



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

Rodrigo Luís Pereira Barreto

**TYPE SYNTHESIS OF MECHANISMS USING MATROID THEORY WITH
APPLICATIONS TO ASSISTIVE TECHNOLOGY**

Florianópolis
2020

Rodrigo Luís Pereira Barreto

**TYPE SYNTHESIS OF MECHANISMS USING MATROID THEORY WITH
APPLICATIONS TO ASSISTIVE TECHNOLOGY**

Tese submetida ao Programa de Pós-Graduação
em Engenharia Mecânica da Universidade Federal
de Santa Catarina para a obtenção do título de doutor
em Engenharia Mecânica.

Supervisor:: Prof. Daniel Martins, Dr. Eng.

Co-supervisor:: Prof. Andrea Piga Carboni, Dr. Eng.

Florianópolis

2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Barreto, Rodrigo Luís Pereira
Type Synthesis of Mechanisms Using Matroid Theory with
Applications to Assistive Technology / Rodrigo Luís Pereira
Barreto ; orientador, Daniel Martins, coorientador, Andrea
Piga Carboni, 2020.
160 p.

Tese (doutorado) - Universidade Federal de Santa
Catarina, Centro Tecnológico, Programa de Pós-Graduação em
Engenharia Mecânica, Florianópolis, 2020.

Inclui referências.

1. Engenharia Mecânica. 2. Mecanismos. 3. Robótica. 4.
Matroide. 5. Síntese do Tipo. I. Martins, Daniel. II.
Carboni, Andrea Piga. III. Universidade Federal de Santa
Catarina. Programa de Pós-Graduação em Engenharia Mecânica.
IV. Título.

Rodrigo Luís Pereira Barreto

**TYPE SYNTHESIS OF MECHANISMS USING MATROID THEORY WITH
APPLICATIONS TO ASSISTIVE TECHNOLOGY**

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca
examinadora composta pelos seguintes membros:

Prof. Marco Carricato, Ph.D.
Alma Mater Studiorum Università di Bologna

Prof. Martín Alejo Pucheta, Dr.
Universidad Tecnológica Nacional Facultad Regional Córdoba

Prof. Luís Paulo Laus, Dr. Eng.
Universidade Tecnológica Federal do Paraná

Prof.(a) Henrique Simas, Dr. Eng.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi
julgado adequado para obtenção do título de doutor em Engenharia Mecânica.

Prof. Paulo de Tarso Rocha de Mendonça,
Dr. Eng.
Coordenador do Programa

Prof. Daniel Martins, Dr. Eng.
Supervisor:

Florianópolis, 25 de agosto de 2020.

This work is dedicated to my wife Gabriela and to my parents. Without their support, this work would never have been possible.

ACKNOWLEDGEMENTS

First, I would like to thank my family. My parents have guided me through all my life and are definitely my biggest inspirations. My wife Gabriela has been by my side helping me during the doctorate even when I am being stubborn, complaining, angry or irritated. She also normally leaves me alone if I am feeling anti-social and I need to be left alone even if for a whole day.

I thank my colleagues from the Black Cow Mechanisms and Robotics Development Group André Molgato, Elias Maletz and Fernando Morlin. I am also grateful for my friends from the Laboratory of Applied Robotics, because without those endless conversations about mechanisms, matroids, and sometimes football, I would not have understood well enough mechanisms and Davies' method in order to propose contributions: Alinne, Estevan, Fabíola, Guilherme, Gustavo, Joãozinho, Julio, Kato, Léo, Luan, Mateus, Marina, Mônica, Paulo, Thais, Thiago, Treze. I thank also friends from other places that helped me deeply: Dino, Douglas, Marcos, Nolli, Paulo and Rômulo.

I am very grateful for my supervisors Professor Daniel Martins and Professor Andrea Piga Carboni. I am grateful for every challenging and creative conversation about mechanisms, for every discussion about ongoing and possible projects, because the potential of what we can do together is one of the aspects I love most about academic life. Professor Daniel Martins was always challenging me while giving me the freedom to research and develop my personal projects, some of which have grown because of his help. Professor Andrea Piga Carboni was always open for discussion about matroids, even when I had no idea what I was talking about. I am thankful for both for the belief in my sometimes crazy ideas and the patience with my frequent stubbornness.

I am thankful for other professors who, maybe unknowingly, have made a huge impact in my life and my work: Lauro César Nicolazzi, Rodrigo de Souza Vieira, Henrique Simas and Roberto Simoni. I am also very grateful for Dr. Helge Wurdemann for hosting me in London and, due to my time there, completely changing my work.

I have a special thanks to the Programa de Pós-Graduação em Engenharia Mecânica of the Universidade Federal de Santa Catarina and to the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior for the financial support through the project Tecnologia Assistiva PGPTA 59/2014.

*Considerate la vostra semenza:
fatti non foste a viver come bruti,
ma per seguir virtute e canoscenza.*

*Consider your origin;
you were not born to live like brutes,
but to follow virtue and knowledge.
(Dante Alighieri)*

