An Emulator-Based Prediction of Dynamic Stiffness for Redundant Parallel Kinematic Mechanisms

by

Mario Luces Rosado

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science

Department of Mechanical and Industrial Engineering
University of Toronto

© Copyright by Mario Luces 2015
Abstract

For Parallel Kinematic Mechanisms (PKMs) with kinematic redundancy, mechanism configurations with higher stiffness can be chosen during motion-trajectory planning. However, substantial computational resources would be required for this optimization problem, specifically, for the solution of the two intertwined sub-problems: (i) calculation of the dynamic stiffness of any considered PKM configuration, at a given task-space location, and (ii) searching for the PKM configuration with the highest stiffness at this location. Herein, the former sub-problem is addressed via a novel effective emulator to provide a computationally efficient approximation of the dynamic-stiffness function suitable for optimization.

The proposed method for emulator development identifies the mechanism’s structural modes in order to break down the high-dimensional stiffness function into multiple functions of lower dimension. Via extensive simulations, some of which are described herein, it is demonstrated that the proposed emulator can predict the dynamic stiffness of a PKM at any given configuration with high accuracy and at a low computational expense.
Acknowledgments

I would like to thank my family for their continuous support and encouragement. A very special thank-you to my parents Mario Luces and Lila Rosado not only for their continuous support, but for being the very reason I was able to start this great project. I would like to thank you for listening, for your advice, and for your trust.

I would like to thank my supervisors Prof. Beno Benhabib and Prof. James K. Mills for their guidance and mentorship, for always reminding me the importance of making a stop and think, for sharing their ideas, for their patience and honest advice, and for believing I could always do better.

I would like to thank my colleagues and friends from the Computer Integrated Manufacturing (CIM) Lab. A special thanks to Mr. Julio Vilela, Dr. Masih Mahmoodi, Dr. Adam Le, Mr. Evgeny Nuger, Mr. Ray Zhao, Mr. Arta Alagheband, Mr. Kerry Zhou and Mr. Mengzhe Zu. The ride would have not been the same without you.

Finally, I would like to acknowledge the Natural Science and Engineering Research Council of Canada (NSERC) - Canadian Network for Research and Innovation in Machining Technology (CANRIMT) for providing funding for this project.
# Table of Contents

Acknowledgments ........................................................................................................ iii

Table of Contents ........................................................................................................ iv

List of Tables ................................................................................................................ vi

List of Figures .............................................................................................................. vii

List of Appendices ..................................................................................................... viii

Nomenclature ............................................................................................................. ix

Chapter 1 Introduction ............................................................................................... 1

1 Introduction ............................................................................................................. 1

1.1 Redundant PKMs ................................................................................................. 1

1.2 Benefits of Kinematic Redundancy .................................................................... 2

1.2.1 Reachable and Dexterous Workspace ............................................................. 3

1.2.2 Singularities and Dexterity ............................................................................ 4

1.2.3 Joint Torques: Stiffness, Load Capability and Platform Motion .................... 6

1.2.4 Trajectory Planning ....................................................................................... 7

1.3 Dynamic Stiffness Optimization ........................................................................ 10

1.3.1 Emulators ....................................................................................................... 10

1.4 Thesis Objectives and Contributions ................................................................ 12

1.5 Thesis Outline .................................................................................................... 13

Chapter 2 Problem Statement .................................................................................. 15

2 Problem Statement ................................................................................................ 15

2.1 Dynamic Stiffness ............................................................................................. 15

2.2 Prediction ............................................................................................................ 16

2.3 Configuration Selection ..................................................................................... 17

Chapter 3 Methodology ............................................................................................. 20

3 Methodology .......................................................................................................... 20
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Single Mode and Artificial Neural Networks</td>
<td>21</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Single Frequency Mode</td>
<td>21</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Artificial Neural Networks</td>
<td>23</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Shortcomings</td>
<td>24</td>
</tr>
<tr>
<td>3.2</td>
<td>Structural Modes</td>
<td>26</td>
</tr>
<tr>
<td>3.3</td>
<td>Training Data</td>
<td>28</td>
</tr>
<tr>
<td>3.4</td>
<td>Moving-Least-Squares Approximation</td>
<td>32</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>Case Study</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>A CASE STUDY: AN EMULATOR FOR A 3×PPRS REDUNDANT PKM</td>
<td>35</td>
</tr>
<tr>
<td>4.1</td>
<td>The PKM and FE Model</td>
<td>35</td>
</tr>
<tr>
<td>4.2</td>
<td>Sample Tool Platform Path and Training Configurations</td>
<td>37</td>
</tr>
<tr>
<td>4.3</td>
<td>Example Emulator Design</td>
<td>39</td>
</tr>
<tr>
<td>4.4</td>
<td>Simulation Results</td>
<td>41</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Comparing Emulator and FEA Results</td>
<td>41</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Determining Optimal PKM Configuration</td>
<td>42</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Comparison</td>
<td>44</td>
</tr>
<tr>
<td>4.4.4</td>
<td>Robustness</td>
<td>46</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Conclusions and Recommendations</td>
<td>49</td>
</tr>
<tr>
<td>5</td>
<td>Conclusions and Recommendations</td>
<td>49</td>
</tr>
<tr>
<td>5.1</td>
<td>Conclusions</td>
<td>49</td>
</tr>
<tr>
<td>5.2</td>
<td>Recommendations</td>
<td>50</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>52</td>
</tr>
<tr>
<td>Appendices</td>
<td></td>
<td>59</td>
</tr>
</tbody>
</table>
List of Tables

Table 1. Overview of methods to estimate dynamic stiffness. .................................................. 30

Table 2. Displacement at optimum redundant configuration.................................................. 43

Table 3. Statistical analysis of displacement data.................................................................. 45

Table 4. Robustness analysis for the proposed emulator – Displacement Errors................. 48
List of Figures

Figure 1. Proposed prediction methodology................................................................. 21

Figure 2. (a) A 6-dof 3×PPRS PKM, (b) Example FRF of the amplitude of the displacement. 27

Figure 3. Example FRFs of the compliance of a PKM configuration. ............................... 31

Figure 4. Architecture of the 3×PPRS PKM................................................................ 36

Figure 5. Sample path of PKM tool-platform..................................................................... 37

Figure 6. Training poses and sample path of PKM tool-platform........................................ 39

Figure 7. Prediction of displacement of tool-platform data.................................................. 42

Figure 9. Displacements at configurations near optimum.................................................... 44

Figure 10. Optimal configuration selection by emulator in comparison to random selection.. 45

Figure 11. Emulator predictions using noisy training data................................................. 47

Figure A1. Symmetric PKM configurations................................................................. 59

Figure A2. Side-by-side comparison of a PKM configuration to 3 symmetric configurations... 61

Figure B1. Joint-Space trajectories.................................................................................. 62
List of Appendices

APPENDIX A: Symmetric Configurations in Emulator Training .................................................. 59

APPENDIX B: Joint-Space Trajectories ...................................................................................... 62
Nomenclature

Symbols

\( C \) compliance matrix
\( C_k \) compliance matrix at structural mode \( k \)
\( C_{k,ij} \) element of the compliance matrix at structural mode \( k \)
\( D_u \) projection of the displacement of the tool-platform center along the direction \( u \) (shortened to simply \( D \) in the text)
\( D_o \) maximum projection of the displacement of the tool-platform center along the direction \( u \), in the frequency range (\( D_{uo} \) shortened to \( D_o \))
\( D_k \) projection of the displacement of the tool-platform center along the direction \( u \) at structural mode \( k \) (\( D_{uk} \) shortened to \( D_k \))
\( F \) force applied to the center of the tool-platform
\( M \) number of multivariate polynomial functions
\( N \) number of training configurations
\( S \) directional dynamic stiffness
\( U \) approximation function
\( a_{min} \) joint acceleration lower limit
\( a_{max} \) joint acceleration upper limit
\( d \) degree of the polynomial functions
\( f \) frequency of oscillation of the applied force
\( f_o \) frequency at which the applied force produces the largest magnitude of \( D_u \)
\( f_k \) frequency of the \( k \) structural mode
\( i \) index for axis of displacement response
\( j \) index for axis of applied force
\( k \) structural mode number
\( l \) index for PKM leg
\( m \) number of structural modes of the PKM
\( n \) number of independent active joints in the PKM
\( p \) \( n \)-variate polynomial function
\( \mathbf{q} \) joint-space vector that represents the PKM configuration

\( r \) distance function

\( s \) dimension of function to approximate

\( \mathbf{u} \) unit vector along the direction of \( \mathbf{F} \)

\( v \) index for approximation function term

\( x \) index of component of vector \( \mathbf{q} \)

\( y_{\text{min}} \) joint displacement lower limit

\( y_{\text{max}} \) joint displacement upper limit

\( \alpha \) coefficients of polynomial functions

\( \delta_l \) Linear position of the second prismatic joint at PKM leg \( l \)

\( \gamma \) index for training configuration

\( \theta_l \) angular position of the first curvilinear joint at PKM leg \( l \)

\( \omega \) weight function

**Acronyms**

- ANN Artificial Neural Network
- DE Differential Evolution
- DOF Degrees of Freedom
- EMA Experimental Modal Analysis
- FEA Finite Element Analysis
- FRF Frequency Response Function
- LHS Latin Hyper-cube Sampling
- MLS Moving Least Squares
- NSIR Normalized Scaled Incircle Radius
- PKM Parallel Kinematic Mechanism
- RBF Radial Basis Function
- ROI Region Of Interest
- ROM Reduced Order Modelling
- SQP Sequential Quadratic Programming
Chapter 1
Introduction

1 Introduction

Parallel kinematic mechanisms (PKM) are mechanisms comprised of one or more closed loop chains. PKMs have been commonly used in innovative machine design for industrial manufacturing, patient rehabilitation, flight simulators, earthquake simulators, and robotic surgery due to their superior stiffness, accuracy and agility. However, PKMs have two inherent disadvantages: (i) limited workspace due to the mobility constraints imposed by the other legs in the mechanism, and (ii) variable, configuration-dependent stiffness due to the changing geometry of the mechanism, which can make motion control a difficult task [1]. In response to above drawbacks, numerous researchers have proposed the use of redundant PKMs to improve mechanism performance.

1.1 Redundant PKMs

Redundancy is the condition when the number of active joints in a PKM is higher than the number of degrees of freedom (DOF) required to fulfill a task. Redundancy is typically introduced by adding one or more active joints to the kinematic chains, by adding one or more extra chains to the non-redundant PKM, or by adding actuators to the passive joints.

Redundancy can be classified in two types depending on the effect the redundant active joints have on the mobility of the mechanism, i.e. the degrees of freedom of the mechanism’s end effector or platform. When the extra active-joints result in increased mobility of the mechanism, i.e. the mobility is greater than the required degrees of freedom, the mechanism is said to be “kinematically redundant”. When the extra-active joints do not result in increased mobility of the mechanism, and therefore the number of active joints is greater than the mobility of the mechanism, the mechanism is said to be “redundantly actuated” [2]-[3].

The design of redundant PKMs generally starts with the selection of the type of structure to be used. Due to the number of factors to be considered, some of them not even of a mechanical nature, like cost, a scheme to select a unique mechanism is not available and typically redundant PKMs are originated from non-redundant PKMs that are modified to improve some the non-redundant
mechanism performance. Some of the schemes to develop non-redundant PKMs focus on specific applications, such as the SCARA PKMs or also known as Schoenflies motion generators by Kong et al. [4], or that focus on a particular issue of the mechanism selection such as determining the number joints and links in the mechanism, also known as number synthesis [5]. Examples of PKMs including redundancy are the redundant Eclipse [6] and redundant Stewart-Gough platforms [7]-[8].

Once the structure of the PKM is chosen the next step is to select the geometric parameters of the mechanism; such as link length, platform dimension, and joint location. The parameters are selected such that they optimize a given criterion. This criterion is typically a kinematic performance specification or a dynamic performance specification. This performance specification becomes the objective function, while the geometric parameters become the decision variables of an optimization problem. Examples of kinematic performance specifications include workspace size, dexterity, and singularities [9]-[10]. On the other hand, dynamic performance specifications often include stiffness, actuator torque and payload capability [11]-[13].

Most applications, however, require more than one criterion to be optimized at the same time. Researchers have come up with different methods of optimizing multiple objectives during mechanism design [9]. While the objectives usually vary according to the application, some of the prominent methods use objective functions comprised of weighted specifications, ratios, and other relevant metrics as well as hierarchical objective classification [9], [14]. Similarly, search engines varying in complexity and efficiency have been proposed including discrete and continuous based search engines.

