INIT=13;
%DP_constant=13;
xPAPos=2e-1;
xPANeg=-2e-1;
yPAPos=1e-1;
yPANeg=-1e-1;
thetaPAPos=0.0872664626;
thetaPANeg=-0.0872664626;
%%
v0=[0 -1 0];
%%
% We also define the value of the threshold for contact, this is so the
% equations dont have to equal a certain value given some leeway of what is
% possible
global thresh
```

```
thresh=1.0e-3;

%% Object Loading
% We now need to import the necessary information that describes the
% objects that are going to be in contact. This is done by reading a dxf
% file
% The _dxf_read_function is used which requires the input of the file to be
% read and returns it as a structure. For easier batch processing
    csv_loader is used as it can modify
[Obj_A,Obj_B]=csv_loader(20-cts);
%%
%Obj_B.Vertices(2,3:4)=Obj_B.Vertices(2,3:4)+0.01;
Obj_B.Vertices(2,3)=Obj_B.Vertices(2,3)+0.01;
figure2_handle=gca;
%%
% The code then decomposes these objects into convex subobjects that are
% suitable to be used in our feasability and extremal identification
% This section is to be removed, however the subobject decomposition
% function has some issues that have to be addressed hence they are
% hardcoded.

SubObjectA{1}=[[Obj_A.Vertices(:,1:5)],[0;Obj_A.Vertices(2,1)]];
SubObjectA{2}=[[Obj_A.Vertices(:,5:9)],[0;Obj_A.Vertices(2,1)]];
SubObjectB{1}=Obj_B.Vertices;
Obj_B.Vertices=Obj_B.Vertices;

SubObjectA_1C=SubObj2Obj(SubObjectA{1},'cw');
SubObjectA_2C=SubObj2Obj(SubObjectA{2},'cw');
SubObjectB_1C=SubObj2Obj(SubObjectB{1},'ccw');

subObjA{1}=SubObjectA_1C;
subObjA{2}=SubObjectA_2C;
subObjB=SubObjectB_1C;

%% Simple Geometrical Evaluation of Single Point Contacts
% We now use a simple geometric evaluation of the possible single point
% (_V-E_,_E-V_) contact states.
% First we iterate throught all the possible _V-E_ contact states and
% eliminate those that are not feasabile due to the require angle change
```

```
% that would be necessary to produce. We also do not evaluate those that
% correspond to a concave vertex

%V-E Combinations
k=1;
for i=1:1:size(Obj_A.Vertices,2);
    for j=1:1:size(Obj_B.Vertices,2);
        if ismember(i,Obj_A.Concave_List)~=true %check if its a concave
            vertex
            [Ea,Eb]=edge_vertex(Obj_A,i);
            temp=Obj_B.Normal(:,j);
            Angle_a=vector_angle(Ea,-temp);
            Angle_b=vector_angle(Eb,-temp);

            [feasability,mina,maxa]=VE_feasability_Angle(Angle_a,Angle_b);
        else
            feasability=false;
        end
        if feasability==true
            ID=strcat('V',num2str(i),'-E',num2str(j));
            Type='V-E';
            precheck_VE(1,1)=i;
            precheck_VE(1,2)=j;
            precheck_EA=Obj_A.Vertices(:,i);
            precheck_EB=Obj_B.Edges(j);
            % We end up obtaining a list of all the geometrically
                admissible
            % contact states by simple geometrical evaluation
            geo_feasable_csP_VE{k}=struct('Type',Type,'ID',ID,'Index',
                precheck_VE,'Element_A',precheck_EA,'Element_B',precheck_EB
                ,'Min_Angle',mina,'Max_Angle',maxa);
            k=k+1;
        end
    end

end
%%
```

```
%% Comple Geometrical Evaluation of Single Point Contacts V-E

disp('Number of Possible Single Point Contacts after Single Point
    Evauluation')
disp(size(geo_feasable_csP_VE,2))
disp('Begin Complete Geo Evaluation')
k=1;
for i=1:1:size(geo_feasable_csP_VE,2) %Only evaluate those that are
    possible
    cs=geo_feasable_csP_VE{i};
    disp(cs.ID)
    [feas,conf]=checkCsM(cs,subObjA,subObjB); %This function does a lot,
        like seriously alot
    if feas==true
        feasable_csP_VEtemp{k}=cs;
        disp(strcat(cs.ID,' is feasable'))

        k=k+1;
    end
end
%% Phase Ib
% Idenfitying E-V combinations (E is object A, V is object B)
k=1;
for i=1:1:size(Obj_A.Vertices,2);
    for j=1:1:size(Obj_B.Vertices,2);
        try
            Convex_Exist=Obj_B.Concave_List;
        catch err
            Convex_Exist=false;
        end

        if Convex_Exist==false || ismember(j,Obj_B.Concave_List)~=true %
            check if its a concave vertex
            [Ea,Eb]=edge_vertex(Obj_B,j); % obtain the edges that connect
                at vertex to be examined In this case object B
            temp=Obj_A.Normal(:,i); %Now we obtain the normal of the edge
                being investigated
            Angle_a=vector_angle(Ea,-temp); %Obtain angles for each
            Angle_b=vector_angle(Eb,-temp); %Obtain second angle
```