RESUMO

Selecionar os tipos de juntas é um dos maiores desafios na síntese de mecanismos. O foco da presente tese é enumerar e selecionar mecanismos auto-alinhantes listando as restrições de todas as combinações possíveis de juntas a partir de um mecanismo embrião. Esta tese apresenta um novo método de selecionar mecanismos auto-alinhantes enumerados usando teoria de helicoides, método de Davies e teoria de matroides. O método de Davies e teoria de helicoides são usados para modelar mecanismos em termos de liberdades e restrições. Matroides são usados para enumerar mecanismos autoalinhantes usando os modelos criados com o método de Davies como entrada. Contração e deleção de matroides são então usadas para selecionar conjuntos de mecanismos viáveis. Esta tese introduz uma notação para representar as operações de matroides usadas na síntese de novos mecanismos. A presente tese também exhibe o conceito de juntas virtuais de Reshetov, uma estrutura com seis restrições e sem liberdades, que pode ser usada com o método de Davies. Implementando as juntas virtuais de Reshetov em mecanismos existentes, um novo método de enumerar mecanismos auto-alinhantes é introduzido, permitindo projetistas criarem novas juntas para receber liberdades e tornar o mecanismo embrião auto-alinhante. Um estudo de caso é exibido, no qual a palma reconfigurável da mão antropomórfica KCL/TJU é avaliada. Dois métodos de projeto de mecanismos auto-alinhantes são usados e comparados. Finalmente, esta tese também apresenta um método de enumerar mecanismos auto-alinhantes no qual os mecanismos embriões são constituídos apenas por juntas virtuais de Reshetov, permitindo que os projetistas possam criar mecanismos com diferentes propriedades cinemáticas. Usando o método de projeto a partir de mecanismos formados por juntas virtuais de Reshetov e o método de seleção de mecanismos usando contração e deleção, dois estudos de caso foram apresentados: os mecanismos de apoios das pernas e encosto para as costas de uma cama hospitalar.

Palavras-chave: Teoria de Helicoides. Método de Davies. Mecanismos Auto-Alinhantes. Teoria de Matroides. Contração. Deleção.

RESUMO EXPANDIDO

Introdução

Selecionar os tipos de juntas é um dos maiores desafios na síntese de mecanismos. As escolhas do projetista deve considerar requisitos de projeto, espaço de trabalho do mecanismo, liberdades adicionais que uma união de juntas pode gerar, entre muitas outras. Para um mecanismo simples, a tarefa pode não ser muito complicada para um projetista experiente, porém um projetista inexperiente pode achar desafiador definir quais juntas e por quê. O desafio de definir as juntas é ainda mais complexo quando se considera mecanismos patenteados: o projetista deve buscar novos e diferentes mecanismos para que um dispositivo inovador seja um possível resultado do seu trabalho. Em uma linha de manufatura ou montagem, gerenciar as tolerâncias de fabricação é vital para toda a linha de produção. Cada componente do sistema, como elos e juntas, tem incertezas de fabricação. Durante a montagem, essas incertezas são somadas. Se a precisão da manufatura é adequada para os requisitos de funcionamento do sistema, não resultando em problemas. Contudo, se a precisão não é adequada, esses erros podem impedir o correto funcionamento do sistema, gerando esforços internos nas peças, diminuindo a confiabilidade do sistema podendo acarretar na impossibilidade da montagem do mecanismo. Mecanismos e robôs geralmente possuem tolerâncias de fabricação apertadas para garantir a remoção ou inclusão de mobilidades perigosas ou restrições. Usando metodologias de projeto de mecanismos auto-alinhantes, que tem como objetivo remover restrições redundantes, o projetista pode criar novos mecanismos que são mais fáceis de fabricar e montar. Usando teoria de helicoides, teoria de grafos e método de Davies, é possível criar tanto modelos cinemáticos quanto estáticos de mecanismos. Usando um mecanismo com restrições redundantes como entrada, usando as referidas teorias e o método de Davies é possível gerar novos mecanismos auto-alinhantes. Grafos são usados para relacionar os esforços existentes entre juntas, enquanto o método de Davies utiliza as leis de Kirchhoff para permitir calcular todos os esforços presentes no mecanismo. A partir do modelo da estática criado segundo o método de Davies para um mecanismo com restrições redundantes, matroides são usados para enumerar todas as combinações possíveis de mecanismos auto-alinhantes derivados do mecanismo original. Neste trabalho é proposto a utilização de operações de matroides chamadas contração e deleção para auxiliar na seleção de mecanismos enumerados com as técnicas mencionadas. Além de facilitar a seleção de bases, usando contração e deleção nos matroides, o tamanho das bases também diminui, diminuindo também o custo computacional necessário para enumerar as bases. Este trabalho também introduz uma notação para identificar cada passo tomado durante a seleção de mecanismos usando contração e deleção de matroides. Outra contribuição do presente trabalho é a introdução das juntas virtuais de Reshetov, uma junta formada apenas por forças e momentos e sem nenhuma liberdade. As juntas virtuais de Reshetov são usadas para introduzir restrições redundantes em mecanismos ou então gerar mecanismos sem liberdades, permitindo a utilização de grafos, método de Davies e matroides para gerar novos mecanismos de maneira sistemática.

Objetivos

O principal objetivo deste trabalho é desenvolver um método gerar e selecionar novos mecanismos durante a síntese do tipo. Os objetivos específicos desta tese são: desenvolver ferramentas teóricas que auxiliem na síntese de novos mecanismos; desenvolver

ferramentas teóricas que auxiliem na enumeração e seleção de mecanismos; desenvolver uma nova abordagem para a síntese de mecanismos auto-alinhantes. Com os métodos e ferramentas apresentadas nesta tese, objetiva-se auxiliar projetistas no desenvolvimento de novos mecanismos, facilitando a geração e seleção de mecanismos inovadores a partir de conjuntos de requisitos de projeto através de ferramentas matemáticas fáceis de utilizar em softwares abertos.