1.2 Benefits of Kinematic Redundancy

The introduction of redundancy can bring numerous benefits to the kinematic and dynamic performance of a PKM depending on their application. The most common benefits of redundancy reported in the literature, however, can be grouped in the following categories: increased reachable and dexterous workspace, elimination or reduction of singularities and increased dexterity, increased stiffness and accuracy, and reduced joint torques which can be used to increase load capacity and platform velocity and acceleration. The following section explores how researchers have used kinematic redundancy to improve the mechanical performance of PKMs in each of the aforementioned areas and provides some notable examples for each case.
1.2.1 Reachable and Dexterous Workspace

The reachable workspace of a mechanism is typically defined as the region in space where all the points are reachable by the end-effector or platform of the mechanism in an arbitrary orientation. In contrast, dexterous workspace is defined as the region in space where all the points are reachable by the end-effector in all possible orientations [15]. Both, the reachable and dexterous workspace of parallel mechanisms are small compared to those of serial mechanisms. The reason is mainly that the workspace of parallel mechanisms is limited by the kinematic constraints imposed by each chain of the mechanism.

Kinematic redundancy can increase the size of the reachable and dexterous workspace of parallel mechanisms. Adding new active joint-link pairs in one or more existing chain of the mechanism increases degrees of freedom of the chain. The extra degrees of freedom increase the range of motion of the chain and reduce the kinematic constraints on the overall mechanism. The reduced kinematic constraints result in mechanisms with larger workspaces. Kinematic redundancy, however, has to be introduced carefully as it has been shown that sometimes it can actually hinder the size of the workspace particularly when extra legs are added to the mechanism as shown in [16].

One of the earliest mechanisms that took advantage of kinematic redundancy to increase the workspace was proposed in 1994 by Zanganeh and Angeles [17]-[18]. The mechanism was a 9-DOF spatial parallel manipulator. The base and the platform of the mechanism were linked by six legs, three external and three internal legs. The three external legs had one active prismatic joint each while the three internal legs had an extra active prismatic joint each for a total of two active prismatic joints on each internal leg.

Significant research on kinematic redundancy over the recent years has resulted in several designs of kinematically redundant parallel mechanisms. Much of this work has taken place on planar manipulators. Some of the mechanisms studied include the 3-RRR, 3-RPR, and 3-PRR planar manipulators, where the underline denotes an active joint. Ebrahimi et al. first proposed adding one prismatic joint to each chain of the planar 3-RRR manipulator, [15], [19]. The redundant 3-PRRR showed to have both, larger reachable and dexterous workspace. Kotlarski et al. on the
other hand, proposed to add a prismatic joint to only one of the legs of the 3-RRR manipulator [20]. The results show a significant increase on the size of the workspace by adding an extra joint to only one leg. Similar results were obtained by Kotlarski et al. by adding one extra prismatic joint to one leg of the 3-RPR manipulator [20]. Ebrahimi et al. also studied a similar approach on the 3-PRR manipulator where they added an active revolute joint to all three legs of the manipulator to obtain a 6-DOF 3-PRRR redundant manipulator. The results showed that the dexterous workspace was particularly improved on the redundant manipulator when compared to its non-redundant counterpart. Gallant et al. expanded the dexterous workspace analysis to redundant manipulators of the n-PRRR [16], n-RRRR and n-RRPR [21] families using a technique based on the Gauss divergence theorem.

Examples of improved workspace through kinematic redundancy have been proposed beyond the planar manipulators. Zarkandi et al. improved the workspace of the 2-DOF PRRRP manipulator by adding two actuated revolute joints to the mechanism [22]. The resulting PRRRRP manipulator shows a substantially larger workspace than its non-redundant predecessor. A 4-DOF spatial manipulator with a 3-DOF end effector was proposed in [23]-[24]. The proposed mechanism resembles a delta robot with one extra revolute joint added to one of the legs. A different variation of the delta robot for increased workspace was proposed in [25]-[27]. Hess-Coelho introduced redundancy to a spherical wrist manipulator in [28]. The manipulator shows an increase in its workspace size, particularly, the roll angle range was significantly increased. A 7th redundant DOF was included in the spatial manipulator proposed by Bai et al. in order to increase the size of the manipulator’s workspace [29]. They tried two configurations of the manipulator and show that in both configurations the redundant DOF enlarges the workspace. Finally, Alagheband et al. recently proposed a redundant version of the pentapod in [30]. The proposed redundant pentapod showed a large workspace and it was particularly able to reach large tilting angles.

1.2.2 Singularities and Dexterity

Parallel mechanisms exhibit new types of singularities that do not occur in serial mechanisms due to the kinematic constraints imposed by the multiple chains in the mechanism. The first type of singularities also found in serial mechanisms is inverse kinematic singularities. Inverse kinematic singularities occur when the platform of the parallel mechanism loses one or more degrees of
freedom. The second type of singularities are known as *direct kinematic singularities*. This type of singularities occur when the platform “gains” one or more degrees of freedom [31].

Singularities affect the ability to control the manipulator, and in turn its ability to manipulate objects, therefore singularities are often related to the dexterity of a manipulator. At inverse kinematic singular configurations the manipulator is not able to move the platform in one or more directions. On the other hand, at direct kinematic singular configurations the platform can move in one or more directions even when the joints aren’t moving. The latter can have a serious effect on the manipulability of the mechanism as the manipulator is not able to resist torque or forces in such directions, and thus, they have become the main focus of research in parallel mechanism singularity elimination.

Kinematic redundancy can be used in parallel manipulators to avoid, reduce or eliminate singular configurations. Kinematic redundancy can help eliminate inverse kinematic singularities by introducing extra degrees of freedom to the mechanism to make up for the “lost” degrees of freedom by the platform at inverse kinematic singular configurations. Kinematic redundancy may also help eliminate direct kinematic singular configurations since its introduction changes the form of the Jacobian of the mechanism [32]. Moreover, if singular configurations remain in the workspace of the mechanism, kinematic redundancy provides the ability to “avoid” singular configurations by selecting alternate configurations to carry on a given task.

Several examples of singularity elimination and/or reduction exist in the literature. Wang and Gosselin introduced one redundant degree of freedom to the 3-RPR planar manipulator by adding one revolute joint to one of the legs [33]. The resulting 4-DOF planar manipulator showed a reduction of one dimension in the singularity loci. Moreover, the remaining singularities could be avoided using trajectory planning. Ebrahimi et al. added an extra prismatic joint to all three legs of the 3-RRR planar manipulator. The resulting 6-DOF 3-PRRR planar manipulator was able to avoid all the singularities in the workspace [15], [19].

One of the earliest studies on the advantages of kinematic redundancy to reduce singularities in spatial manipulators was proposed by Zanganeh and Angeles [17]-[18]. The authors introduced one extra prismatic joint to the three internal legs of the six legged mechanism to help it cope with singularities. More recently, similar to their work on the 3-RPR manipulator, Wang and Gosselin added an extra revolute joint to one of the legs of the 3-UPS spherical manipulator and the 6-DOF
Stewart-Gough platform. Both redundant mechanisms showed significant singularity reduction compared to their non-redundant counterparts [33].

1.2.3 Joint Torques: Stiffness, Load Capability and Platform Motion

Joint torques on parallel mechanisms are related to most indices of good dynamic performance. The limit of the amount of torque that an actuator can handle is the main limitation on dynamic properties such as stiffness, load capability and platform velocities and accelerations.

Stiffness of a mechanism is defined as the ability of the mechanism to resist deformation due to external forces [34]. In parallel mechanisms, stiffness is typically related to the displacement of the platform due to the applied external force. Thus, stiffness is directly related to the precision of the mechanism. Stiffness in parallel mechanism is composed of the structural stiffness of the mechanism and joint stiffness. The constraints imposed by the multiple legs of the parallel mechanism virtually eliminate bending and torque on the closed-loop structure, having only to deal with the limited effects of tension and compression resulting in high structural stiffness [35]-[36]. However, joint stiffness also contributes to the deflection of the platform [37]. The higher the external force the larger the joint torque required. Larger joint torques result in larger joint displacements, which in turn results in deviation of the platform position. Reducing the joint torques as a result increases the stiffness of the mechanism.

Other benefits of reduced joint torques include increased load or force capability [38], and increased platform motion capability [39]. The force capability of a mechanism is related to the maximum wrench the platform of the mechanism can apply to the external environment, while the platform motion capability is related to the maximum velocity and acceleration of the platform related to the maximum available joint torques.

Kinematic redundancy can help reducing joint torques significantly. The extra amount of actuators available allows for the joint torques to be better distributed so that the torque applied on each active joint may be reduced [40]. Similarly, the introduction of kinematic redundancy may also help increasing the structural stiffness of the mechanism, since structural stiffness is a configuration dependent property [41]. The stiffness of the manipulator may be improved by selecting the configuration that provides the best combined joint and structural stiffness from the redundant configurations introduced by kinematic redundancy.
Some of the research in the literature about the use kinematic redundancy to reduce joint torque in parallel mechanisms is listed below. Boudreau and Nokleby added an extra prismatic joint to the 3-RPR manipulator [40]. The resulting kinematically redundant 3-PRPR manipulator was able to reduce the joint torque required by optimizing its configuration. Similarly Fontes et al. applied kinematic redundancy to the 3-RRR planar manipulator by adding active prismatic joints to the legs [42]. The resulting PRRR+2-RRR, 2-PRRR+RRR, and 3-PRRR all showed reduced joint torques. Moreover, the results showed that the torque reduction increased with increasing redundancy. Bai et al. introduced kinematic redundancy to a single DOF mechanism and optimized the mechanism’s configuration to maximize its load capability [43].

1.2.4 Trajectory Planning

Trajectory planning in parallel mechanisms usually consists in translating a desired task to be carried on by the platform into the joint motions required for the platform to fulfill such task. Tasks can include object positioning, machining, etc. Trajectory planning usually starts by representing the desired task as a desired motion sequence of the moving platform. The desired motion of the moving platform then is broken down into smaller discrete motions. The smaller discrete motions of the platform are then translated into joint motions using the inverse kinematic model of the mechanism. The trajectory planning usually ends with a discrete sequence of joint motions that allow the mechanism to fulfill the given task.

Kinematic redundancy can introduce a new challenge to trajectory planning. The introduction of kinematic redundancy increases the degrees of freedom of the mechanism. The extra (redundant) degrees of freedom may bring several benefits to the parallel mechanism like the ability to avoid singularities, an enlarged workspace and increased stiffness. However, they also bring an infinite amount of joint-motion solutions to the inverse kinematic problem. In non-redundant parallel mechanisms the inverse kinematic model provides a single solution or a finite number of solutions for a desired platform motion. Thus, the selection of a joint-motion solution usually becomes a trivial problem. On the other hand, the selection of joint motions among an infinite set of possible solutions in kinematically redundant mechanisms is a challenging problem that needs to be solved in order to fully take advantage of kinematic redundancy.

Trajectory planning has been the focus of much research in parallel redundant mechanisms where the main challenge is to select the best joint-motions solution in a computationally efficient
manner. Finding the best solution can be described as a two stage problem: the first stage is to
determine what a “good” solution is, while the second stage consists of the search of the “best”
solution available. Thus, trajectory planning in kinematically redundant mechanisms is often
associated with an optimization problem. The optimization problem in trajectory planning consists
of an objective function representing the criterion to be optimized, a search algorithm and the
constraints imposed on the solutions.

Different kinematic and dynamic criteria have been proposed to be optimized, and with them
different performance indices have been proposed as the objective function. Indices based on the
Jacobian of the manipulator and its relationship to dexterity and singularities have been the focus
of most of the research. Harib et al. proposed the condition number of the Jacobian as the objective
function in order to maximize dexterity as the optimization criterion [44]. Ebrahimi et al. also
proposed the condition number of the Jacobian as the objective function to maximize the distance
to singular configurations [45]. Kotlarski et al. proposed the two-norm of the maximal
homogenized pose error based on a modified Jacobian to maximize both the distance to
singularities and manipulator accuracy [46]. Cha et al. and Kotlarski et al. proposed objective
functions that maximize the determinant of the Jacobian respect to the platform coordinates to
avoid singular configurations in [47]-[48].

Alternatively, Ebrahimi et al. also proposed a geometrical method known as the Normalized
Scaled Incircle Radius (NSIR) to measure closeness to singular configurations, which they claim
is more useful for comparison between non-redundant and redundant manipulators than Jacobian
based indices [45], [49]-[50]. In [49]-[50] however, the authors use an objective function based on
the average and the minimum of this index of the entire trajectory as opposed to a point to point
optimization. Zarkandi et al. also proposed a geometrical measure of singular configurations as
the objective function in [22].