```
            [feasability ,minf,max]=EV_feasability_Angle(Angle_a,Angle_b); %
                Check feasability of EV combination
        else
            feasability=false;
        end

        if feasability==true
            ID=strcat('E',num2str(i),'-V',num2str(j));
            Type='E-V';
            precheck_EV(1,1)=i;
            precheck_EV(1,2)=j;
            precheck_EA=Obj_A.Edges(i);
            precheck_EB=Obj_B.Vertices(:,j);
            % We end up obtaining a list of all the geometrically
                admissible
            % contact states by simple geometrical evaluation
            geo_feasable_csP_EV{k}=struct('Type',Type,'ID',ID,'Index',
                precheck_EV,'Element_A',precheck_EA,'Element_B',precheck_EB
                ,'Min_Angle',minf,'Max_Angle',max); %#ok<SAGROW>
            k=k+1;
        end
    end

end
%% COMPLETE CHECK E-V
% Check all geometrically possible ones
% completely, ie check the bounds etc.
disp('Number of Possible Single Point Contacts after Single Point
    Evauluation')
disp(size(geo_feasable_csP_EV,2))
disp('Begin Complete Geo Evaluation')
k=1;
for i=1:1:size(geo_feasable_csP_EV,2)
    %Only evaluate those that are possible
    cs=geo_feasable_csP_EV{i};
    disp(cs.ID)
    [feas,conf]=checkCsM(cs,subObjA,subObjB);
```

```
    %This function does a lot, like seriously alot
    %       disp(feas)
    if feas==true
        feasable_csP_EVtemp{k}=cs; %#ok<SAGROW>
        disp(strcat(cs.ID,' is feasable'))

        k=k+1;
    end
end
%% Extremals for Single Point Contact
% This is done at this point to reduce the number of unnecessary
% calculations further along the program, why waste time evaluating the
% contact states that are already considered properly assembled

for i=1:1:length(feasable_csP_VEtemp)
    feasable_csP_VEtemp{i}.ID
    [temp]=get_bounds_csPM(feasable_csP_VEtemp{i},Obj_A,Obj_B,subObjA,
        subObjB);
    feasable_csP_VEtemp{i}=set_ex(temp); %#ok<SAGROW>
end

%%
for i=1:1:length(feasable_csP_EVtemp)
    feasable_csP_EVtemp{i}.ID
    temp=get_bounds_csPM(feasable_csP_EVtemp{i},Obj_A,Obj_B,subObjA,subObjB
        );
    feasable_csP_EVtemp{i}=set_ex(temp); %#ok<SAGROW>
end

%%
% Set proper bounds for single point contacts, meaning succesful assemblies
feasable_csP_VE=set_proper(feasable_csP_VEtemp,xPAPos,xPANeg,yPAPos,yPANeg,
    thetaPAPos,thetaPANeg);
%%
feasable_csP_EV=set_proper(feasable_csP_EVtemp,xPAPos,xPANeg,yPAPos,yPANeg,
    thetaPAPos,thetaPANeg);
clearvars temp
%% Summary Phase I
% The following just diplays the CS that have been found feasabile
```

```
% (Primitives)
disp('%-----------------------------------%')
disp('Feasable Single Point Contact States')

for i=1:1:size(feasable_csP_VE,2)
    disp(feasable_csP_VE{i}.ID)
end

for i=1:1:size(feasable_csP_EV,2)
    disp(feasable_csP_EV{i}.ID)
end
disp('%-----------------------------------%')

%%
% Reorganize the obtained contact states
k=1;
feasable_csP=cell(1,length(feasable_csP_VE)+length(feasable_csP_VE));
for i=1:1:size(feasable_csP_VE,2)
    feasable_csP{k}=feasable_csP_VE{i};
    k=k+1;
end

for i=1:1:size(feasable_csP_EV,2)
    feasable_csP{k}=feasable_csP_EV{i};
    k=k+1;
end
%% Phase II
% Face Two identifies the contact states that are possible for a two point
% contact such as <V-E, E-V> or vice versa
k=1;
s=1;
for i=1:1:size(feasable_csP_VE,2)
    for j=1:1:size(feasable_csP_VE,2)
        %Iterate through every possible combination of primitive PC's
        cs1=feasable_csP_VE{i};
        %Assign cs
        cs2=feasable_csP_VE{j};
        Type='V-E V-E'; %Assign type
        ID=strcat(cs1.ID,32,cs2.ID); % Create ID
```