Metodologia

Este trabalho apresenta abordagens para síntese de mecanismos auto-alinhantes. Em todas as abordagens utiliza-se teoria de helicoides, teoria de grafos e método de Davies para criar o modelo estático de um mecanismo. Com o referido modelo é possível avaliar a quantidade de restrições redundantes existentes no mecanismo. Utilizando as juntas virtuais de Reshetov, é possível criar um mecanismo sem liberdades porém com restrições redundantes, ou então adicionar juntas a um mecanismo existente que já possui restrições redundantes. Com os modelos estáticos dos mecanismos, este trabalho apresenta duas abordagens para gerar novos mecanismos. A primeira abordagem resume-se a remover colunas da matriz que representa o modelo estático do mecanismo enquanto se verifica o posto da matriz. Ao retirar colunas da matriz, restrições estão sendo eliminadas do mecanismo. Se, ao retirar uma coluna, o posto do mecanismo não foi alterado, uma próxima coluna deve ser retirada. Este processo é realizado até que o mecanismo esteja livre de restrições redundantes. Outra abordagem utilizada nesta tese é utilizar matroides para enumerar todas as combinações possíveis de mecanismos auto-alinhantes a partir de um mecanismo com restrições redundantes. Todas as bases dos matroides gerados utilizando as matrizes do modelo estático são enumeradas. Cada base destes matroides correspondem a um conjunto de restrições que formam um mecanismo auto-alinhante. O número de bases dos matroides cresce consideravelmente com o aumento da complexidade dos mecanismos, justificando-se então a utilização de métodos para seleção de mecanismos enumerados utilizando matroides. Este trabalho introduz a utilização de contração e deleção de matroides para selecionar as bases. Com a contração, o projetista consegue escolher quais restrições devem estar presente no mecanismo auto-alinhante, enquanto usando deleção o projetista escolhe quais liberdades quer dar para o sistema. As abordagens discutidas são apresentadas com diversos exemplos para apresentar e discutir as contribuições desta tese.

Resultados e Discussão

Diversos estudos de caso são apresentados. Inicialmente, utilizam-se as juntas virtuais de Reshetov, o método de Davies e cadeias cinemáticas com mobilidade igual a zero (cadeias de Baranov) para gerar novas estruturas para mecanismos tipo *grippers*. Uma cadeia sem mobilidade com nove juntas e doze elos, onde as nove juntas são juntas virtuais de Reshetov, é utilizada para gerar um modelo estático de mecanismo com restrições redundantes. Discutindo alguns requisitos de projeto, mostra-se como quatro novos mecanismos tipo *grippers* podem ser gerados. A palma antropomórfica reconfigurável da KCL/TJU, um mecanismo com restrições redundantes, é modelada e avaliada seguindo duas abordagens diferentes. Na primeira abordagem utilizada, novos mecanismos auto-alinhantes são gerados e selecionados usando contração e deleção de matroides. Inicialmente foram enumerados 2066 mecanismos auto-alinhantes possíveis a partir da palma reconfigurável. Utilizando contração e deleção chegou-se

em um conjunto de 45 mecanismos viáveis, representando 2,2% do total. A segunda abordagem introduz três juntas virtuais de Reshetov no mecanismo da palma para permitir maior flexibilidade na escolha dos tipos de juntas. O matroide gerado apresentou 4242 bases diferentes, possibilitando 4242 novos mecanismos auto-alinhantes. Utilizando os requisitos de projeto para fazer as contrações e deleções desejadas, chegou-se a um conjunto de 33 novos mecanismos auto-alinhantes, representando 0,78% do total. Novos mecanismos para uma cama hospitalar foram desenvolvidos utilizando as técnicas discutidas. Primeiro, um mecanismo para a seção das pernas foi modelado utilizando uma cadeia com duas mobilidades e oito juntas virtuais de Reshetov. O matroide referente ao modelo deste mecanismo apresentou 25566 bases, porém utilizando as operações de contração e deleção foi possível atingir um conjunto de 22 novos mecanismos viáveis, representando 0,086% do montante inicial. O mecanismo da seção das costas de uma cama hospitalar também foi desenvolvido seguindo a mesma abordagem de mecanismo formado apenas por juntas virtuais de Reshetov, chegando em um matroide com 1344797 bases. Usando contração e deleção, o número foi reduzido para 189 bases, representando 0,014%. Porém, como 189 bases ainda é um número considerável, utilizou-se as matrizes de cobases binárias como um método de seleção adicional, selecionando 55 mecanismos viáveis.

Considerações Finais

As contribuições desta tese provaram-se úteis para geração de novos mecanismos autoalinhantes. As juntas virtuais de Reshetov ajudam na abstração dos mecanismos no método de Davies, podendo ser usada sem a necessidade de um mecanismo conhecido como entrada. A utilização das juntas virtuais de Reshetov garantiu uma grande flexibilidade no desenvolvimento de novos mecanismos, podendo criar mecanismos integralmente ou ser usada para auto-alinhar mecanismos existentes. As operações de contração e deleção são eficientes para a seleção de mecanismos. Primeiro, devido ao fato de diminuir a quantidade de elementos na base, as operações permitem que o custo computacional para enumerar as bases seja diminuído consideravelmente. Segundo, com a utilização de requisitos de projetos, as operações são facilmente aplicadas utilizando bibliotecas já disponíveis no software aberto *SageMath*.

Palavras-chave: Teoria de Helicoides. Método de Davies. Mecanismos Auto-Alinhantes. Teoria de Matroides. Contração. Deleção.

ABSTRACT

Selecting the joint types is one of the biggest challenges in mechanisms design. The focus of this thesis is to enumerate and select self-aligning mechanisms by enumerating the constraints of all possible combination of joints starting from a seed mechanism. Using screw theory, Davies' method and matroid theory, this thesis presents a new method to select self-aligning mechanisms enumerated by matroid theory. Davies' method and screw theory are used to model mechanisms in terms of freedoms and constraints. Matroids are used to enumerate self-aligning mechanisms using the model created with Davies' method as input. Contraction and deletion of matroids are then applied to select a set of feasible mechanisms. This thesis introduced a notation for representing matroid operations when designing new mechanisms. This thesis also presents the concept of Reshetov virtual joint, a structure with six constraints and without freedoms, which can be used in Davies' method. Implementing Reshetov virtual joints into existing mechanisms, a new method to enumerate self-aligning mechanisms was introduced enabling the designer to create new joints to receive freedoms making the overconstrained seed mechanism self-aligning. A case study was presented, in which the reconfigurable palm of the KCL/TJU anthropomorphic hand was evaluated. Two self-aligning mechanisms design methodologies were used and compared. Finally, this thesis also presents a method to enumerate self-aligning mechanisms starting from a seed mechanism constituted solely by Reshetov virtual joints, enabling the designer to create mechanisms with different kinematic properties. Using the mechanism design methodology of seed mechanisms constituted by Reshetov virtual joints and the selection method by contraction and deletion, two case studies were presented: the leg rest and the backrest mechanisms of a hospital bed.