Some researchers have also proposed using kinematic redundancy to optimize the forces involved
in the manipulator trajectory. Oen and Wang proposed an objective function based on the Jacobian
in order to maximize the force at the mechanisms tool tip [51]. Similarly Boudreau and Nokleby
proposed an objective function that given an applied force at the mechanism’s tooltip it would
minimize the joint torques on the mechanism [40]. Fontes et al. proposed the maximum absolute
driving torque as the objective function for trajectory planning [42]. Using a similar approach Jiang
et al. used kinematic redundancy to choose trajectories that optimize the joint torques in the mechanism in order to minimize the tracking error [52]. The joint torques are considered to be the main dynamic sources contributing the tracking error through the axial deformation of the links. He and Lu used the extended Jacobian of the manipulator as the base of an objective function to minimize the shaking forces in the mechanism [53].

Other objectives for the trajectory planning problem of redundant manipulators proposed by researchers include tracking error minimization and energy efficient. Chen et al. proposed optimizing the accuracy of the platform by choosing a trajectory that minimized the error transmission from the active joints to the platform [54]. Smirnov et al. proposed configuration optimization in order to improve energy efficiency [55]. The reduction in energy consumption is mainly tackled by avoiding the breaking of non-regenerative actuators like electric drives used in PKMs. Ozgoren used the potential energy and the kinematic energy as objective functions for optimization [56].

The second stage of the optimization problem is the search for the optimum solution. The search algorithm is usually selected from classical optimization algorithms. The search algorithm is selected taking into account the dimensionality of the search space, which is related to the number of redundant degrees of freedom of the mechanism, and the nature of the objective function to be used. Oen and Wang used an algorithm based on the approximate programming method which is applicable to non-linear problems [51]. Boudreau and Nokleby implemented their search using an algorithm based on the Sequential Quadratic Programming (SQP) method [40]. Carretero et al. also applied the SQP method, however they obtained better results using a Differential Evolution (DE) algorithm to search for the optimum values [50]. Li et al. used an approach based on the Genetic algorithm to find an optimal solution [57]. Ozgoren proposed a method that combines the analytical and numerical approach [56]. An analytical approach is first used to reduce the complexity of the problem and then a numerical approach is used to solve the less complex problem. Niemann et al. presented a strategy to reduce the amount of optimization points necessary to solve the optimization problem [58]. The strategy could be combined with an arbitrary numerical search algorithm. Alternatively to traditional optimization algorithms Harib et al. proposed a knowledge based approach to make optimization more efficient than traditional methods [44].
1.3 Dynamic Stiffness Optimization

Mechanism stiffness optimization has become the focus of several studies due to its industrial relevance, particularly for those applications requiring high precision under potentially large external forces [59]-[62]. Large external forces can adversely affect accuracy due to the compliance of the mechanism. Compliance errors are hard to correct by most robot control schemes since the compliance of a PKM cannot be observed by its internal sensors, creating a need to maximize the mechanism stiffness. Many of the external forces that PKMs are subject to in industrial applications are dynamic in nature. For example, the deformation of a machine-tool structure due to constantly changing dynamic forces is known as dynamic stiffness. This deformation is a function of the amplitude and frequency of the dynamic excitation loads [63]-[67]. Moreover, the frequency of these forces is usually stochastic, creating the need to maximize stiffness not only at a specific frequency but over a frequency range. A good example of such forces is cutting forces present during the milling process in high precision machining [68]-[69].

Although redundant PKMs allow users to choose configurations with preferred dynamic stiffness, the complexity of evaluating dynamic stiffness via analytical models is prohibitively expensive for practical use [62]. Most analytical models of dynamic stiffness comprise variations of the Dynamic Stiffness Matrix method, where individual modules are modelled first and, then, assembled together [70]-[71]. The models are based on the principle of Virtual Power [72]-[73].

Some of the factors that make dynamic modeling of stiffness a complex task are the need of accurate distributed parameter modeling of the structural components, as well as, incorporation of the dynamic effects of the joints [69], [74]. As an alternative to analytical methods, Finite-Element-Analysis (FEA) based approaches, employing a solid model of the PKM, including detailed geometrical connection features of the structural components, can also yield accurate and reliable results [75]-[76]. However, FEA-based approaches are, typically, computationally-intensive solutions, which require the use of a new solid model for every new configuration in order to calculate stiffness, making them unsuitable for optimization.

1.3.1 Emulators

In order to overcome model complexities, researchers in different fields have utilized emulators (also known in the literature as surrogate models and meta-models) as less computationally
expensive approximations. For example, in [77], a Single Value Decomposition based emulator is used to obtain a dynamically simpler model of ecological processes with reduced dimensionality. Some of the more commonly used emulators in the field of mechanism dynamics are Artificial Neural Networks (ANNs) and Kriging-based emulators [78]-[81].

ANNs have been used in several studies including approximation of the dynamic behavior of structural systems subject to changing load conditions [80], dynamic model approximation of a redundant manipulator [79], and inverse kinematic model and Q-value approximation in trajectory planning [78]. Despite successful approximation, the main disadvantages of ANN-based emulators lie in that they typically require large sets of training data, long trial-and-error training procedures, modifying their structure to include known knowledge of the function to be approximated may be a very complex process, and training of the network does not take into account the distance between estimation locations and training points.

Kriging-based emulators, on the other hand, take into account the distance between estimation locations and training points and have been used successfully in the estimation of dynamic parameters. In [81], a Kriging-based emulator is used to model the effect of joint clearance size and the input crank speed on the dynamic behavior of a classical slider-crank mechanism with a revolute clearance joint at the piston. However, Kriging-based emulators need to generate a mesh in order to calculate the estimates. Mesh generation is, typically, the most computationally expensive part of most mesh-based simulations, having a negative effect in the efficiency of the method.

Recently, the use of the Moving Least Squares (MLS) model has gained attention as an alternative method [82]-[83]. MLS-based emulators have been used successfully in different applications, such as flood-inundation modelling [84] and structure engineering design and optimization [85]. Some of the advantages of MLS include its ability to deal with highly irregularly spaced points, as well as the mesh-free nature of the method (i.e., there is no need to generate a regular grid of training points), and that it requires a relatively smaller training data set in order to produce accurate results [83]. Moreover, the method formulation allows simple ways of identifying modifications in order to include previous known knowledge of the function to be approximated.
1.4 Thesis Objectives and Contributions

Dynamic stiffness is a key requirement in high precision applications. Parallel kinematic mechanisms have shown to have superior stiffness characteristics. Moreover, the introduction of redundancy in PKMs provides an opportunity to further improve the mechanism stiffness behavior. However, the lack of computationally efficient dynamic stiffness models prevents researcher from fully taking advantage of redundancy to improve dynamic stiffness. The goal of this thesis is:

- To provide a computationally efficient method to predict dynamic stiffness in redundant parallel kinematic mechanisms.

In this thesis an emulator is proposed to fulfill this goal. The proposed emulator could then be used to optimize stiffness during trajectory planning. However, in order for the emulator to be a feasible solution some conditions need to be satisfied. First, the amount of training data required needs to be minimized. Second, the accuracy of the prediction needs to be good enough to differentiate between different configurations. Finally, the emulator needs to be computationally efficient. These conditions result in the following sub-objectives:

- To reduce the dimensionality of the function to approximate in order to reduce the training data required.

- To find an approximation method that provides a good balance between training data required to produce accurate predictions and computational efficiency.

The contribution of this thesis is related to these two objectives above. The first contribution is the development of a methodology based on the structural modes of the PKM to reduce the dimensionality of the dynamic stiffness function. The methodology decomposes the high-dimensionality directional dynamic stiffness function into lower dimensionality functions. The decomposition of the stiffness function results in an exponentially smaller number of training data required to develop the emulator.

The second contribution is the development of a method based on Latin Hypercube Sampling and Moving Least Squares approximation to predict the dynamic stiffness function. The Moving Least Squares approximation provides a computationally efficient way to predict the lower dimensionality functions that comprise the dynamic stiffness functions while providing the
accuracy required to differentiate among redundant configurations. Methods of modifying the MLS formulation to introduce known characteristic of the objective function are also introduced in this thesis.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2 starts by presenting the problem of resolving a trajectory for a redundant PKM that maximizes stiffness as a two part problem: (i) the calculation of the dynamic stiffness of the PKM configuration at hand, and (ii) the search for the optimal configuration across a solution space potentially containing infinite number of feasible configurations, where the objective function is non-analytical in nature.

Chapter 3 describes the methodology used to address the first sub-problem, calculating the dynamic stiffness of the PKM configuration. The Chapters starts by proposing a methodology based on single frequency mode approach and approximation with ANNs. It highlights the issues that arise with this approach, namely, large training data sets and approximation errors; and proposes an alternate methodology to deal with them. The new methodology starts by tackling the high dimensionality nature of the problem by taking advantage of the mechanism’s structural modes to decompose the problem into lower dimensionality functions. It then continues to the training of the emulator, including data collection using FEA. Finally, it makes use of an MLS approximation based approach to predict the value of the lower dimensionality functions and calculate the directional dynamic stiffness prediction. The search for the optimal configuration is not within the scope of this paper, however, a comprehensive example is provided in Chapter 4 to illustrate the method.

Chapter 4 presents a case study considering a 6-DOF redundant PKM used in 5-DOF high-precision machining. The Chapter starts by describing the PKM and the sample path used in the study. It then shows how to build the emulator for this particular PKM and finally it shows the results provided by the emulator. The emulator results include a comparison of the proposed MLS-based emulator to FEA analysis, the ability of the emulator to predict which configuration is the stiffest configuration, how it compares to a scenario where no stiffness prediction is available, and
a robustness study that analyses the emulator performance under noise conditions and how the ability to select the optimal configuration is affected.

Chapter 5 provides a summary of the current situation, the solution proposed by the study and the results obtained by the study.
Chapter 2
Problem Statement

2 Problem Statement

Effective utilization of redundant PKMs requires the determination of optimal mechanism configurations that provide, in our case, the highest achievable dynamic stiffness. The solution of this optimization problem is subject to two inherent computationally intensive constraints: (i) the calculation of the directional dynamic stiffness of the PKM configuration at hand along a trajectory, and (ii) the search for the optimal configuration across a solution space potentially containing infinite number of feasible configurations. Herein, furthermore, the objective function for calculating dynamic stiffness, as suggested below, is non-analytical in nature. The focus of this thesis is, thus, to address the first constraint, namely, the computationally efficient estimation of the directional dynamic stiffness for a given PKM configuration.

This chapter describes the two main challenges involved in the problem of dynamic stiffness optimization: the prediction of the dynamic stiffness of a configuration and the search of stiffest configuration based on the prediction. Section 2.1 starts by defining dynamic stiffness and how it changes as a function of the mechanism configuration, force applied and trajectory to be followed. Section 2.2 continues to highlight the challenges that arise when trying to create a surrogate model particularly the high dimensionality problem and its relation to large training data sets. Section 2.3 continues to describe the problem of finding an optimum solution given an objective function. It shows the different factors to take into consideration including the nature of the function, whether is analytical or non-analytical, uni-mode or multi-mode, or differentiable or non-differentiable; and the nature of the search space, including its constraints and dimensionality.

2.1 Dynamic Stiffness

The stiffness of a PKM is, typically, defined as the ratio of the magnitude of the external force applied to the tool-platform center to the magnitude of the displacement of the tool-platform center. Directional stiffness is defined as the ratio of the magnitude of the external force applied to the tool-platform center to the magnitude of the projection of the displacement of the tool-platform center along the direction of the applied force. When the external force applied is oscillatory, the magnitude of the displacement projection of the tool-platform center would vary with the
frequency of the oscillatory force. Namely, the resulting ratio between force and displacement would depend on the frequency of the oscillations, referred to as dynamic stiffness [63]-[67]. In this thesis, a quasi-static motion of the PKM is assumed. Thus, herein, dynamic stiffness refers to frequency-dependent response of the displacement at the tool-platform, whereas the effect of the PKM’s actual motion on stiffness (such as inertial forces) is not considered. The frequency at which the dynamic stiffness is the least is considered as the worst-case stiffness of the configuration.

In this thesis, directional dynamic stiffness, $S$, refers to the worst-case dynamic stiffness along the direction of the applied force, formulated as:

$$S(q, u) = \frac{|F|}{|D_u(q, u, f_0)|} \quad S \in R,$$

where $q = [q_1, ..., q_n]^T$ is the joint-space vector that represents the PKM configuration and comprises the $n$ actuated joints of the PKM, $F$ is the force applied to the center of the tool-platform, $u$ is a unit vector along the direction of $F$, $D_u$ is the projection of the displacement of the tool-platform center along the direction $u$, and $f_0$ is the frequency at which the applied force produces the largest magnitude of $D_u$. $D_u$ is a function of the configuration $q$, the force direction $u$, and the frequency $f$; and, is referred to only as $D$, hereafter, for simplicity.