```
            Element_1=cs1.Element_A;
            Element_2=cs1.Element_B;
            Element_3=cs2.Element_A;
            Element_4=cs2.Element_B;

            if(i~=j)%Get rid of repeated ones <V1-E1><V1-E1>
                if(s==1) %Special case for first combo possible
                    list{s}=strcat(num2str(i),32,strcat(num2str(j)));
                    s=s+1;
                    [feasable]=feasability_VE_VE(cs1,cs2,Obj_A,Obj_B);%Check if
                         it can be done
                    %Create the contact state
                    if feasable
                        precheck_feasable_cs{k}=struct('Type',Type,'ID',ID,'
                            Index',strcat(num2str(i),32,strcat(num2str(j))),'
                            CS_1',cs1,'CS_2',cs2,'Element_3',Element_3,'
                            Element_4',Element_4);
                        k=k+1;
                    end

                elseif(false==any(strcmp(strcat(num2str(j),32,strcat(num2str(i)
                    )),list)))
                    %Check that this concact state hasn't been created before
                    list{s}=strcat(num2str(i),32,strcat(num2str(j)));
                    s=s+1;
                    [feasable]=feasability_VE_VE(cs1,cs2,Obj_A,Obj_B);
                    %Check if it can be done
                    if feasable
                        precheck_feasable_cs{k}=struct('Type',Type,'ID',ID,'
                            Index',strcat(num2str(i),32,strcat(num2str(j))),'
                            CS_1',cs1,'CS_2',cs2,'Element_3',Element_3,'
                            Element_4',Element_4);
                        k=k+1;
                    end
                end
            end
        end
    end
```

```
% %Display found contact states precheck
% for i=1:1:size(precheck_feasable_cs,2)
%     disp(precheck_feasable_cs{i}.ID)
% end
%% <E-V, E-V>
s=1;
clear list
for i=1:1:size(feasable_csP_EV,2)
    for j=1:1:size(feasable_csP_EV,2)
        %Iterate through every possible combination of primitive PC's
        cs1=feasable_csP_EV{i}; %Assign cs
        cs2=feasable_csP_EV{j};
        Type='E-V E-V'; %Assign type
        ID=strcat(cs1.ID,32,cs2.ID); % Create ID
        Element_1=cs1.Element_A;
        Element_2=cs1.Element_B;
        Element_3=cs2.Element_A;
        Element_4=cs2.Element_B;

        if(i~=j)%Get rid of repeated ones ie(<V1-E1><V1-E1>)
            disp(ID)
            if(s==1) %Special case for first combo possible
                list{s}=strcat(num2str(i),32,strcat(num2str(j)));
                s=s+1;
                [feasable]=feasability_EV_EV(cs1,cs2,Obj_A,Obj_B)
                %Check if it can be done
                %Create the contact state
                if feasable
                    precheck_feasable_cs{k}=struct('Type',Type,'ID',ID,'
                        Index',strcat(num2str(i),32,strcat(num2str(j))),'
                        CS_1',cs1,'CS_2',cs2,'Element_3',Element_3,'
                        Element_4',Element_4);
                    k=k+1;
                end


            elseif(false==any(strcmp(strcat(num2str(j),32,strcat(num2str(i)
                )),list)))
                %Check that this concact state hasn't been created before
```

```
                list{s}=strcat(num2str(i),32,strcat(num2str(j)));
                s=s+1;
                [feasable]=feasability_EV_EV(cs1,cs2,Obj_A,Obj_B)
                if feasable
                    precheck_feasable_cs{k}=struct('Type',Type,'ID',ID,'
                        Index',strcat(num2str(i),32,strcat(num2str(j))),'
                        CS_1',cs1,'CS_2',cs2,'Element_3',Element_3,'
                        Element_4',Element_4);
                    k=k+1;
                end
            end
        end
    end
end
%
% %Display found contact states precheck
% for i=1:1:size(precheck_feasable_cs,2)
%     disp(precheck_feasable_cs{i}.ID)
% end
%% <V-E, E-V> == <E-V, V-E>
s=1;
k=1;
clearvars list
for i=1:1:size(feasable_csP_VE,2)
    for j=1:1:size(feasable_csP_EV,2)
        %Iterate through every possible combination of primitive PC's
        cs1=feasable_csP_VE{i}; %Assign cs
        cs2=feasable_csP_EV{j};
        Type='V-E E-V'; %Assign type
        ID=strcat(cs1.ID,32,cs2.ID); % Create ID
        Element_1=cs1.Element_A;
        Element_2=cs1.Element_B;
        Element_3=cs2.Element_A;
        Element_4=cs2.Element_B;



        if(s==1) %Special case for first combo possible
            list{s}=strcat(num2str(i),32,strcat(num2str(j)));
```