Keywords: Screw Theory. Davies' Method. Self-Aligning Mechanisms. Matroid Theory. Contraction. Deletion.

LIST OF FIGURES

Figure 1 – Reshetov’s table of the overconstrained four-bar mechanism.	25
Figure 2 – Table with Legs Labelled.	27
Figure 3 – List with the Combinations possible for Legs for New Table.	27
Figure 4 – Unfeasible Legs Combinations.	28
Figure 5 – Unfeasible combinations on the list.	28
Figure 6 – New list with the combinations possible for legs for new table.	28
Figure 7 – Combinations of Legs with L3 Present.	29
Figure 8 – Remaining Combinations.	29
Figure 9 – Simplified Remaining Combinations.	29
Figure 10 – Combinations of Legs with L3 Removed.	30
Figure 11 – Remaining Combinations.	30
Figure 12 – Simplified Remaining Combinations.	30
Figure 13 – Path generation.	40
Figure 14 – Motion generation.	40
Figure 15 – Methodology proposed by Murai (2019).	42
Figure 16 – Example of overconstrained and self-aligning mechanisms.	43
Figure 17 – Example of four-bar overconstrained and self-aligning mechanisms.	44
Figure 18 – Example of four-bar overconstrained and self-aligning mechanisms.	44
Figure 19 – Reshetov’s table of the overconstrained four-bar mechanism.	45
Figure 20 – Reshetov’s table of the self-aligning four-bar mechanism.	46
Figure 21 – Flowchart of Carboni’s Method for Enumerating Self-Aligning Mechanisms.	47
Figure 22 – Four-bar mechanism.	53
Figure 23 – Four-bar mechanism	55
Figure 24 – Seed Baranov chain with $n = 5$ and $j = 6$	59
Figure 25 – Example of joints and the remaining constraints.	60
Figure 26 – Action Graph for the Seed Baranov Chain with $n = 5$ and $j = 6$	62
Figure 27 – Two different mechanisms generated using the proposed method.	63
Figure 28 – Mechanism with dangerous mobility.	63
Figure 29 – Seed Baranov chain with $n = 9$ and $j = 12$	65
Figure 30 – First potential scheme for robot grippers based on the nine-link Baranov chain.	66
Figure 31 – Second potential scheme for robot grippers based on the nine-link Baranov chain.	66
Figure 32 – Third potential scheme for robot grippers based on the nine-link Baranov chain.	67

Figure 33 – Fourth potential scheme for robot grippers based on the nine-link Baranov chain.	68
Figure 34 – Graph example G	70
Figure 35 – Dual graph G^*	71
Figure 36 – Graph example G	73
Figure 37 – Graph G_1 - Graph G after a contraction of edge f	74
Figure 38 – Graph G_2 - Graph G after a deletion of edge f	75
Figure 39 – Different approaches used in this work to model overconstrained mechanisms.	81
Figure 40 – Forces on the tabletop.	82
Figure 41 – Flowchart of the Filtering Process.	86
Figure 42 – Reshetov notation for joint types.	87
Figure 43 – Four-bars used in the example.	88
Figure 44 – Organization of the joints.	89
Figure 45 – Joint types organized according to the seed mechanism.	90
Figure 46 – First step - contraction of the constraint from joint a	91
Figure 47 – Representation after deletion and contractions.	92
Figure 48 – Joint types organized according to the seed mechanism.	92
Figure 49 – Four-bars used in the example.	93
Figure 50 – computer aided design (CAD) model of the anthropomorphic reconfigurable palm.	95
Figure 51 – Some prehensile actions of the anthropomorphic reconfigurable hand.	96
Figure 52 – Screw system of the Reconfigurable Palm	96
Figure 53 – Parameters of the reconfigurable palm	97
Figure 54 – Joint types of the seed mechanism.	100
Figure 55 – Contractions of the constraint from joints a and e	101
Figure 56 – Axial clearance applied to a revolute joint.	101
Figure 57 – Operation diagram for deletion of the constraint U and contractions of for Joint b	102
Figure 58 – Self-Aligning Mechanism by the Method I	103
Figure 59 – Position of the New Joints	105
Figure 60 – Joint types of the seed mechanism.	106
Figure 61 – Operation diagram for the contraction of the constraints from joints a , b , c , d and e	106
Figure 62 – Operation Diagram for deletion of the constraint W and contractions of remaining constraints for Joint f	107
Figure 63 – Self-aligning mechanism developed using method II.	108
Figure 64 – New Self-Aligned Reconfigurable Anthropomorphic Hand	109
Figure 65 – Seed mechanisms constituted by Reshetov virtual joints.	111

Figure 66 – Joint types of the seed mechanism composed by Reshetov virtual joints.	113
Figure 67 – Operation diagram for contraction of the constraints of joint a	113
Figure 68 – Operation diagram for the deletion of constraint T and contraction of the constraints U and V of joint b	114
Figure 69 – Examples of New Mechanisms Enumerated Using Davies' Method and Matroid Theory	115
Figure 70 – Kinematic Chain for the Leg Rest Support Mechanism.	117
Figure 71 – Structure of off-the-shelf commercial actuator for hospital bed applications	118
Figure 72 – Action graph of the seed mechanism for the leg mechanism.	119
Figure 73 – Joint types of the seed mechanism for the leg-rest mechanism composed only by Reshetov virtual joints.	119
Figure 74 – Operation diagram for the contraction of the constraints T , U and V of joints e and g	120
Figure 75 – Operation diagram for the deletion of constraint T of joint d	120
Figure 76 – Operation diagram for the contraction of the constraints U and V of joint d	121
Figure 77 – Operation diagram for the deletion of constraint T of joints a , b and h	121
Figure 78 – Leg Rest Mechanism Developed	122
Figure 79 – Kinematic Chain for the Backrest Support Mechanism.	123
Figure 80 – Backrest Mechanism Developed	126
Figure 81 – Graph with circuits C_1 and C_2	146
Figure 82 – Graph with circuit C_3	147
Figure 83 – Graph for the examples.	152
Figure 84 – Graph with edge e deleted.	153
Figure 85 – Graph with edge e contracted.	153