2.2 Prediction

Based on the discussion above, the first part of the dynamic stiffness optimization problem, namely the prediction part, can be stated as:

*Given a joint-space configuration, $q$, a given direction, $u$, and an oscillating frequency, $f$, determine the approximation function $U(q, u, f)$ for the underlying unknown non-linear, multivariate function $D(q, u, f)$, which maps the configuration space $q$, to the displacement of the tool-platform center projected along direction $u$, at frequency $f$. Subsequently, search for the largest value of $D$ over a frequency range.*

In order to solve a function-approximation problem, like the one described above, it would be necessary to obtain information about the function, also referred to as the target function. The target function is, thus, typically, sampled at a set of points spread throughout the domain of the
function, also referred to as training points. The more complex the target function is, the more training points would be required to generate a valid approximation. The complexity of the target function would depend on the degree of non-linearity of the function as well as the dimensionality of the function, where the latter is determined by the number of independent variables in the domain of the function. Namely, the accuracy of the approximation is constrained by the relationship between the target function complexity and the training points available.

As will be further discussed in Chapter 3 below, there exist different methods to obtain stiffness data of PKM configurations, the most popular ones being based on FEA-based simulations and/or Experimental Modal Analysis (EMA) [30], [86]. However, both of these methods are time consuming and require a significant amount of resources, thus, making the number of available training points an important constraint when approximating the stiffness of a PKM configuration. Moreover, it is not required to approximate $D$ over the entire frequency range, but to determine the maximum value of $D$ over the frequency range instead. The problem addressed in this paper can, thus, be re-formulated as:

Given a joint-space configuration, $\mathbf{q}$, and a given direction, $\mathbf{u}$, determine the approximation function $U(\mathbf{q}, \mathbf{u})$, for the underlying unknown non-linear, multivariate function $D(\mathbf{q}, \mathbf{u}, f_0)$, using an emulator, which would map the configuration space, $\mathbf{q}$, to the maximum displacement of the tool-platform center projected along the direction $\mathbf{u}$ at frequency $f_0$, using a finite set of training points $((\mathbf{q}, \mathbf{u}, f), D((\mathbf{q}, \mathbf{u}, f_0)))$.

### 2.3 Configuration Selection

Once there is a method available to predict the directional dynamic stiffness as defined above, the second part of the stiffness optimization problem becomes selecting the configuration that exhibits the maximum stiffness, i.e. the minimum displacement in the tool-platform path direction in a computationally efficient way. The configuration selection problem can be stated as:

Given a search space $\mathcal{Q}$, where $\mathcal{Q}$ represents the subset of redundant configurations in the joint-space of the mechanism that satisfy a given tool-platform coordinate, a tool-platform direction $\mathbf{u}$, a displacement approximation function, $U: \mathcal{Q} \rightarrow R^+$, determine the configuration $\mathbf{q}_o$, in the search space $\mathcal{Q}$, such that $U(\mathbf{q}_o, \mathbf{u}) \leq U(\mathbf{q}, \mathbf{u})$ for all $\mathbf{q}$ in $\mathcal{Q}$. 
The search problem, like the one described above, consists of two main aspects: the nature of the objective function and the nature of the search space. The objective function can be linear or non-linear. For linear functions, if an optimal solution exists, at least one of the extreme points in the feasible region would be an optimal solution, which reduces the problem to a finite computation since there is a finite number of extreme points. Optimizing non-linear functions, typically takes an iterative approach, where information about the function is used to decide the next evaluation, until the iterations converge. Typically non-linear optimization methods rely on function, gradient and Hessian evaluations to reach convergence. Typically, the more information available the less iterations that might be required to reach convergence, and hence the computational expense is reduced. However, evaluating all this information may have a high computational expense, particularly if explicit formulae isn’t available.

The second aspect in the optimization problem is the nature of the search space. The search space is defined by the dimensionality of the space and the constraints of the problem. The higher the dimensionality of the search space the more computationally expensive it becomes to find a solution, which might require the use of more sophisticated approaches to find the optimal solutions in a computationally efficient way [87]. For the optimization problem in redundant mechanisms the dimensionality of the search space is given by the number of redundant DOFs of the mechanism. The solutions to be found are also subject to the physical constraints of the mechanism. The physical constraints of the mechanism are given typically by the dynamic specifications of the actuators, which include typically maximum joint displacement and maximum joint force/torque which determine the maximum joint acceleration.

Taking the aspects mentioned above, the search problem, can thus be re-formulated as:

Given a search space $Q$, where $Q$ represents the multi-dimensional subset of redundant configurations in the joint-space of the mechanism that satisfy a given tool-platform coordinate, a tool-platform direction $u$, a non-linear displacement approximation function, $U: Q \rightarrow R^+$, determine the configuration $q_o$, in the search space $Q$, such that $U(q_o, u) \leq U(q, u)$ for all $q$ in $Q$, subject to the actuator displacement constraints, $y_{min} \leq q_x \leq y_{max}$ for all $q_x$, and actuator acceleration constraints, $a_{mn} \leq \ddot{q}_x \leq a_{max}$ for all $\ddot{q}_x$. 
where \( q_x \) is the \( x \)-th component of the joint-vector \( \mathbf{q} \), \( \ddot{q}_x \) is the \( x \)-th component of the joint acceleration vector \( \mathbf{\ddot{q}} \). \( y_{\text{min}} \) represents the joint displacement lower limit, \( y_{\text{max}} \) represents the joint displacement upper limit, \( a_{\text{min}} \) represents the joint acceleration lower limit, and \( a_{\text{max}} \) represents the joint acceleration upper limit. The solution the search problem stated above, however, falls outside the scope of this thesis.

It is important to note that as stated above, there exist two computational issues at hand: (1) developing an emulator via off-line calculations – namely, to obtain training data via FEA, which is indeed a time-consuming process, but is required only once in order to create the emulator; and, (2) choosing the optimal configuration at a given platform pose (position and orientation) – namely, calculation of the PKM stiffness at any configuration considered by the search engine of the optimization process. The computational savings come from not needing to invoke FEA repeatedly when searching for the optimal configuration at a given platform pose. Furthermore, although with an efficient optimization algorithm the selection of the best configuration may be on-line feasible when an emulator is available, the same would never be true if one has to rely on FEA.
Chapter 3
Methodology

3 Methodology

This Chapter outlines the methodology used to design an emulator to approximate the directional dynamic stiffness of a PKM configuration. Section 3.1 starts describing the initial methodology to tackle the problem based on a single frequency mode approach and approximation using artificial neural networks (ANN). The subsection describes the issues with the proposed methodology including the large amounts of training data required to produce accurate predictions and the introduction of errors by the single frequency mode approach. The following sections outline the new methodology proposed to produce more accurate approximations with a smaller training data set. Section 3.2 describes how to reduce the dimensionality of the function approximation problem by taking advantage of the compliance matrix at the structural modes of the mechanism, by breaking down the high-dimensional function approximation problem using the elements of the compliance matrix at the structural modes. The elements of the compliance matrix are functions of lower dimension than the original stiffness function. The reduced-dimensionality approximation results in a smaller amount of training points required. Section 3.3 discusses how to obtain a finite set of training points suitable for the reduced-dimensionality function approximations using FEA-based simulations. This section also provides a comparison to other methods available to obtain stiffness data, including closed-form analytical methods and EMA-based approaches. Section 3.4 describes how the proposed MLS-based method can be used to compute the individual reduced-dimensionality function-approximation of the compliance matrix elements using a smaller training data set. It continues to show how the compliance matrix element approximations are combined to build an emulator that predicts directional dynamic stiffness of a PKM configuration. Figure 1 below shows a high-level description of the methodology proposed.
3.1 Single Mode and Artificial Neural Networks

3.1.1 Single Frequency Mode

The problem in sub-section 2.2 can be described as a function approximation problem, where the stiffness function can be predicted by approximating the maximum displacement of the tool-platform center along a given direction over a frequency range:

\[ D_o = \max_f (D(q, u, f) = D(q, u, f_o)). \]  \(2\)

As shown in Eq. (2), the displacement response of the tool-platform center varies with the PKM configuration \( q \), the force direction \( u \), and the force frequency \( f \). In order for the emulator to be able to approximate the displacement function \( D \), for an arbitrary tool-platform path direction at an arbitrary frequency, the training data needs to capture the changes in the function with respect to all the changing variables, i.e. \( q, u, \) and \( f \). However, the interest herein is only in the largest value of the displacement over a frequency range and not in the displacement response at every frequency value; therefore, the emulator needs to approximate or ‘learn’ the patterns of the displacement only at the frequency where the maximum displacement occurs. Thus, the frequency \( f \), no longer needs to be a variable in the training, and instead all the training data can be obtained at frequency \( f_o \).
Now that the displacement is fixed at frequency \( f_0 \), it can be expressed using the compliance matrix of the mechanism, \( \mathbf{C} \), as a function of the PKM configuration. Since only the displacement at frequency \( f_0 \) is of interest, the compliance matrix at frequency \( f_0 \) is referred to as \( \mathbf{C}_o \):

\[
\mathbf{C}_o(q) = \begin{bmatrix}
C_{o,xx} & C_{o,xy} & C_{o,xz} \\
C_{o,yx} & C_{o,yy} & C_{o,yz} \\
C_{o,zx} & C_{o,zy} & C_{o,zz}
\end{bmatrix}.
\] (3)

The projection of the displacement, \( D_o \), along the direction, \( \mathbf{u} \), of the applied force, \( \mathbf{F} \), in terms of the compliance matrix \( \mathbf{C}_o \) is given as:

\[
D_o = \mathbf{u}^T \mathbf{C}_o(q) \mathbf{F}.
\] (4)

For simplicity, but without loss of generality, the magnitude of \( \mathbf{F} \) is set to 1 herein. Since the direction of the force is given by \( \mathbf{u}, \mathbf{F} = 1 \ast \mathbf{u} \), yielding:

\[
D_o = \mathbf{u}^T \mathbf{C}_o(q) \mathbf{u}.
\] (5)

Above, since the direction \( \mathbf{u} \) is given, the displacement \( D_o \) can be determined simply by calculating the compliance matrix \( \mathbf{C}_o \). The compliance matrix \( \mathbf{C}_o \) is only a function of \( q \), thus, to compute the directional dynamic stiffness, the training data only needs to represent changes in \( q \), however, training data has to be obtained for each of the compliance matrix elements \( C_{o,ij}; \mathbf{C}_o \in \mathbb{C}; i = x, y, z; j = x, y, z \).

In order to obtain the training data, a set of \( N \) PKM configurations, uniformly scattered throughout the joint-space domain of interest, is selected. For each of the \( N \) PKM configurations, a force is first applied to the tool-platform center along the \( x \)-axis and the maximum displacement response (i.e., amplitude and phase) along the \( x, y, \) and \( z \) axes is recorded, yielding all the \( C_{o,ix}; i = x, y, z \) elements of the compliance matrix. The procedure is repeated with the force being applied along the \( y \)-axis to obtain the \( C_{o,iy} \) elements, and once more with the force being applied along the \( z \)-axis to obtain the \( C_{o,iz} \) elements, respectively, providing a set of uniformly scattered training points for each \( C_{o,ij} \) element.
3.1.2 Artificial Neural Networks

Once a training set is obtained, the compliance matrix elements $C_{o,ij}(q)$, have to be approximated at any given random PKM configuration. The aforementioned problem can be described as a hyper-surface approximation problem [88]. Artificial neural networks have shown numerous advantages when approximating surfaces with high dimensionality, including accurate predictions [87], and have been used in the prediction of several dynamic aspects of manipulators including friction modeling [89], dynamic model approximation [79], and inverse kinematic model approximation [78].

A network of nested non-linear sigmoid functions with one hidden layer can approximate any continuous multivariate function arbitrarily well [90], thus, the ANN creates an approximation function of the $U(q)$ of the form:

$$U(q) = \sum_{z=1}^{o} c_z \sigma (\sum_{x=1}^{n} b_x q_x + b_0) \text{ and}$$

$$||C_{o,ij} - U||^2 = \sum_{\gamma=1}^{N} [C_{o,ij}(q_{\gamma}) - U(q_{\gamma})]^2 \rightarrow \min,$$  \hspace{1cm} (6)

where $\sigma$ represents a sigmoid transformation, $o$ is the number of nodes in the hidden layer, and the weights $c_z, b_x$, and $b_0$ are to be adjusted such that the error metric in Eq. (7) at the $q_{\gamma}, \gamma = 1, ..., N$ training configurations is minimized.

The neural-network parameters such as the number of nodes in the hidden layer, and the type of nodes to be used in each layer, represented by the function $\sigma$, have both an impact in determining the neural-network’s capabilities to approximate a given function [91]. The quality of the approximation is not only given by the capability of the network but also by the information available on the target function that is obtained through the training data samples available [92]. The capability of the network needs to go hand in hand with the information (i.e., number of samples) provided. The network needs to be powerful enough to fit the training data, however if it is too powerful, it will be able to fit the data in many different ways and it will be unlikely that the resulting approximation generalizes well, leading to overfitting [93].