```
            s=s+1;
            [feasable]=feasability_VE_EV(cs1,cs2,Obj_A,Obj_B);
            %Check if it can be done
            %Create the contact state
            if feasable
                precheck_feasable_cs{k}=struct('Type',Type,'ID',ID,'Index',
                    strcat(num2str(i),32,strcat(num2str(j))),'CS_1',cs1,'
                    CS_2',cs2,'Element_3',Element_3,'Element_4',Element_4);
                k=k+1;
            end



        elseif(false==any(strcmp(strcat(num2str(j),32,strcat(num2str(i))),
            list)))
            %Check that this concact state hasn't been created before
            list{s}=strcat(num2str(i),32,strcat(num2str(j)));
            s=s+1;
            [feasable]=feasability_VE_EV(cs1,cs2,Obj_A,Obj_B);
            %Check if it can be done
            if feasable
                precheck_feasable_cs{k}=struct('Type',Type,'ID',ID,'Index',
                    strcat(num2str(i),32,strcat(num2str(j))),'CS_1',cs1,'
                    CS_2',cs2);
                k=k+1;
            end
        end
    end
end

% %Display found contact states precheck
% for i=1:1:size(precheck_feasable_cs,2)
%     disp(precheck_feasable_cs{i}.ID)
% end

%%
k=1;
for i=1:1:size(precheck_feasable_cs,2) %Only evaluate those that are
    possible
```

```
    cs=precheck_feasable_cs{i};
    disp(cs.ID)
    [feas,conf]=checkCsM(cs,subObjA,subObjB); %This function does a lot,
        like seriously alot
    %     disp(feas)
    if feas==true
        feasable_cs{k}=cs;
        disp(strcat(cs.ID,' is feasable'))
        conf_sel_cs{k}=conf;
        k=k+1;
    end
end
%%
no_2P=false;
try
     feasable_cs;
catch
    no_2P=true;
    disp('%-----------------------------------%')
    disp('No Two Point Contacts Possible')
    disp('%-----------------------------------%')
end

%%
if no_2P==false
    disp('%-----------------------------------%')
    disp('Feasable Two Point Contact States')
    for i=1:1:size(feasable_cs,2)
        disp(feasable_cs{i}.ID)
    end
    disp('%-----------------------------------%')
end
% In this face we will identify the extremals
% clear feasable_csP_EV*
% clear feasable_csP_VE*
% clearvars -except subObjA subObjB Obj_A Obj_B thresh INIT* figure1_handle
    feasable* PENALTYFACTOR DP_* xPA* yPA* thetaPA*
%%
```

```
if no_2P==false
    % Extremals of two point contact
    cs2P=cell(1,length(feasable_cs));
    for i=1:1:length(feasable_cs)
        feasable_cs{i}.ID
        %Get the actall bounds
        cs2Ptemp{i}=get_bounds_cs2PM(feasable_cs{i},Obj_A,Obj_B,subObjA,
            subObjB,INIT_MAX_X, INIT_MAX_Y, INIT_MIN_X, INIT_MIN_Y,
            INIT_MAX_THETA,INIT_MIN_THETA);
    end

    %%
    csP=feasable_csP;
    %%
    %Identify the extremals
    for i=1:1:length(cs2P)
        cs2P{i}=set_ex2(cs2Ptemp{i});
    end
end

csP=feasable_csP;
%%
% clearvars -except subObjA subObjB Obj_A Obj_B thresh INIT* figure1_handle
    PENALTYFACTOR DP_* xPA* yPA* thetaPA* csP cs2P
%%V-E Decomposition phase
%Phase III+ V-E Decomposition (also anything that ahs a V-E contact in it
%aka, everything is a waste of time pretty much unless the V-E was an E-V
%or an E-V E-V
%%
V_E_no=0;
for i=1:1:length(csP)
    cs=csP{i};
    if strcmp(cs.Type,'V-E')
        V_E_no=1+V_E_no;
    end
end
%Calculate the decomposition constant
DP_constant=getdecomp(feasable_csP_EV,v0,DP_INIT);
if no_2P==false
```