LIST OF TABLES

Table 1 – Comparison between the five legged table example with matroid and mechanisms.	32
Table 2 – Explanation of the notation used in this work.	50
Table 3 – Comparison of Notations.	51
Table 4 – Freedoms and Constraints Screws of Joints.	51
Table 5 – Examples of joints derived from a Reshetov virtual joint.	60
Table 6 – Examples of Couplings and the Constraints Correlations.	61
Table 7 – Comparison between linear and graphic matroids.	71
Table 8 – Comparison of Columns, Rank and Maximum Possible Bases for Different Matrices.	72
Table 9 – List of bases and cobases from matroid $\mathcal{M}[M]$	77
Table 10 – List of cobases that include element m_5 and their respective bases.	77
Table 11 – List of bases and cobases of matroid $\mathcal{M}[M \setminus m_5]$	77
Table 12 – List of bases and cobases from matroid $\mathcal{M}[M]$	78
Table 13 – List of bases and cobases of matroid that include element m_6 and their respective cobases.	79
Table 14 – List of bases and cobases of matroid $\mathcal{M}[M/m_6]$	79
Table 15 – Forces location on the table.	82
Table 16 – Explanation of the proposed notation.	89
Table 17 – Constraint sets symbols.	91
Table 18 – Summary of the steps in the filtering process taken in the Method I.	102
Table 19 – Network Action Matrices Comparison Method I.	104
Table 20 – Network action matrices comparison with additional joints.	106
Table 21 – Summary of the steps in the filtering process taken in the Method II.	107
Table 22 – Network Action Matrices Comparison Method II.	108
Table 23 – Summary of the Steps in the Filtering Process for the Example Mechanism.	115
Table 24 – Summary of the steps in the filtering process for the leg-rest mechanism.	122
Table 25 – Summary of the steps in the filtering process for the backrest mechanism.	124
Table 26 – Position for the joints for the Synthesis of the Gripper Mechanisms.	141
Table 27 – Position for the joints for the Synthesis of the Leg-rest Mechanisms.	142
Table 28 – Position for the joints for the Synthesis of the Backrest Mechanisms.	142
Table 29 – Summary of the contraction and deletion of linear matroids.	152
Table 30 – Summary of the contraction and deletion of graphical matroids.	153
Table 31 – Eliminated constraints of the mechanisms enumerated by method I.	155
Table 32 – Eliminated constraints of the mechanisms enumerated by method II.	156

Table 33 – Eliminated constraints of the example mechanism enumerated by the type synthesis method.	157
Table 34 – Eliminated constraints of the leg-rest mechanisms enumerated by the type synthesis method.	158
Table 35 – Eliminated constraints of the backrest mechanisms enumerated by the type synthesis method.	159

LIST OF ABBREVIATIONS AND ACRONYMS

BBBM	building blocks-based methodology
CAD	computer aided design
EBM	enumeration-based methodology
KCL	King's College London
rref	reduced row echelon form
SM	specialized methodology
TJU	Tianjin University

LIST OF SYMBOLS

M	Mobility of a mechanism
λ	Screw system order
j	Number of joints
n	Number of links
f_i	Degrees of freedom of the i kinematic pair
ν	Number of independent loops or circuits
C	Number of constraints
q	Number of redundant constraints
$[N]_{\mu,C}$	Binary cobases matrix
K_i	Criterion i of the Artmann method
$\$$	Screw
h	Pitch of the screw
τ	Linear velocity
ω	Angular velocity
$\m	Motion screw
r	Angular velocity in the x axis
s	Angular velocity in the y axis
t	Angular velocity in the z axis
u	Linear velocity in the x axis
v	Linear velocity in the y axis
w	Linear velocity in the z axis
S	Direction of the screw
S_0	Vector of the position of the screw
$\a	Action screw
R	Moment constraint in the x axis
S	Moment constraint in the y axis
T	Moment constraint in the z axis
U	Force constraint in the x axis
V	Force constraint in the y axis
W	Force constraint in the z axis
G_C	Coupling graph
$[M_D]$	Unit motion matrix
$[B_M]$	Circuit matrix
$[M_N]$	Network unit motion matrix
F	Gross degree of freedom of the system
$[B_i]$	Diagonal matrix from the i row of the circuit matrix
ϕ	Vector with unknown magnitudes of velocities
$[A_D]$	Unit action matrix

$[Q_A]$	Cutset matrix
$[A_N]$	Network unit action matrix
$[Q_i]$	Diagonal matrix from the i row of the cutset matrix
ψ	Vector with unknown magnitudes forces and moments imposed by couplings
E	Ground set of the matroid \mathcal{M}
\mathcal{I}	Collection of subsets of E
J	A generic subset of \mathcal{I}
I	A generic subset of \mathcal{I}
G	A generic graph
$[M]$	A generic matrix
\mathcal{J}	Collection of subsets of E_G
G^*	Dual graph
\mathcal{B}	The bases of a matroid \mathcal{M}
b	Number of desired elements in a combination
k	Number of elements in a combination
Y	A generic subset
X	A generic subset
T	A generic subset
α	Angles between the joints of the reconfigurable palm of the KCL/TJU hand
P_a	Point of the position of the joint a of the reconfigurable palm of the KCL/TJU hand
P_b	Point of the position of the joint b of the reconfigurable palm of the KCL/TJU hand
P_c	Point of the position of the joint c of the reconfigurable palm of the KCL/TJU hand
P_d	Point of the position of the joint d of the reconfigurable palm of the KCL/TJU hand
P_e	Point of the position of the joint e of the reconfigurable palm of the KCL/TJU hand
$[A_{N_{palm_I}}]$	Network unit action matrix of the palm mechanism modelled by the method I
$[Q_{A_{leg}}]$	Cutset matrix of the palm mechanism modelled by the method I
$[A_{D_{palm_I}}]$	Unit action matrix of the palm mechanism modelled by the method I
\mathbb{R}	Set of real numbers
$[A_{D_{palm_{II}}}]$	Unit action matrix of the palm mechanism modelled by the method II
$[A_{N_{palm_{II}}}]$	Network unit action matrix of the palm mechanism modelled by the method II
$[A_{D_{leg}}]$	Unit action matrix of the leg rest mechanism