Once the network parameters, i.e. the number and type of nodes in the hidden layer, are determined a learning algorithm needs to be chosen. Although not explicitly a part of the network structure, the learning algorithm is the one responsible for finding the set of weights $c_z, b_x$, and $b_0$ that
minimize the error according to Eq. (7). The learning algorithm needs to be able to deal with non-linear functions, multi-mode shapes, and converge to a solution in a feasible amount of time.

Once the $C_{o,ij}$ elements are computed, they are used to build the $C_o$ compliance matrix of the PKM, as shown in Eq. (3). Finally, the displacements, $D_o$, can be computed as:

$$D_o(q, u) = u^T \begin{bmatrix} C_{o,xx}(q) & C_{o,xy}(q) & C_{o,xz}(q) \\ C_{o,yx}(q) & C_{o,yy}(q) & C_{o,yz}(q) \\ C_{o,zx}(q) & C_{o,zy}(q) & C_{o,zz}(q) \end{bmatrix} u.$$  \hspace{1cm} (8)

3.1.3 Shortcomings

The single frequency mode and artificial neural network approach described above, while providing a computationally efficient approximation of the dynamic stiffness, is only able to produce accurate approximations with a very large set of training data, or on the other hand, it produces approximations of very limited accuracy when only a limited set of training data is available. The main reasons behind the shortcomings of the approach are described below.

- $C_{o,ik}(q)$ is complex: The dynamic stiffness of a PKM is a configuration-dependent property, and thus the frequency $f_o$ at which the maximum displacement occurs changes with the configuration $q$. As a result the function $C_{o,ij}(q)$ ends up being a complex function with high variability. The complexity of the function of $C_{o,ij}(q)$ requires a large number of training configurations in order to obtain a good approximation.

- Artificial Neural Network Parameters: The approximation structure provided by the ANN requires to find optimum parameters to maximize the fit through-out the entire approximation space. Thus, in order for the ANN to approximate the target function, it needs to have enough parameters to be powerful enough to approximate the function over the approximation space. The more parameters required to approximate the function the more training data that is required to fit those parameters. The requirement of large training data sets becomes evident especially when approximation functions with high dimensionality where the size of the required training data set becomes exponentially larger with the increasing dimensionality of the target function.
• **Artificial Neural Network Structure:** The approximation structure provided by the ANN consists on several nodes working in parallel. While this structure has shown to be very flexible and therefore ideal to fit functions without requiring a model, it relies mainly in the training samples to obtain information about the target function. Including known information about the target function in the approximation structure by any other means than training samples becomes a complex procedure, resulting in the need to obtain more training data in order to produce the approximation.

• **Single Frequency Mode:** The method relies in lowering the dimensionality of the displacement function by making use of the compliance matrix. It assumes that the elements of the compliance matrix $C_{o,ij}(q)$, for the same give configuration $q$ all happen around the same frequency $f_o$; and therefore the displacement $D_o$ can be approximated using matrix operations. However, the frequency $f_o$ at which the largest displacement occurs was found to vary significantly with the direction of the applied force and the direction of the measured displacement even when the configuration remained unchanged. And thus, the single frequency mode approach introduces extra inaccuracies in the model.

For applications where large amounts of training data is available the conditions described above might not be an issue, however, for this particular application obtaining training data is costly and hence it is a determinant factor for the feasibility of the emulator. In order to produce an emulator capable of producing accurate predictions with a smaller training data set all the above issues need to be addressed. The methodology proposed in the next section addresses those issues by using the structural modes of the mechanism and using MLS as a local approximation method. The structural modes of the PKM are used to breakdown the $C_{o,ij}$ functions into simpler functions that are easier to approximate and to maintain consistency in the frequency at which the matrix operations occur. The MLS approximation is provides a structure where the parameters need to be optimized only for a single point in the approximation space at a time, therefore requiring fewer parameters to provide accurate approximations, which results in a smaller training data set required. Its simpler structure also makes it easier to be modified in order to include extra knowledge about the target function that may be known for particular mechanisms, e.g. symmetrical PKMs, PKMs with curvilinear joints.
3.2 Structural Modes

As described in Chapter 2, the emulator to be designed should be capable to predict the directional dynamic stiffness of a given PKM configuration by approximating the maximum displacement of the tool-platform center along a given direction over a frequency range:

\[ D_o = \max_f (D(q, u, f)) = D(q, u, f_0). \] (2)

where the dimensionality of such a function, \( s \), is given by the sum of the dimensionality of all the independent input variables \( q \in \mathbb{R}^n, u \in \mathbb{R}^3 \), and \( f \in \mathbb{R} \). Thus, for a PKM comprising \( n \) independent active joints, the dimension of the displacement with the arguments, shown in Eq. (2), is \( s = n + 4 \). While the number of training points required to produce a valid approximation varies according to the method used to produce the approximation and the non-linearity of the target function, the number increases exponentially with the dimensionality of the problem [82], [87], [93]. Hence, if the emulator is to be practical, the dimensionality of the problem needs to be reduced as much as possible.

As shown in Eq. (2), the displacement response of the tool-platform center varies with the frequency, therefore, adding one extra dimension to the problem. However, the interest herein is only in those frequencies at which the displacement may be the largest and not in the displacement response at every frequency value. Thus, determining the frequencies at which the displacement is the largest would reduce the dimensionality of the problem by one dimension.

The displacement of the tool-platform center by an external force is the largest when the oscillating frequency of the external force resonates with the PKM’s structural mode frequencies. Thus, by identifying the PKM’s structural mode frequencies, we can associate the frequency variable to the structural mode frequencies. Since the frequency is no longer an independent variable, the dimensionality of the problem is reduced to \( s = n + 3 \).

Figure 2(a) below shows a 6-dof 3×PPRS PKM developed in our laboratory, where \( P, R, \) and \( S \) indicate prismatic, revolute, and spherical joints, respectively. Figure 2(b) shows an example Frequency Response Function (FRF) of this PKM’s tool-platform center’s displacement along the \( x \)-axis by an oscillating force along the \( x \)-axis. The two peaks in Fig. 2(b) appear at the PKM’s two structural modes. By predicting the displacement at the structural modes, we can predict the
maximum displacement over the entire frequency range. The specific 6-dof PKM used for the construction of this graph, and others in this Chapter, is further detailed in Chapter 4.

Figure 2. (a) A 6-dof 3×PPRS PKM, (b) Example FRF of the amplitude of the displacement.

Eq. (9) below shows the displacement formulation taking advantage of the structural modes of the PKM:

\[
D_0 = \max_f(D(q, u, f = f_1), \ldots, D(q, u, f = f_m)) = \max_f(D_1(q, u), \ldots, D_m(q, u)), \tag{9}
\]

where \(f_1, \ldots, f_m\) are the \(m\) structural mode frequencies of the PKM configuration. One can note that the frequency, \(f\), is no longer a variable, but it is fixed to a structural mode frequency. In order to simplify the notation, we will refer to the displacement at the \(m\) structural mode frequencies, \(f_1, \ldots, f_m\), as \(D_1, \ldots, D_m\), respectively.

Now that the displacements to be predicted are at fixed frequencies, \(f_1, \ldots, f_m\), they can be expressed using the compliance matrix of the mechanism, \(C\), which is a function of the PKM configuration and the oscillating frequency. Since we are only interested in the displacement at the structural mode frequencies, we refer to the compliance matrix at frequency \(f_1, \ldots, f_m\) as \(C_1, \ldots, C_m\), respectively:

\[
C_k(q) = \begin{bmatrix}
C_{k,xx} & C_{k,xy} & C_{k,xz} \\
C_{k,yx} & C_{k,yy} & C_{k,yz} \\
C_{k,zx} & C_{k,zy} & C_{k,zz}
\end{bmatrix}, \quad k = 1, \ldots, m, \tag{10}
\]
where $k$ is the structural mode number.

The projection of the displacement, $D_k$, $k = 1, ..., m$, along the direction, $u$, of the applied force, $F$, in terms of the compliance matrix $C_k$ is given as:

$$D_k = u^T C_k(q) F; \quad k = 1, ..., m. \quad (11)$$

For simplicity, but without loss of generality, the magnitude of $F$ is set to 1 herein. Since the direction of the force is given by $u, F = 1 \ast u$, yielding:

$$D_k = u^T C_k(q) u; \quad k = 1, ..., m. \quad (12)$$

Above, since the direction $u$ is given, the displacement $D_k$ can be determined simply by calculating the compliance matrix $C_k$.

The compliance matrix $C_k$ is only a function of $q$, and, therefore, the dimensionality of the original approximation problem is reduced from $s = n + 4$ to $s = n$. The reduced dimensionality implies less training points required for the approximation. However, the number of function approximations increases since every element of the $C_k$ matrices will have to be approximated. Namely, in order to compute the directional dynamic stiffness, we now need to approximate all the elements $C_{k,ij}; C_{k,ij} \in \mathbb{C}; k = 1, ..., m; i = x, y, z; j = x, y, z$. The net number of training points required is significantly reduced as the relationship between the number of training points and the dimensionality of the problem is exponential, thus, by far exceeding the number of extra training points required to compute the extra function approximations.

### 3.3 Training Data

As noted in Sub-Section 3.1 above, the construction of an emulator requires effective and sufficient training data to approximate each $C_{k,ij}$ element. It is proposed herein that, in order to obtain the training data, a set of $N$ PKM configurations, uniformly scattered throughout the joint-space domain of interest, should be selected. Dynamic stiffness data can, then, be obtained for each of the $N$ configurations using a FEA software package, e.g., ANSYS.

Alternatives to FEA-based methods for obtaining the dynamic stiffness of a PKM configuration, when a physical mechanism is available, include EMA techniques [86] or closed-form analytical models. The former produce accurate results, however, these approaches require the use of high-
precision experiments, but more importantly, they may be impractical due to high computational requirements.

Analytical models can provide time-efficient solutions. However, accurate prediction of dynamic stiffness using an analytical model would require distributed parameter modeling of the structural components, as well as, incorporation of the dynamic effects of kinematic joints and bolted connections, such as interfacial slip damping, friction, and backlash. The incorporation of joint dynamic effects requires accurate identification of the joint parameters, which is usually performed through experimental measurements.

In contrast to EMA techniques and use of analytical models, which may need to be validated through extensive experiments, the use of FEA-based calculations along with limited experimental measurements, when a physical mechanism is available, can provide both accurate and reliable results. For example, EMA techniques, through the use of an impact hammer, can be utilized to identify the equivalent damping ratio of the entire PKM structure at a given configuration. Although the use of FEA-based approaches in conjunction with EMA techniques are unlikely to provide time-efficient solutions, off-line calculations and measurements of the dynamic stiffness can be carried out to provide the training data required to build the emulator. A comparison of the various methods for determining dynamic stiffness is provided in Table 1.

In order to obtain accurate training data to predict $C_{k,ij}$ using FEA-based simulations, the PKM solid model must provide an accurate representation of the mechanism, where parameters, such as dimensions, structural material, joint stiffness, as well as other dynamic parameters (e.g., masses of the structural components, equivalent damping constants, etc.) are well known. Meshing parameters are also important in order to obtain consistent and accurate results in a computationally efficient way. Some of the most important parameters are the meshing method, and the mesh element size. Different mesh sizes may need to be selected for different components of the PKM depending on their geometrical complexity. Different mesh sizes may result in slightly different results, however, a good practice is to select the mesh size that results in the least variation between the result to the number of elements in the mesh.

For any given PKM configuration, a force is first applied to the tool-platform center along the $x$-axis and the displacement responses (i.e., amplitude and phase) along the $x$, $y$, and $z$ axes are
recorded at every structural mode, yielding all the $C_{k,ix}; k = 1, \ldots, m, \ i = x, y, z$ elements of the compliance matrix. The procedure is repeated with the force being applied along the $y$-axis to obtain the $C_{k,iy}$ elements, and once more with the force being applied along the $z$-axis to obtain the $C_{k,iz}$ elements, respectively, providing a set of uniformly scattered training points for each $C_{k,ij}$ element.

Table 1. Overview of methods to estimate dynamic stiffness.