```
    extremal_list_csP_size=length(csP)-V_E_no+V_E_no*DP_constant+length(
        cs2P)*DP_constant;
    %%We want to know how big to make our list
else
    extremal_list_csP_size=length(csP)-V_E_no+V_E_no*DP_constant;
    %%We want to know how big to make our list
end
extremal_list_csP=cell(1,extremal_list_csP_size);

k=1;
for i=1:1:length(csP)
    cs=csP{i};
    if strcmp(cs.Type,'V-E')
        temp=bounds_VE_decompose(cs,DP_constant,subObjA,subObjB,Obj_B);
        for c1=1:1:length(temp)
            extremal_list_csP{k}=temp{c1};
            k=k+1;
        end
    else
        extremal_list_csP{k}=cs;
        k=k+1;
    end
end

%% If two point contat occurs also decompose the two point contacts that
    contain V-E
if no_2P==false
    for i=1:1:length(cs2P)
        cs=cs2P{i};
        if strcmp(cs.Type,'V-E V-E') || strcmp(cs.Type, 'V-E E-V')
            temp=bounds_VE2_decompose(cs,DP_constant,subObjA,subObjB,Obj_B)
                ;
            for c1=1:1:length(temp)
                extremal_list_csP{k}=temp{c1};
                k=k+1;
            end
        else
            extremal_list_csP{k}=cs;
            k=k+1;
```

```
        end
    end
end
%%
v0=[0 -1 0];
%%
% Kind of a hack to resize the vertices to their actual size so we dont
% get strange values, needs to be changed for different geometries, can be
% taken as an input but its hardcoded for batch work

%Obj_B.Vertices(2,3:4)=Obj_B.Vertices(2,3:4)-0.01;
Obj_B.Vertices(2,3)=Obj_B.Vertices(2,3)-0.01;
%%
% Adjust the home position as explained in literature
E=adjust_home(v0,Obj_B,INIT_MAX_THETA);
%%
home=transfer_N(Obj_B.Vertices,[0 E 0]');
%%
print_ACSG(extremal_list_csP,File_name)

%%
%%
[opt_matrix]=id_matrix(extremal_list_csP,v0,home,Obj_A,Obj_B);
%%
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
%This performs the friction based optimization
disp('Friction Optimization')
matrix_optimized_friction=opt_matrix;
%Save the variablse
save(S_name)
%%
% This all prepares the extremals for the velocity optimization
if no_2P==false
    P2_No=length(cs2P);
    Tot_Ex=length(extremal_list_csP);


    ks=1;
    for count=1:1:P2_No*2
```

```
        if count==1
            No(ks)=Tot_Ex-P2_No*DP_constant+1;

        else
            if mod(count,2)==0 %even
                ks=ks+1;

                No(ks)=No(ks-1)+DP_constant-1;
            else
                ks=ks+1;

                No(ks)=No(ks-1)+1;
            end
        end
    end

    No=reshape(No,[],2)';

  % if two point contat occurs, generate new extremals with appropirate
  % bounds
    for i=1:1:P2_No
        extremals_2P_Vel{i}=extremal_list_csP{No(i,1)};
        extremals_2P_Vel{i}=rmfield(extremals_2P_Vel{i},'lower_bounds');
        extremals_2P_Vel{i}=rmfield(extremals_2P_Vel{i},'upper_bounds');
        extremals_2P_Vel{i}.Ext1=extremal_list_csP{No(i,1)};
        extremals_2P_Vel{i}.Ext2=extremal_list_csP{No(i,2)};
    end
    extremal_list=[feasable_csP,extremals_2P_Vel];
else
    extremal_list=feasable_csP;
end
%%
% Generate the extremal list for the velocity optimizations
for i=1:length(extremal_list)
    extremal_list{i}=possible_extremals_csP(extremal_list{i},Obj_A,Obj_B,
        subObjA,subObjB,INIT_MAX_THETA,INIT_MIN_THETA,thresh);
end
%%
% This deals with the minimax optimization using matlabs fminimax, this is
```

```
% only used for comparison and can be commented out
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('Fminimax Vm')
[matrix_velocity,Vm_min,gmin]=id_vm_matrix(extremal_list,extremal_list_csP,
    v0,home,Obj_A,Obj_B,friction);
matrix_optimized_velocity=matrix_velocity;
save(S_name);