$[Q_{A_{leg}}]$	Cutset matrix of the leg rest mechanism
$[A_{N_{leg}}]$	Network unit action matrix of the leg rest mechanism
$[A_{N_{back}}]$	Network unit action matrix of the backrest mechanism
$[A]$	A generic matrix
D	A $r \times (n - r)$ matrix over the ground set of $[A]$

CONTENTS

1	INTRODUCTION	23
1.1	SYNTHESIS OF SELF-ALIGNING MECHANISMS	24
1.1.1	Reshetov's Method	24
1.1.2	Carboni's Method	25
1.2	A FIVE LEGGED TABLE	26
1.3	THESIS OBJECTIVES	32
1.4	THESIS CONTRIBUTIONS	32
1.4.1	Davies' Method	32
1.4.2	Self-Aligning Mechanisms Synthesis	33
1.5	THESIS OUTLINE	33
2	MECHANISM DESIGN TECHNIQUES AND METHODOLOGIES . .	35
2.1	NUMBER SYNTHESIS	35
2.2	TYPE SYNTHESIS	36
2.2.1	Motion-Based Methods	37
2.2.2	Constraint-Based Methods	38
2.2.3	Other Methods	38
2.2.4	Considerations About Type Synthesis	39
2.3	DIMENSIONAL SYNTHESIS	39
2.4	MURAI'S MECHANISM DESIGN METHODOLOGY	41
2.5	SYNTHESIS OF SELF-ALIGNING MECHANISMS	42
3	INTRODUCTION TO SCREW THEORY AND DAVIES' METHOD . .	49
3.1	SCREW THEORY	49
3.2	LINEAR DEPENDENCE OF FREEDOMS AND CONSTRAINTS . . .	51
3.3	DAVIES' METHOD	52
3.3.1	Kinematic Model	52
3.3.2	Static Model	54
4	RESHETOV VIRTUAL JOINTS	58
4.1	NEW GRIPPER MECHANISMS	64
5	BRIEF REVIEW OF MATROID THEORY	69
5.1	COMPARISON OF GRAPHIC AND LINEAR MATROIDS	69
5.2	COMBINATORICS PROBLEM	71
5.3	CONTRACTION ($/$) AND DELETION (\backslash) OF MATROIDS	73
5.4	ANOTHER APPROACH INTO DELETION	76
5.5	ANOTHER APPROACH INTO CONTRACTION	78
6	CONTRACTION AND DELETION APPLIED TO DAVIES' METHOD	80
6.1	COMPARISON TO THE TABLE ANALOGY	82
6.2	APPLYING CONTRACTION TO CONSTRAINTS	83

6.3	APPLYING DELETION TO CONSTRAINTS	84
6.4	SUMMARY OF THE SELECTION PROCESS	85
6.5	PROPOSITION OF A NOTATION TO REPRESENT MATROID OPERATIONS	86
7	ENUMERATION OF SELF-ALIGNING MECHANISMS USING DAVIES' METHOD AND MATROID THEORY: CASE STUDY OF THE KCL/TJU HAND	94
7.1	SELF-ALIGNING APPROACHES	98
7.2	METHOD I: INCREASING THE FREEDOM OF SPECIFIC JOINTS	99
7.3	METHOD II: INCREASING THE NUMBER OF JOINTS	104
8	MECHANISM SYNTHESIS METHODOLOGY	110
9	DEVELOPMENT OF HOSPITAL BED MECHANISMS	117
9.1	LEG REST SUPPORT MECHANISM	117
9.2	BACKREST ELEVATION MECHANISM	122
10	CONCLUSION	127
10.1	SUGGESTIONS FOR FUTURE WORK	128
10.2	PUBLICATION LIST	129
	REFERENCES	131
	APPENDIX A – INPUTS USED IN DAVIES' METHOD	141
A.1	GRIPPER MECHANISMS	141
A.2	LEG-REST MECHANISM	141
A.3	BACKREST MECHANISM	142
	APPENDIX B – INTRODUCTION TO MATROID THEORY	144
B.1	OTHER MATROID FORMULATIONS	145
	APPENDIX C – CONTRACTION AND DELETION OF MATROIDS	149
C.1	CONTRACTION AND DELETION OF LINEAR MATROIDS	149
C.2	CONTRACTION AND DELETION OF GRAPHS AND GRAPHICAL MATROIDS	152
	APPENDIX D – SETS OF MECHANISMS ENUMERATED	154
D.1	RECONFIGURABLE PALM MECHANISMS ENUMERATED BY METHOD I	154
D.2	RECONFIGURABLE PALM MECHANISMS ENUMERATED BY METHOD II	156
D.3	MECHANISMS ENUMERATED BY THE TYPE SYNTHESIS METHOD	157
D.4	LEG-REST MECHANISMS ENUMERATED BY THE TYPE SYNTHESIS METHOD	158
D.5	BACKREST MECHANISMS ENUMERATED BY THE TYPE SYNTHESIS METHOD	159

1 INTRODUCTION

Choosing the joints of a mechanism is one of the biggest challenges for a mechanism designer. The designer's choice must consider design requirements, the workspace of the mechanism, additional freedoms that a set of joints together may provide, among many other issues. For a simple mechanism, this task may not prove very challenging for an experienced designer, but an inexperienced designer may find quite challenging defining why and how to select the joints. The definition of joints proves yet more challenging when considering mechanisms already patented. The designer should seek new and different mechanisms so a new device is a possible outcome of the work.