<table>
<thead>
<tr>
<th>Method</th>
<th>Computational Cost / Experimental Effort</th>
<th>Accuracy</th>
<th>Integration to Optimization Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduced Order Modeling (ROM) [94]</td>
<td>- Requires vibration mode shapes obtained using impact hammer.</td>
<td>Medium</td>
<td>Difficult: the model has many dofs.</td>
</tr>
<tr>
<td></td>
<td>- Requires FEA only for validation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Requires transformation of generalized coordinates to local ones.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-intrusive and indirect ROM [76]</td>
<td>- Requires FEA results that actually run for certain settings.</td>
<td>Medium</td>
<td>Difficult: the model has many dofs.</td>
</tr>
<tr>
<td></td>
<td>- A series of static loading cases must be simulated.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transfer function modeling approach [95]</td>
<td>- Full alternative to FEA models.</td>
<td>High</td>
<td>Difficult: it can be used in control algorithms.</td>
</tr>
<tr>
<td></td>
<td>- No validation required.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3 below shows some example FRFs of the compliance (amplitude and phase) of a 3×PPRS PKM developed in our laboratory with two structural modes (i.e., $m = 2$). The figure shows the compliance FRFs of a PKM configuration for all force axis and displacement axis combinations, $\forall C_{ij}; i = x, y, z; j = x, y, z$, respectively: (a) force along the $x$-axis and displacement along the $x$-axis, $C_{xx}$, (b) force along the $x$-axis and displacement along the $y$-axis, $C_{yx} = C_{xy}$, (c) force along the $x$-axis and displacement along the $z$-axis, $C_{xz} = C_{zx}$, (d) force along the $y$-axis and displacement along the $y$-axis, $C_{yy}$, (e) force along the $y$-axis and displacement along the $z$-axis, $C_{zy} = C_{yz}$, and (f) force along the $z$-axis and displacement along the $z$-axis, $C_{zz}$. Due to the linear
properties of the mechanism, the compliance matrix is symmetric and, therefore, \( C_{ij} = C_{ji} \), hence, these elements are shown only once. The elements \( C_{k,ij} \) that make up the compliance matrices at the structural modes are displayed in each graph.

Figure 3. Example FRFs of the compliance of a PKM configuration for (a) \( C_{xx} \), (b) \( C_{yx} \), (c) \( C_{xy} \), (d) \( C_{yy} \), (e) \( C_{xy} = C_{yx} \), (f) \( C_{zz} \).
3.4 Moving-Least-Squares Approximation

Once a set of $N$ known training points $\mathbf{q}, C_{k,ij}(\mathbf{q})$, scattered throughout the domain of interest, is obtained, the objective is developing an effective emulator to predict $C_{k,ij}(\mathbf{q})$ at any given random PKM configuration. There exist several types of emulators that have been used in mechanism dynamics, including Moving Least Squares (MLS), Artificial Neural Networks (ANNs), Kriging-based emulators, and Radial Basis Function (RBF) [79], [81], [85], [96].

Among the several alternatives, MLS approximation offers several advantages in regards to dynamic stiffness approximation, primarily, yielding accurate predictions with a relatively small training data set. Other methods, like Artificial Neural Networks (ANNs), have shown to produce accurate predictions only when utilizing vast numbers of training data points [97]. This is especially true when the target function is quite complex. Due to the resource-intensive nature of the acquisition of dynamic stiffness data, the feasibility of the emulator construction depends on the required size of the training data set.

Another advantage of the MLS approximation method is that it does not require the training points to be aligned on a grid, or alternatively to generate a mesh with extra training points. Mesh generation is usually a time-consuming and computationally-expensive process, particularly when dealing with high-dimensional data, which in our context would imply PKMs with a high number of DOFs.

A commonly used mesh-free approximation method is Radial Basis Function (RBF) interpolation [82], [96]. However, while RBF interpolation does not require the generation of a mesh, it usually requires solving a large system of equations. The size of such system of equations would depend on the size of the training data set. Thus, complex functions requiring a large training data set for successful approximation would result in large systems of equations to be solved.

MLS approximation, on the other hand, requires a small system of equations to be solved. However, the system of equations needs to be solved every time a new prediction is to be made. The simplicity of the system of equations used allows for easy modification of the system to include extra information that might be known about the dynamic stiffness of the PKM, such as circular behavior of joints (e.g., revolute joints) and PKM symmetry, providing for more accurate predictions of the elements of the compliance matrix.
The elements of the dynamic compliance matrices to approximate are functions in the complex
domain, i.e., $C_{k,ij}(q) \in \mathbb{C}$. Therefore, in order to maintain the approximations in the real domain,
the $C_{k,ij}$ functions need to be broken into real and imaginary components, where both $Re\{C_{k,ij}\}$
and $Im\{C_{k,ij}\}$ are real functions:

$$C_{k,ij} = Re\{C_{k,ij}\} + Im\{C_{k,ij}\} \ast i; \quad Re\{C_{k,ij}\}, Im\{C_{k,ij}\} \in \mathbb{R}.$$ (13)

MLS can, then, be used to approximate both the real and imaginary components of $C_{k,ij}$.

Alternatively, in order to keep the approximation in the real domain, the complex $C_{k,ij}$ elements
could be divided, using Euler notation, into amplitude and phase components. MLS could, then,
be used to predict the amplitude and phase of $C_{k,ij}$. However, the abrupt changes in the phase of
the response would make it a harder function to approximate and, therefore, approximating the
real and imaginary components of the response is preferred over approximating the amplitude and
phase.

In order to approximate the real components $C_{k,ij}(q)$, the MLS method creates an approximation
function $U(q)$ of the form:

$$U(q) = \sum_{\nu=1}^{M} \alpha_{\nu}(q)p_{\nu}(q), \quad q \in \mathbb{R}^n,$$ and

$$||Re\{C_{k,ij}\} - U||_2^2 = \sum_{\nu=1}^{N} [Re\{C_{k,ij}(q_{\gamma})\} - U(q_{\gamma})]^2 \omega(q_{\gamma}, q) \rightarrow \min.$$ (15)

where $M$ is the number of multivariate polynomial functions, $p_{\nu}$, and the coefficients $\alpha_{\nu}(q)$ are to
be determined such that $U(q)$ satisfies the condition in Eq. (15). Above, $q_{\gamma}, \gamma = 1, \ldots, N$ are the $N$
training configurations and $\omega(q_{\gamma}, q)$ is a weighting function that decreases with increasing
distance between $q_{\gamma}$ and $q$.

In order to compute $U(q)$, to approximate $Re\{C_{k,ij}\}$, the elements of the MLS model, such as the
degree, $d$, of the $n$-variate polynomials, $p(q)$, and the weight function, $\omega(q_{\gamma}, q)$ need to be chosen.
Polynomials of higher degree, $d$, would produce more accurate approximations, however, may
result in a larger and hence more complex systems of equations to be resolved in order to satisfy
Eq. (9), making the method computationally more expensive. If the degree of the polynomials is
chosen such that, together with dimension, $n$, of the joint-space vector $q$, the size of the equation
system, $M = \left( \frac{n + d}{d} \right)$, is small enough that it can be solved analytically, the computational cost could be significantly reduced. Low-order systems, typically up to $M = 4$, can be solved analytically by using matrix manipulation, or Cramer’s rule, in order to obtain explicit formulae for the MLS coefficients. Explicit formulae, typically, result in faster implementation, however, solving larger systems analytically quickly becomes a complex procedure [82].

The weight function, $\omega(q, q')$, also needs to be chosen. The weight function needs to meet two main conditions. It needs to be a positive function and it needs to decrease with increasing distance from the origin. These two conditions implicitly ensure that the nearness of fit is achieved by the MLS formulation [82]. A typical example of a weight function would be the Gaussian weight function.

The imaginary component of the compliance matrix elements, $Im(C_{k,ij})$, can be computed in a similar way as the one described above. Once, both real and imaginary components are computed, $C_{k,ij}$ is computed, as shown in Eq. (7). The $C_{k,ij}$ elements are used to build the $C_k$ compliance matrices at the $m$ structural modes of the PKM, as shown in Eq. (4). Finally, the displacements, $D_k$, can be computed as:

$$D_o(q, u) = \max \left( u^T \begin{bmatrix} C_{k,xx}(q) & C_{k,xy}(q) & C_{k,xz}(q) \\ C_{k,xy}(q) & C_{k,yy}(q) & C_{k,yz}(q) \\ C_{k,xz}(q) & C_{k,yz}(q) & C_{k,zz}(q) \end{bmatrix} u \right), k = 1, ..., m. \quad (16)$$
Chapter 4
Case Study

4 A CASE STUDY: AN EMULATOR FOR A 3×PPRS REDUNDANT PKM

In this Chapter, the development of an efficient and effective emulator for stiffness approximation is detailed for the University of Toronto 3×PPRS PKM. It is based on the generic methodology described in Chapter 3 above. First, the architecture of the PKM and the sample path chosen for the study are described in Sub-Section 4.1. Subsequently, the details of the emulator built are detailed in Sub-Section 4.2. Section 4.3 provides some numerical results obtained by the emulator for configuration optimization for the path the PKM is required to follow.

4.1 The PKM and FE Model

The implementation of the proposed methodology for the prediction of mechanism dynamic stiffness via an emulator is illustrated herein through an example 3×PPRS PKM, where P, R, and S indicate prismatic, revolute, and spherical joints, respectively. This specific 6-dof PKM, designed and built at the University of Toronto for 5-axis meso-milling applications, provides a relatively large tool workspace and high-stiffness, through the 1-dof kinematic redundancy – typical requirements for precision milling, [86].

Specifically, the PKM shown in Fig. 4 consists of a (fixed) base on which three identical kinematic chains are mounted. Each chain comprises two actuators: the first (actuated) prismatic joint moves along a curvilinear rail, and its angular position is denoted by \( \theta_1, l = 1, 2, 3 \); the second (actuated) prismatic joint, mounted on top of the first one, moves linearly in the radial direction, and its linear position is denoted by \( \delta_1 \); a (passive) revolute joint is mounted on top of the second prismatic joint, which connects a fixed-length link to the moving platform via a spherical joint. The joint-space vector that represents the PKM configuration is given by \( \mathbf{q} = [\theta_1, \theta_2, \theta_3, \delta_1, \delta_2, \delta_3]^T \). We express the position of the curvilinear joints, \( \theta_1 \), in degrees (°) and the position of the prismatic joints, \( \delta_1 \), in millimeters (mm). The 6-DOF PKM described above has a 5-DOF task-space requirement (comprising the 3-DOF position of the tool-platform center and its 2-DOF orientation) leaving one redundant DOF.
In order for the FEA model to provide accurate dynamic stiffness data of the PKM several aspects have to be carefully considered, such as, accurate dimensions of the PKM obtained via the solid model, accurate modeling of contact interfaces and appropriate meshing. The solid model of the PKM was developed in SolidWorks, Figure 4. The main dimensions in the PKM solid model, are the link length, the base radius and the platform radius. The link length is 150.0 mm and is defined as the distance between the center of the revolute joint to the center of the spherical joint in a PKM leg. The base radius is 150.0 mm and is defined as the radius of the circular rail. The platform radius is 28.6 mm and is defined as the radius of the circumference that contains the center points of the three spherical joints. The structural components of the PKM were modeled using structural steel.

The contact interfaces between the structural components of the PKM consisted of rolling interfaces and bolted interfaces. The rolling interfaces included the joint bearings in the revolute and spherical joints, the curvilinear guide and the linear guide in the prismatic joints. The rolling interfaces were modeled as no separation contacts in ANSYS. The bolted interfaces included the connections between the prismatic actuator stage and the housing of the revolute joints, the spherical joint housing to the links, and the spherical joint housing to the platform. The bolted interfaces were modeled as bonded contacts. The damping ratios of the contact surfaces were modeled using experimental results.
The mesh used in the FEA model was created using tetrahedron-shaped elements. The mesh size used in the overall structure was 3.5 mm, however parts of the structure with more complex geometries were model using smaller mesh sizes. The contact interfaces between the link and the revolute joint housing were modeled using a mesh size of 2.0 mm, while the contact interfaces between the spherical joint and the link, and the spherical joint and the platform were modeled using a mesh size of 0.8 mm. The mesh sizes were optimized using a convergence test, where the optimum mesh size is determined as the mesh size that minimizes the variation of the results with respect to the amount of elements in the mesh.

### 4.2 Sample Tool Platform Path and Training Configurations

A 5 DOF sample path was chosen, where the tool-platform travels along the segment of a hemisphere arch while keeping the orientation of the tool-platform perpendicular to the surface of the hemisphere. The goal of this study is to maximize the dynamic stiffness of the PKM along the given path (i.e., dynamic stiffness in the direction tangent to the path). 15 reference points, \( P_0, \ldots, P_{15} \), uniformly distributed along the path were chosen to maximize the dynamic directional stiffness of the PKM. Figure 5 shows the sample path. The blue circles represent the reference points on the path, while the red arrows show the direction tangent to the path along which the directional stiffness is to be calculated.

![Figure 5. Sample path of PKM tool-platform.](image)
In order to build the proposed emulator, the first step is to define a region-of-interest (ROI), while the second step is to choose a finite set of 5-dof PKM platform position and orientations – *training poses*, within the ROI. The ROI is, naturally, expected to contain the potential PKM tool-platform paths (e.g., Fig. 5). Thus, a hemispherical ROI was selected herein with 120 uniformly scattered poses on it to model the variability of the dynamic stiffness in the ROI, in our case, using Latin hypercube sampling (LHS) [98]. Although we refer to the 120 poses as *training poses*, the training is carried out in joint-space using PKM configurations obtained at these specific poses.