%% This section deals with the Vm Optimization using the genetic algorithm
disp('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
disp('Genetic Vm')
%define the bounds of the optimization, this are selected through trial and
%error and observing results for friction
lb=[-1,-1,-2.5,-2.5,-2.5];
ub=[7,7,2.5,2.5,2.5];
%load our inital guesses
X_init_dat;
%Get the length of the part
L=getpartlength(Obj_B,v0);
%transform the inital points into the compliant matrix
XINIT=compliant_trans(X_INIT_NORM,L);
%inser initial guesses into population
sample_temp=reshape(cell2mat(XINIT),6,length(cell2mat(XINIT))/6)';
sample=[sample_temp(:,2:end)];
%Create option structure for the genetic algorithm
opga=gaoptimset('Initialpopulation',sample,'PopInitRange',[lb;ub],'
    CrossoverFraction',0.3,'EliteCount',3,'Display','iter','FitnessScaling'
    ,@fitscalingrank,'Generations',300,'PopulationSize',30,'StallGenLimit'
    ,15,'TolFun',1e-8,'UseParallel','always','TolCon',1e-8,'MutationFcn',
    @mutationadaptfeasible);
%loop through several iterations
for ij=1:1:4
    [mvgt{ij},Vm_min_ga_t(ij),eft(ij)]=ga(@(x)Overal_Vm_ga(x,extremal_list,
        Obj_A,Obj_B,0,friction),5,[],[],[],[],[],[],@(x)AdConstraints_vel(x,
        extremal_list_csP,v0,home,Obj_A,Obj_B,friction),opga);
end
%%
% Record important values and save all variables
clearvars min
```

```
[Vm_min_ga, itera]=min(Vm_min_ga_t);
mvg=mvgt{itera};
save(S_name);
```

H GET_VM_VE

The following presents the code used to calculate the V_m of a given admittance for a V-E contact. This function is presented as an example of how the process is performed for all contact states.

```
function [Vm]=get_Vm_VE(cs,Obj_A,Obj_B,v0,l,ex,A,mu,pl)
%% This function takes as input the contact state, object descriptions,
    nominal twists, length of the moving part, extremal number, admittance
    and friction value. pl is used for printing purposes
switch ex
    case 1 %min %min
        preconf=cs.Pos_EX{1};
        trans=preconf(2);
        rot=preconf(1);
        conf=confVE(cs,trans,rot);
    case 2 %min %max
        preconf=cs.Pos_EX{2};
        trans=preconf(2);
        rot=preconf(1);
        conf=confVE(cs,trans,rot);

    case 3 %max %min
        preconf=cs.Pos_EX{3};
        trans=preconf(2);
        rot=preconf(1);
        conf=confVE(cs,trans,rot);

    case 4 %max %max
        preconf=cs.Pos_EX{4};
        trans=preconf(2);
        rot=preconf(1);
        conf=confVE(cs,trans,rot);

end

%Grab all the Vertices and transform (For debugging)
```

```
for i=1:1:length(Obj_B.Vertices)
    V=Obj_B.Vertices(:,i);
    Vertices_world(:,i)=transfer(V,conf);
    Vector_w(:,i)=-Vertices_world(:,i)+Obj_B.Vertices(:,i); %Vector from
        part to zero
    Vector_body(:,i)=transfer(Vector_w(:,i),[0,0,-conf(3)]'); %Vector from
        part to zero in body frame
end

%%Necessary to calculate the Centroid of the Body
Verts=[Obj_B.Vertices,Obj_B.Vertices(:,1)];
n=length(Verts);
x=Verts(1,:);
y=Verts(2,:);
%%
for i=1:1:n-1
Cx(i)=(x(i)+x(i+1))*(x(i)*y(i+1)-x(i+1)*y(i));
Cy(i)=(y(i)+y(i+1))*(x(i)*y(i+1)-x(i+1)*y(i));
Ar(i)=x(i)*y(i+1)-x(i+1)*y(i);
end
At=(1/2)*sum(Ar);
Cxt=(1/(6*At))*sum(Cx);
Cyt=(1/(6*At))*sum(Cy);
G=[Cxt;Cyt];



%Transfer to world frame
G_world=transfer(G,conf);
G_Vector_w=-G_world+G; %Vector from part to zero
G_Vector_body=transfer(G_Vector_w,[0,0,-conf(3)]'); %Vector from part to
    zero in body frame

%vector from current to desired
G_prime=G_world+G_Vector_w;

% Identify the best motion
r_i=G_Vector_body;
if(conf(3)~=0)
```

```
    theta=-conf(3);
    %%CHANGED
    omega_i=(sin(theta)/theta)*theta;
else
    omega_i=0;
end
t=1;
t_i=[r_i/t;omega_i/t];

%Coordinate Axis
f=[0,0]';
f_world=transfer(f,conf);

ro_world=Obj_A.Vertices(:,cs.Index(1))-f_world;
ro_body=transfer(ro_world,[0 0 -conf(3)]');


%Calculate Contact Wrenches
t=cs.trans_bound.direction;
if dot([t;0],v0)>=0
    t=-t;
end
n=-[Obj_B.Normal(:,(cs.Index(2)));0];

r01 = ro_body(1);
r02 = ro_body(2);
n1 = n(1);
n2 = n(2);
t1 = t(1);
t2 = t(2);

t_n = [n1; n2; r01 * n2 - r02 * n1];
t_t = [t1; t2; r01 * t2 - r02 * t1];