For such work, designers can rely on mechanism design methodologies. These methodologies consist of steps required in a design process to get from basic requirements to an innovative device. The mechanism design methodologies classification proposed by Murai (2019) is used herein, in which the methodologies are classified as building blocks-based methodology (BBBM), enumeration-based methodology (EBM) or specialized methodology (SM). Building block-based methodologies (BBBM) use atlases or databases as starting point in the design process by assembling structures present in databases. Enumeration based methodologies (EBM) seek to enumerate every possible mechanism with the same set of structural characteristics. Finally, specialized methodologies (SM) are those methodologies that do not belong to either of the previous categories, using many different approaches and usually focusing on specific mechanism structures or classes.

Screw theory is a powerful tool when analysing or designing mechanisms by the possibility of modelling freedoms or constraints present in couplings. Screws are used to represent forces, moments, linear or angular velocities of rigid bodies as a spatial vector.

The multibody systems modelling proposed by Davies (1981) is an adaptation of Kirchhoff's laws and uses screw theory to model the interaction between the bodies, as well as external forces. Davies' method allows the modelling of complex systems as matrices constituted by the freedoms or constraints present in the multibody system. Through Davies' method, every velocity, force or constraint present in a mechanism can be calculated, thus creating either the static or kinematic models of said mechanism.

Davies' method is usually used as a tool for analysing mechanisms, but it can also be used when designing mechanisms. Carboni (2015) presented a method to enumerate every possible self-aligning mechanism from an overconstrained seed mechanism using Davies' method and matroid theory, first applying this method to enumerate self-aligning mechanisms based on the Tripteron mechanism (RICHARD et al., 2006). Later, the methodology proposed by Carboni was used for enumerating sev-

eral self-aligning mechanisms: reconfigurable palm (BARRETO et al., 2018), leg-rest of a hospital bed (ARTMANN et al., 2019a) and a clamping device (ARTMANN et al., 2019b).

This thesis focuses on enumeration of mechanisms using Davies' method and matroid theory. The matrices created using Davies' method for representing the static model of the mechanisms are used alongside matroid theory to enumerate new mechanisms. Herein a new approach is presented to select self-aligning mechanisms enumerated by matroids using contraction and deletion of matroids. Moreover, a new approach of enumerating self-aligning mechanisms is also proposed.

1.1 SYNTHESIS OF SELF-ALIGNING MECHANISMS

In a manufacturing or assembly line, tolerance management is vital to the whole production line. Each piece of the system, such as links and joints, has manufacturing uncertainties. During assembling, these uncertainties are added up. If manufacturing precision is adequate for the system's working characteristics, then no problems arise. However, if the precision is not adequate these errors may prevent the system to operate correctly and generate internal stress in the parts, decreasing reliability and even preventing the assembly of the system (WHITNEY, D. E., 2004).

Mechanisms and robots generally need strict manufacturing and assembly tolerances, in order to avoid removing or adding unwanted freedoms and constraints. By using self-aligning methodologies, *i.e.* eliminating redundant constraints, the designer can create a new mechanism that will, essentially, be easier to manufacture and assemble.

A mechanism without redundant constraints is known as self-aligning. A redundant constraint is defined by Reshetov (1979) as a constraint whose elimination does not change the mobility of the mechanisms. The elimination of redundant constraints, *i.e.* turning the original mechanism into a self-aligning one, does not change the way the mechanism works. In other words, the self-aligning derived mechanism has the same mobility and workspace of the original mechanism.

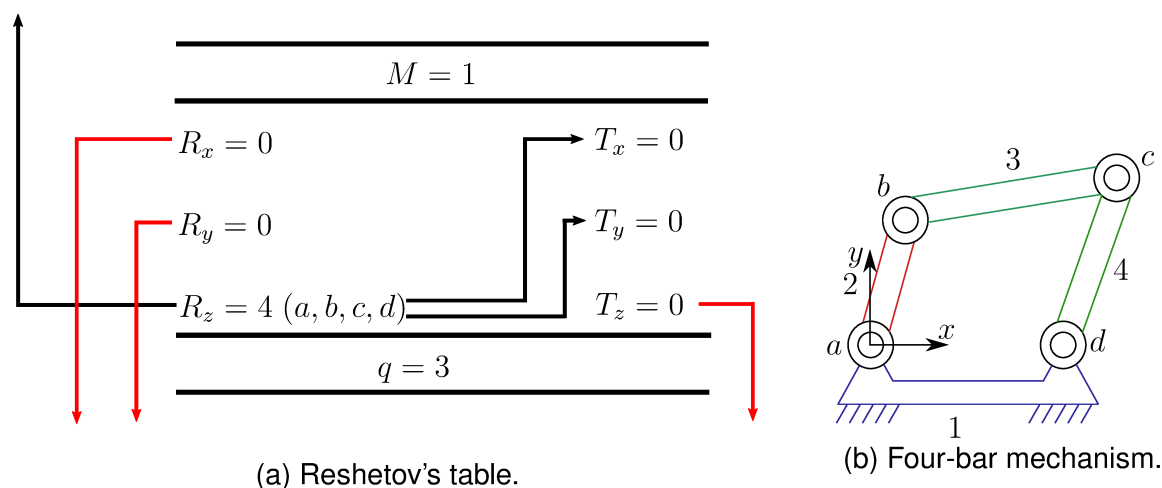
On the other hand, self-aligning mechanisms present in general less stiffness when compared to over-constrained ones (PASHKEVICH et al., 2009). Also, the cost of manufacturing self-aligning mechanisms must be considered when compared with the cost of manufacturing over-constrained mechanisms with stricter tolerances. It is up to the designer to decide whether self-aligning is important to the mechanism or not.