Fig. 6(a) shows the 3D view of the 120 training poses, as blue circles, scattered on ROI, where the tool-path reference points given in Figure 5 above are noted as red circles. Fig. 6(b) shows the top view of the distribution of the training poses and the reference points.

For each of the 120 training poses, three redundant configurations were randomly selected, resulting in a moderate-sized set of 360 PKM training configurations, \( q_\gamma, \gamma = 1, \ldots, 360 \). It is important to note that even if a training pose were to coincide with a reference point by chance, only three redundant configurations of such a pose are used as training configurations in order to approximate the stiffness of the infinite number of possible redundant configurations at the reference points. No such coincidences occurred with our data.
4.3 Example Emulator Design

Once the training configurations were selected, the FEA software package ANSYS was used to obtain the compliance matrix elements $C_{k,ij}$ at the two PKM’s structural modes for every training configuration. As a result, a set of 360 training points for $(q, C_{k,ij}(q)); k = 1,2; i = x, y, z; j = x, y, z,$ were used to generate the MLS approximation.

Two key parameters were considered: the degree of the polynomial functions, $d$, and the weight function, $\omega$. The degree of the polynomial functions was chosen as $d = 1$ (i.e., linear reproduction), so that together with the dimension of the PKM configuration space, $s = 6$, the dimension of the MLS approximation space is $M = 7$. Keeping the size $M$ of the linear system small is important to ensure the system can be solved analytically in order to keep the computations efficient. If the degree of the polynomials were to be increased, even to $d = 2$, the result would be an MLS approximation space of $M = 28$, resulting in a large system of equations that would be more computationally expensive to solve.

The weight function employed was a Gaussian weight function:

$$\omega(r) = e^{-\epsilon r^2},$$

(11)
where $r$ represents the distance between two points and $\epsilon$ is a shape parameter. Typically, the distance function $r$ used is the Euclidean distance, which for two PKM configurations would be given by:

$$r(q_y, q) = \left\| q_y - q \right\|_2 = \sqrt{\sum_{l=1}^{3} (\theta_{yl} - \theta_{l})^2 + \sum_{l=1}^{3} (\delta_{yl} - \delta_{l})^2}.$$  \hspace{1cm} (12)

This is an acceptable fit for PKMs with linear joints. However, for PKMs with joints with circular behavior, joints whose initial and final position are the same (e.g., revolute joints, curvilinear joints), the Euclidean distance could fail to properly capture the distance between two configurations. As an example, let us consider the PKM configurations $q_1 = [0, 120, 240, 25, 25, 25]$ and $q_2 = [1, 120, 240, 25, 25, 25]$, where the first joint has been moved one degree clockwise from its position in $q_1$. The Euclidean distance between these two configurations is $r(q_1, q_2) = 1$. Now, let us consider a third configuration $q_3 = [359, 120, 240, 25, 25, 25]$, where the first joint has been moved one degree counter-clockwise from its position in $q_1$. The Euclidean distance between these two configurations is $r(q_1, q_3) = 359$. Clearly, a suitable distance function would present $q_2$ and $q_3$ at an equal distance from $q_1$.

In order to tackle the circular behavior of the curvilinear joints, we, thus, used the modified Euclidean distance function $r(q_y, q)$:

$$r(q_y, q) = \left\| q_y - q \right\|_2 = \sqrt{\sum_{l=1}^{3} \left( r_\theta(\theta_{yl}, \theta_{l}) \right)^2 + \sum_{l=1}^{3} (\delta_{yl} - \delta_{l})^2}, \text{ and}$$ \hspace{1cm} (13)

$$r_\theta(\theta_{yl}, \theta_{l}) = \min(|\theta_{yl} - \theta_{l}|, \theta - |\theta_{yl} - \theta_{l}|).$$ \hspace{1cm} (14)

where $\theta$ represents the angular measurement at one full revolution of the joint (e.g., 360°). Modifying the original model to accommodate for known information about the particular PKM studied allows for more accurate approximations.

PKM symmetry, as in our case, may also occur when the legs that comprise the PKM are identical, a common feature in PKM designs. This symmetry implies that shifting the leg positions around would result in the same topology of the PKM, referred to as ‘symmetric configurations.’ Thus, the value of the compliance matrix elements, $C_{k,ij}$, of symmetric configurations would be the same.
PKM symmetry would allow the replacement of each of the training configurations, $q_{\gamma_i}$, by their symmetric configuration that is closest to the configuration $q$ being approximated. Closer training configurations would result in more accurate approximations. Appendix A details an example of symmetric configuration replacement in emulator training for more accurate approximation.

Once all the $C_{k,ij}$ elements are computed, the displacement predicted by the emulator would be given by:

$$D_o(q, u) = \max \left( u^T \begin{bmatrix} C_{1,xx}(q) & C_{1,xy}(q) & C_{1,xz}(q) \\ C_{1,yx}(q) & C_{1,yy}(q) & C_{1,zy}(q) \\ C_{1,zx}(q) & C_{1,zy}(q) & C_{1,zz}(q) \end{bmatrix} u, u^T \begin{bmatrix} C_{2,xx}(q) & C_{2,xy}(q) & C_{2,xz}(q) \\ C_{2,yx}(q) & C_{2,yy}(q) & C_{2,zy}(q) \\ C_{2,zx}(q) & C_{2,zy}(q) & C_{2,zz}(q) \end{bmatrix} u \right).$$

4.4 Simulation Results

Herein, first, we examine how accurate the emulator predictions are when compared to the (assumed) ground-truth FEA data. Our objective is to validate that the emulator is capable of yielding a valid approximation of the platform displacement that can be used to choose the optimal PKM configuration for a given task-space requirement. Subsequently, we show some example results in determining the optimal configurations for the 15 reference points along the sample path shown in Figure 5 above. These results are then compared to the case in which no stiffness prediction may be available. A robustness study is also included to show the impact of potential noise on optimal PKM configuration estimation via the proposed emulator.

4.4.1 Comparing Emulator and FEA Results

The redundancy of the PKM considered in this paper is expressed in terms of the platform roll angle, i.e., the rotation about the axis normal to the platform (0 to 360°), Fig. 4. Due to symmetry of the PKM, i.e., the three legs being identical and equidistant from each other, the cycle of the displacement in terms of the platform roll angle is 120° instead of 360°. Thus, at each and every one of the 15 reference points along the sample path, the emulator was used to predict platform (minimum) displacement only for 121 redundant configurations (i.e., one configuration at each platform roll angle, 0 to 120°), respectively.

For an in-depth analysis, Fig. 7 below shows the emulator predictions for one specific Reference Point, $P_{02}$, as well as FEA analysis carried at some of these configurations. Fig. 7(a) shows the first PKM structural mode, and Fig. 7(b) the second PKM structural mode. Fig. 7(c) shows the
final displacement emulator prediction, and Fig. 7(d) the final displacement emulator predictions (blue dot) vs FEA data (black dot) obtained at a sub-set of the 121 configurations, i.e., 29 configurations. In this and all other cases, the final displacement prediction error was found to be the highest around the peaks and valleys of the displacement response.

![Graphs showing displacement predictions](image)

**Figure 7.** Prediction of displacement of tool-platform data.

### 4.4.2 Determining Optimal PKM Configuration

The emulator was used to determine the stiffest configuration, i.e., the PKM configuration which yields the smallest displacement at the tool platform center along the path direction, for each of the 15 reference points along the sample path. The displacement values at the emulator-based optimal configurations were, then, compared to the displacement values at the ‘absolute’ FEA-based optimal configurations, Table 2. As one can notice, the emulator’s ‘guesses’ are quite acceptable.
Table 2. Displacement at optimum redundant configuration.

<table>
<thead>
<tr>
<th>Reference Point</th>
<th>FEA (µm)</th>
<th>Emulator (µm)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P01</td>
<td>78.47</td>
<td>78.47</td>
<td>0.0%</td>
</tr>
<tr>
<td>P02</td>
<td>78.86</td>
<td>80.68</td>
<td>2.3%</td>
</tr>
<tr>
<td>P03</td>
<td>77.75</td>
<td>81.14</td>
<td>4.4%</td>
</tr>
<tr>
<td>P04</td>
<td>78.97</td>
<td>78.97</td>
<td>0.0%</td>
</tr>
<tr>
<td>P05</td>
<td>77.36</td>
<td>77.36</td>
<td>0.0%</td>
</tr>
<tr>
<td>P06</td>
<td>77.90</td>
<td>77.90</td>
<td>0.0%</td>
</tr>
<tr>
<td>P07</td>
<td>77.34</td>
<td>77.34</td>
<td>0.0%</td>
</tr>
<tr>
<td>P08</td>
<td>76.07</td>
<td>76.07</td>
<td>0.0%</td>
</tr>
<tr>
<td>P09</td>
<td>76.27</td>
<td>76.27</td>
<td>0.0%</td>
</tr>
<tr>
<td>P10</td>
<td>76.49</td>
<td>77.38</td>
<td>1.2%</td>
</tr>
<tr>
<td>P11</td>
<td>76.41</td>
<td>76.54</td>
<td>0.2%</td>
</tr>
<tr>
<td>P12</td>
<td>76.09</td>
<td>76.16</td>
<td>0.1%</td>
</tr>
<tr>
<td>P13</td>
<td>74.09</td>
<td>74.09</td>
<td>0.0%</td>
</tr>
<tr>
<td>P14</td>
<td>73.90</td>
<td>73.90</td>
<td>0.0%</td>
</tr>
<tr>
<td>P15</td>
<td>73.01</td>
<td>73.01</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

Figure 8 shows a close-up of the emulator prediction vs FEA data for some of the PKM configurations around the ground-truth FEA-based optimal configuration for Reference Point P02. First, the emulator-based optimal PKM configuration is determined by the emulator’s lowest displacement yield (i.e., the lowest-value blue-dotted point) — it is marked on Figure 8 by the blue dashed vertical bar. Next, the corresponding ground-truth displacement value, at this emulator-based optimal configuration, is calculated via FEA (black-dotted point), whose value is shown through the blue dashed horizontal bar.

Similarly, the ‘absolute’ optimal PKM configuration, for Reference Point P02, is determined via FEA analysis (i.e., the lowest-value black-dotted point) — it is marked on Figure 8 by the black dashed vertical bar. The corresponding (ground-truth) displacement value, at this FEA-based optimal configuration, is shown through the black dashed horizontal bar. As one can notice, the two corresponding optimal configurations are minimally separated in joint space.
4.4.3 Comparison

The use of the proposed emulator was compared to a scenario in which no stiffness prediction may be feasible, where the PKM configuration would need to be chosen randomly. The error of a selected configuration, in such a case, could vary from 0, if the optimum configuration were to be selected by chance, all the way to the error produced by the configuration with the maximum displacement. Figure 9 shows such a case for Reference Point P02.
Figure 9. Optimal configuration selection by emulator in comparison to random selection.

Table 3. Statistical analysis of displacement data.

<table>
<thead>
<tr>
<th>Reference Point</th>
<th>Emulator Error (%)</th>
<th>Min Random Error (%)</th>
<th>Max Random Error (%)</th>
<th>Avg. Error Random (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P01</td>
<td>0.0%</td>
<td>0.0%</td>
<td>104.5%</td>
<td>52.3%</td>
</tr>
<tr>
<td>P02</td>
<td>2.3%</td>
<td>0.0%</td>
<td>103.8%</td>
<td>51.9%</td>
</tr>
<tr>
<td>P03</td>
<td>4.4%</td>
<td>0.0%</td>
<td>106.8%</td>
<td>53.4%</td>
</tr>
<tr>
<td>P04</td>
<td>0.0%</td>
<td>0.0%</td>
<td>103.6%</td>
<td>51.8%</td>
</tr>
<tr>
<td>P05</td>
<td>0.0%</td>
<td>0.0%</td>
<td>109.3%</td>
<td>54.7%</td>
</tr>
<tr>
<td>P06</td>
<td>0.0%</td>
<td>0.0%</td>
<td>107.6%</td>
<td>53.8%</td>
</tr>
<tr>
<td>P07</td>
<td>0.0%</td>
<td>0.0%</td>
<td>109.1%</td>
<td>54.5%</td>
</tr>
<tr>
<td>P08</td>
<td>0.0%</td>
<td>0.0%</td>
<td>112.4%</td>
<td>56.2%</td>
</tr>
<tr>
<td>P09</td>
<td>0.0%</td>
<td>0.0%</td>
<td>111.7%</td>
<td>55.8%</td>
</tr>
<tr>
<td>P10</td>
<td>1.2%</td>
<td>0.0%</td>
<td>111.1%</td>
<td>55.6%</td>
</tr>
<tr>
<td>P11</td>
<td>0.2%</td>
<td>0.0%</td>
<td>111.6%</td>
<td>55.8%</td>
</tr>
<tr>
<td>P12</td>
<td>0.1%</td>
<td>0.0%</td>
<td>112.7%</td>
<td>56.4%</td>
</tr>
<tr>
<td>P13</td>
<td>0.0%</td>
<td>0.0%</td>
<td>118.4%</td>
<td>59.2%</td>
</tr>
<tr>
<td>P14</td>
<td>0.0%</td>
<td>0.0%</td>
<td>118.3%</td>
<td>59.2%</td>
</tr>
<tr>
<td>P15</td>
<td>0.0%</td>
<td>0.0%</td>
<td>120.6%</td>
<td>60.3%</td>
</tr>
</tbody>
</table>
In order to illustrate this more clearly, Table 3 below shows the errors resulting from using the emulator in comparison to the minimum-maximum possible error range at each reference point along the path, as well as the average error in each range. Note that, the emulator’s ‘guesses’ are quite preferable to random guesses, which although could be as low as 0%, they could be as high as 120%, or at best an average of about not lower than 50%. Appendix B provides the resulting joint-space trajectories.