%Calculate twist t
B = t_n'*A*t_n + t_n'*A*t_t*mu;
phi = -inv(B)*t_n'*v0;

t0_C=v0+A*(t_n+t_t*mu)*phi;
```

```
%Transform to centroid, currently expressed from top of frame
p_cross=-[0 0 G(2);0,0,-G(1);-G(2),G(1),0];
s_t_g=-[eye(3),p_cross*eye(3);zeros(3),eye(3)];
t0_C_R6=[t0_C(1);t0_C(2);0;0;0;t0_C(3)];
t_0=-s_t_g*t0_C_R6;


disregard=0;

%Structure the ideal twist
Vi=t_i(1:2);
V0=t_0(1:2);
Pi=Vi;
%Time Constant Calculation
ti=sqrt(Vi(1)^2+Vi(2)^2)/sqrt(v0(1)^2+v0(2)^2);
Vi=Vi/ti;

%%This is to guard againts abberant contact states
if abs(Pi(1))<0.51 && abs(Pi(2))<0.05
    disregard=1;
end

%Calculate differences
omegai=t_i(3);
omegai=omegai/ti;
omega0=t_0(6);
delta_V=V0-Vi;
delta_omega=omega0-omegai;
checks=1;

%Is delta_omega is zero the use one expression
if delta_omega==0 || abs(delta_omega)<0.01

    Vm=sqrt(delta_V(1)^2+delta_V(2)^2);
    checks=0;

end
%Calculate the location of the difference rotation center
omega_cross=-[0,delta_omega;-delta_omega,0];
```

```
rgo=-inv(omega_cross)*delta_V;
rgo_world=transfer(rgo,[0 0 conf(3)]');

%----------------------DEBUG-----------------------------------%
%Locate C to the edge of the rectange (For debugging only
C=Obj_B.Vertices(:,2)-G;
C_world=transfer(C,[0 0 conf(3)]');

rco=C-rgo;
rco_world=transfer(rco,[0 0 conf(3)]');

l1=1;%Change Depending on Part
l2=l;

a=rco(1);
b=a+l1;
c=rco(2);
d=c+l2;
%--------------------------------------------------------------%
%%Routine of Triangle Parts
%Decompose part into triangles
[Body_Tri,FlagsTri]=Rbar_Decomp(Obj_B.Vertices,0);
size_BodyTri=size(Body_Tri);
rtAr=[];
rtAr_neg=[];
Area=[];
Area_neg=[];
%For each triangle calculate rbar and area, determine if its a whole or not
for i=1:1:size_BodyTri(2)
    BTri=Body_Tri(:,i);
    Flag=FlagsTri(i);
    W=BTri(3);
    H=BTri(4);
    if Flag~=-1
        Area_t=(W*H)/2;
        Area=[Area,Area_t];
        if (H~=0 && W~=0)
            [rt]=rbartri(BTri,rgo,G);
            rtAr=[rtAr,rt*Area_t];
```

```
            else
                rtAr=[rtAr ,0];
            end
        elseif Flag==-1
            Area_t=(W*H)/2;
            Area_neg=[Area_neg ,Area_t];
            if (H~=0 && W~=0)
                [rt]=rbartri(BTri,rgo,G);
                rtAr_neg=[rtAr_neg ,rt*Area_t];
            else
                rtAr_neg=[rtAr_neg ,0];
            end
        end
    end
    %Obtain Total r
    rbar_t=(sum(rtAr)-sum(rtAr_neg)/(sum(Area)-sum(Area_neg)));
    r=rbar_t;

    %-----------------DEBUG-----------------------------------%
    rss1=(1/(6*l1*l2));
    rss2=2*a*c*sqrt(a^2+c^2)+a^3*log(c+sqrt(a^2+c^2))-2*b*c*sqrt(b^2+c^2);
    rss3=-b^3*log(c+sqrt(b^2+c^2))-2*a*d*sqrt(a^2+d^2)-a^3*log(d+sqrt(a^2+d^2))
        ;
    rss4=2*b*d*sqrt(b^2+d^2)+b^3*log(d+sqrt(b^2+d^2))-d^3*log(a+sqrt(a^2+d^2));
    rss5=c^3*log(a+sqrt(a^2+c^2))+d^3*log(b+sqrt(b^2+d^2))-c^3*log(b+sqrt(b^2+c
        ^2));
    rsss=rss1*(rss2+rss3+rss4+rss5);
    %---------------------------------------------------------%

    if checks==1
        Vm=abs(delta_omega)*r;
    end

    %If its already assembled then the Value is zero so it doesnt affect the
    %optimization
    if disregard==1
        Vm=0;
    end
```