Calculation of a stiffness value via the emulator was less than 0.1 s on a PC with Intel i7-4770 at 3.40 GHz processor. A similar calculation via FEA (using the ANSYS software on the same computer) took approximately 1000 s.

### 4.4.4 Robustness

The proposed emulator was tested for robustness to noise that may be present in the training data. During the acquisition of training data, whether the data is obtained via FEA simulations or experimentally, the major sources of error would often include: errors in the displacement (accelerometer) readings, errors in the direction in which the displacement should be measured, errors in joint positioning, and errors in the force applied. Different systems could, naturally, have some of their error sources contributing more than others. However, in spite of various sources of error, all error contributions would be reflected in the displacement readings used to train the emulator.

Herein, Gaussian noise (with zero mean) was added to the training data used to build the emulator, and its impact on selecting the optimal PKM configuration analyzed. First, the noise level to be added to the training data was selected as 5%. This noise level represents the maximum error introduced in the training data, which for a Gaussian distribution (with zero mean) it is equivalent to three standard deviations. Subsequently, the ‘noisy’ emulator was built using the noisy data and asked to predict the displacements at the redundant configurations of each of the 15 reference points discussed above. At every reference point, the noisy emulator is used to determine the optimal configuration. The above procedure was repeated for noise levels of 25%, 50%, and 100%.

Figure 10 shows the emulator predictions at redundant configurations for Reference Point P02 for all four noise levels. The ideal case with no noise was shown in Figure 8. For each case, the emulator-based optimal PKM configuration is marked by the blue dashed vertical bar and its
corresponding ground-truth displacement value marked by the blue dashed horizontal bar. The ‘absolute’ optimal PKM configuration is marked by the black dashed vertical bar and its corresponding (ground-truth) displacement value is marked by the black dashed horizontal bar. The difference in the displacement values at the emulator-based optimal configuration and at the ‘absolute’ optimal configuration is defined herein as the estimation error.

Figure 10. Emulator predictions using noisy training data: (a) 5%, (b) 25%, (c) 50%, and (d) 100% noise levels, respectively.

The estimation error for each Reference Point (of the 15 considered) varies due to the random nature of the noise introduced. Thus, individual values of estimation errors provide little information on the effect of noise on the emulator’s ability to determine the optimal configuration. Instead, the maximum estimation error observed among all 15 reference points would provide a worst-case scenario metric for the noise level being introduced. Table 4 below shows the maximum observed estimation error for all 15 Reference Points when noise levels were increased from 5%
to 100%. The table also includes the maximum possible error when random selection is used instead of using our proposed methodology in selecting the optimal PKM configuration. For the example considered herein, as can be noted, the maximum random error is larger even than the case of 100% noise level, thus, showing the clear robustness of our method to noise.

Table 4. Robustness analysis for the proposed emulator – Displacement Errors.

<table>
<thead>
<tr>
<th>Noise Level (±3σ):</th>
<th>0%</th>
<th>5%</th>
<th>25%</th>
<th>50%</th>
<th>100%</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Estimation Error (%)</td>
<td>4%</td>
<td>4%</td>
<td>14%</td>
<td>70%</td>
<td>85%</td>
<td>120.6%</td>
</tr>
</tbody>
</table>
Chapter 5
Conclusions and Recommendations

5 Conclusions and Recommendations

5.1 Conclusions

PKM-based architectures have been widely used in various industrial applications, e.g., machining, due to their superior stiffness, agility, and accuracy, when compared to serial-mechanism-based machines. However, PKMs have two main restrictive properties: limited workspace and configuration-dependent stiffness. The latter can be turned into an advantage, if redundant DOFs are incorporated into the reconfigurable PKM architecture. Namely, platform-trajectory planning can be optimized by choosing the stiffest configurations amongst many possible.

Although promising, the stiffness optimization problem has two potential computational constraints: (i) the calculation/evaluation of the dynamic stiffness as the objective function, and (ii) the search for the global optimal solution. Herein, the focus is on the former problem, where the use of an emulator is advocated as a less computationally expensive approximation of the dynamic stiffness.

Despite of their computational efficiency, emulators approximating high-dimensional functions often require large amounts of training data to produce valid approximations. Training data is usually computationally expensive to obtain, thus, such emulators are often unfeasible in practice. Herein, the emulator proposed takes advantage of the PKM structural modes MLS approximation to provide accurate approximations with a smaller training data set.

In order to reduce the size of the training data set required the high dimensionality directional dynamic stiffness function needs to be reduced. To reduce the dimensionality of the target function a novel method is proposed using the structural modes of the mechanism. Using this approach the high dimensionality stiffness function is broken down into functions of lower dimensionality consisting of the individual elements of the compliance matrices at the different mechanism’s structural modes. The individual compliance matrix elements have a significant lower dimensionality than the stiffness function resulting in an exponential reduction of the amount of training data required for the approximation.
The individual elements of the compliance matrices are approximated using an MLS approximation based approach. The method provides an accurate approximation of the compliance components suitable for the reconstruction of the dynamic stiffness function. The resulting approximation is accurate enough to allow the discrimination of the stiffness capabilities among redundant configurations. Thus, it allows the selection of the stiffest redundant configuration in a computationally efficient way.

The proposed method is shown to yield approximations with errors between 0.01% and 6.35%, providing adequate means for optimization. Stiffness optimization via the emulator is shown to yield configurations within a 4.4% error from the optimal configurations, while the average error of random guesses is no lower than 50%.

5.2 Recommendations

The prediction of dynamic stiffness in redundant PKMs using emulators was motivated by the need to optimize the dynamic stiffness of the PKM along a desired path. Once information about the dynamic stiffness of a configuration is made available the next step is to use that information to generate trajectories that maximize the dynamic stiffness of the PKM along the desired path. The generation of the trajectory becomes an optimization problem where the stiffest configurations along the mechanism’s path need to be selected. Some of the issues that need to be looked out for when implementing the trajectory optimization are:

- **Search Algorithm**: The search algorithm needs to be able to handle some particular specifications of the problem. It needs to be able to handle non-analytical, multivariate and multi-mode functions.

- **On-line vs Off-line Trajectory Planning**: The optimization of the trajectory can be carried off-line using the emulator so that it not takes away from the cycle time of the control loop. However, on-line implementations potentially opens new possibilities like taking into consideration error or sensor feedback information when optimizing the configuration.

- **Continuous vs Discrete Optimization**: If the on-line approach is chosen then the computational expense of trajectory planning has to be included in the control loop. If the optimization is continuously taken place then it could potentially lead to control problems
if the control cycle is severely increased. Therefore, an alternative is to optimize the configuration at discrete points during the trajectory so off-load the controller from some of the computational burden.

- **Continuous Joint-Space Trajectory**: Due to the multimode-nature of the stiffness function multiple configurations might have close to the same stiffness values. It is important for the optimization algorithm to constraint the configuration selection to configurations that will result in continuous or smooth joint-space trajectories in order to avoid control issues.
References


S. Niemann, J. Kotlarski, T. Ortmaier, and C. Muller-Schloer, “Reducing the optimization problem for the efficient motion planning of kinematically redundant


Appendices

APPENDIX A: Symmetric Configurations in Emulator Training

This Appendix describes how to improve stiffness prediction for a 3×PPRS PKM known to be symmetric. We start, first, by explaining the symmetry in the 3×PPRS PKM, how symmetric configurations can provide training configurations that are closer to the configuration of interest and, finally, how having closer training configuration may result in more accurate predictions.

The three legs that comprise the 3×PPRS PKM are identical and symmetrically spaced on the platform. This symmetry implies that shifting the leg positions around, while producing different configurations, would still result in the same topology of the PKM. Thus, the value of the compliance matrix elements, $C_{k,ij}$, at these configurations would be the same. These configurations that result in the same PKM topology are referred to as ‘symmetric configurations.’ Symmetric configurations have the same value of $C_{k,ij}$ while being at different distances from a given configuration $q$.

Figure A1 below shows three symmetric configurations of the PKM, whose legs are labeled as 1, 2, and 3, respectively. All three symmetric configurations, (a) 1-2-3, (b) 2-3-1, and (c) 3-1-2, result in the same PKM topology and same tool-platform pose.

![Figure A1. Symmetric PKM configurations (a) $q_a$, (b) $q_b$, and (c) $q_c$.](image)

When the MLS-based emulator makes a prediction at a configuration $q$, it uses the information provided by the training configurations $q_y$. This information is in terms of the value of $C_{k,ij}(q_y)$ at $q_y$ and the weight $\omega(q_y)$ assigned to the training configuration, based on the distance $r(q_y, q)$.
between \( q_y \) and \( q \). The closer the training configuration is to where the prediction is computed, the more valuable the information it carries about the function, thus, configurations with a smaller distance \( r \) are assigned a greater weight \( \omega \).

In order to have more valuable information about the function, it would be preferable to have training configurations that are as close as possible to where the prediction is computed. Replacing training configurations with their symmetric configuration that is closest to the location of the prediction allows us to obtain closer training configurations while the information on \( C_{k,ij} \) remains the same.

As an example, let us first consider Configuration 1-2-3, and \( q_a = [265, 24, 143, 24, 27, 23] \) as a training configuration whose \( C_{k,ij}(q_a) \) is known. Next, let us consider a configuration \( q = [145, 268, 27, 25, 22, 28] \) as the configuration whose \( C_{k,ij}(q) \) we want to predict. The distance between \( q_a \) and \( q \) is given by \( r(q_a, q) = \sqrt{120^2 + 116^2 + 1^2 + 5^2 + 5^2} = 203 \). Now let us consider Configurations \( q_b = [143, 265, 24, 23, 24, 27] \) and \( q_c = [24, 143, 265, 27, 23, 24] \). Configurations \( q_b \) and \( q_c \) are the result of shifting the legs of the PKM, hence \( q_a, q_b, \) and \( q_c \) are symmetric and, therefore, \( C_{k,ij}(q_a) = C_{k,ij}(q_b) = C_{k,ij}(q_c) \). The distance to Configuration \( q \) on the other hand is given by \( r(q_b, q) = 6 \) and \( r(q_c, q) = 213 \). Replacing the original training configuration, \( q_a \), with its symmetric configuration, \( q_b \), results in a training point \((q_b, C_{k,ij}(q_b))\) that is closer to the prediction configuration, \( q \).

Figure A2 below shows the Configuration \( q \) on the left next to the symmetric configurations \((a) \) \( q_a \), \((b) \) \( q_b \), and \((c) \) \( q_c \) on the right. It can be seen clearly that Configuration \( q_b \) in Figure A2 \((b) \) is closer to \( q \) than Configurations \( q_a \) and \( q_c \) in Fig. A2 \((a) \) and Fig. A2 \((c) \), respectively.

The process described above is applied to every training configuration \( q_y, y = 1, \ldots, 360 \). For every time a new prediction is to be computed, the symmetric configurations of each training configuration are determined by shifting the legs of the PKM around, the distance of each of the symmetric configurations to the configuration \( q \) is determined, and finally the symmetric configuration with the shortest distance is selected as the training configuration \( q_y \). The new set of training configurations \( q_y \) is, then, used to build the emulator.
Figure A2. Side-by-side comparison of a PKM configuration to three symmetric configurations.
APPENDIX B: Joint-Space Trajectories

This Appendix shows the resulting joint-space trajectories after using the proposed emulator to select the optimal configuration at each reference point of the sample path, Figure B1.

Figure B1. Joint-Space trajectories: (a) \( \theta_1 \), (b) \( \theta_2 \), (c) \( \theta_3 \), (d) \( \delta_1 \), (e) \( \delta_2 \), and (f) \( \delta_3 \).