```
%----------------------DEBUG PRINT----------------------%
if pl==1
    figure(1)
axis equal
plot(Obj_B.Vertices(1,:),Obj_B.Vertices(2,:))
hold all
plot(Obj_A.Vertices(1,:),Obj_A.Vertices(2,:))
plot(G_world(1),G_world(2),'or')
plot(G(1),G(2),'ob')
plot([G_world(1),G_prime(1)],[G_world(2),G_prime(2)],'--+k')
plot([Vertices_world(1,:),Vertices_world(1,1)],[Vertices_world(2,:),
    Vertices_world(2,1)],'-m')
plot(G_world(1)+rgo_world(1),G_world(2)+rgo_world(2),'ok');
plot(f_world(1),f_world(2),'r+','MarkerSize',14)
plot([f_world(1),f_world(1)+ro_world(1)],[f_world(2),f_world(2)+ro_world(2)
    ],'m')
plot(G_world(1)+C_world(1),G_world(2)+C_world(2),'*')
plot(conf(1),conf(2),'or')
axis equal
end

end

%Obtain the configuration (x,y,theta) from two variables (delta, theta)
function conf=confVE(cs,trans,rot)
preconf=[trans,rot];
delta=preconf(1);
theta=preconf(2);
rA=cs.Element_A;
dirB=((cs.Element_B(:,1)-cs.Element_B(:,2))/norm(cs.Element_B(:,1)-cs.
    Element_B(:,2))); %%% Changed
bd1=cs.Element_B(:,1);
bd2=cs.Element_B(:,2);
norms=norm(cs.Element_B(:,1)-cs.Element_B(:,2));
if bd1+dirB*norms==bd2
    cPt=bd1;
else
    cPt=bd2;
end
```

```
rB=delta*dirB+cPt;
rAB=rA-transfer_Ns(rB,[0;0;theta]);
conf=[rAB;theta];
end

% Calculation of rbar for a given triange
function [rt]=rbartri(Rtri,Pt,G)
% locate the pivot point, its width and height and rotation angle
xh=Rtri(1);
yh=Rtri(2);
W=Rtri(3);
H=Rtri(4);
alpha=Rtri(5);
% IC location
P=Pt;
C=[xh;yh]-G;
rcot=C-P;
rco=transfer(rcot,[0;0;alpha]);
x_co=rco(1);
y_co=rco(2);
a=x_co;
b=x_co+W;
c=y_co;
d=y_co+H;
m=(b-a)/(c-d);
g= a-m*d;

u = sqrt(1+m^2);
v= sqrt(c^2+c^2*m^2+2*m*c*g+g^2);
w= sqrt(d^2+d^2*m^2+2*m*d*g+g^2);

%Components of Rbar equation
rt1=(1/(3*W*H*u^3));
rt2=u*m^2*d^3*log(m*d+g+w);
rt3=2*a*c*u*sqrt(a^2+c^2)-2*a*d*u*sqrt(a^2+d^2);
rt4=-m*u*v*c^2+g^3*log((d+d*m^2+m*g+w*u)/u);
rt5=-g^3*log((c+c*m^2+m*g+v*u)/u)-2*g*c*v*u*m^2;
rt6=-u*m^2*c^3*log(m*c+g+v)-m*u*v*g^2+m*u*w*g^2;
rt7=2*d*g*u*w-2*c*g*u*v+u*d^3*log(m*d+g+w);
```

```
rt8=-u*c^3*log(m*c+g+v)+u*c^3*log(a+sqrt(a^2+c^2));
rt9=u*a^3*log(c+sqrt(a^2+c^2))-u*d^3*log(a+sqrt(a^2+d^2));
rt10=-u*a^3*log(d+sqrt(a^2+d^2))+2*d*g*u*w*m^2;
rt11=2*a*c*u*m^2*sqrt(a^2+c^2)-2*a*d*u*m^2*sqrt(a^2+d^2);
rt12=-u*m^2*a^3*log(d+sqrt(a^2+d^2));
rt13=-u*m^2*d^3*log(a+sqrt(a^2+d^2));
rt14=u*m^2*c^3*log(a+sqrt(a^2+c^2));
rt15=u*m^2*a^3*log(c+sqrt(a^2+c^2));
rt16=-u*v*c^2*m^3+u*w*d^2*m^3+m*u*w*d^2;

%Rbar calculated
rt=rt1*(rt2+rt3+rt4+rt5+rt6+rt7+rt8+rt9+rt10+rt11+rt12+rt13+rt14+rt15+rt16)
    ;
end
```