1.1.1 Reshetov's Method

Reshetov (1979) introduced a method for creating self-aligning mechanisms based on a seed mechanism with overconstraints. A seed mechanism can be described

as the model used as input in design methodologies, such as Reshetov's method. In his method a table is used for separating the freedoms present in the mechanism according to the circuit each joint is situated. From the table, the freedoms and redundant constraints are easily verified, as well as how to improve the mechanism. An example of Reshetov's table is shown in Figure 1.

Figure 1 – Reshetov's table of the overconstrained four-bar mechanism.



Source – Adapted from Carboni (2015).

The freedoms are the arrows pointing upwards while the redundant constraints are the arrows pointing downwards. The table is constructed based on the joints present in the mechanism and will be later explained in the next chapter. The method by Reshetov is easy to use, however counterexamples exist for multi-loop mechanisms, such as the Tripteron parallel manipulator (CARBONI, 2015). Moreover, Reshetov's method does not contain a technique for enumerating a variety of self-aligning mechanisms. For generating new mechanisms, the designer must manually create and evaluate new mechanisms.

1.1.2 Carboni's Method

Carboni (2015) presented a method to analyse overconstrained mechanisms using Davies' method and screw theory and the proof of the method is provided. Carboni also introduced a method for enumerating self-aligning mechanisms using Davies' method and matroid theory. By using both Davies' method and matroid theory, Carboni achieved the task of enumerating every self-aligning mechanism based on the seed mechanism. In his approach, Carboni used greedy algorithm for selecting self-aligning mechanisms. Despite enumerating every possible self-aligning mechanism, using greedy algorithm proved to be difficult when applying design requirements for selecting mechanisms. The weights used by the greedy algorithm are defined by the

user, which means a prior knowledge is required, including how the mechanism works as well as matroid theory and greedy algorithm.

Using Carboni's methodology, Barreto et al. (2018) evaluated the metamorphic reconfigurable palm of the KCL/TJU hand. Barreto et al. (2018) introduced a new approach for generating self-aligning mechanisms based on Reshetov (1979). Two methods for generating self-aligning mechanisms were discussed: adding freedoms to existing joints or adding new joints with the purpose of adding freedoms for the self-aligning method.

Later, Artmann et al. (2019a) presented a method to filter mechanisms generated by the methodology developed by Carboni (2015) and applied the methodology as well as the filter to a hospital bed mechanism. Using the method proposed in Artmann et al. (2019a), different mechanisms were created, from hospital bed mechanisms to clamping devices.

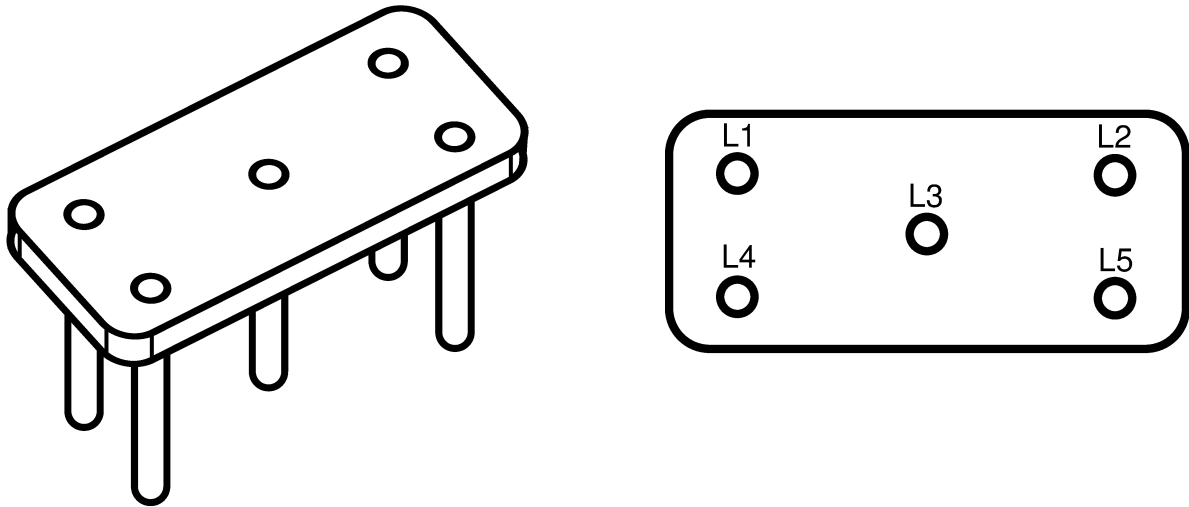
All these works employed Davies' method and matroid theory to enumerate mechanisms; however, the enumeration has always been focused on spatial self-aligning mechanisms. Furthermore, with the exception of the work by Barreto et al. (2018), all of the works focused solely on changing the types of existing joints. Each of these works also faced challenges due to the great number of results generated by means of matroid theory, requiring a considerable computer time for enumeration and mechanism selection.

1.2 A FIVE LEGGED TABLE

This section uses a simple example of the redesign of a five legged table to introduce many concepts that will be used later on this work. Suppose a company manufactures five legged tables; however, due to problems in the manufacturing the legs do not have the same length. The assembled table always has a wobble. The company then hires a designer to review the design of the table to solve the issue. Since the table has five legs, and the minimal required number of legs for a table is three, two legs can be removed and thus improve the stability of the table. It can be said that two legs are redundant.

The designer studies the table and enumerates the legs according to Figure 2.

Figure 2 – Table with Legs Labelled.



Source – From the author.

The designer decides that two legs will be removed from the original table to solve the redundant legs problem. Every possible combination of legs consisted in groups of three are listed by the designer and presented in Figure 3. The legs that should remain on the table are placed on the left written in bold, while on the right inside the parenthesis the legs that should be removed legs are placed written in italic.

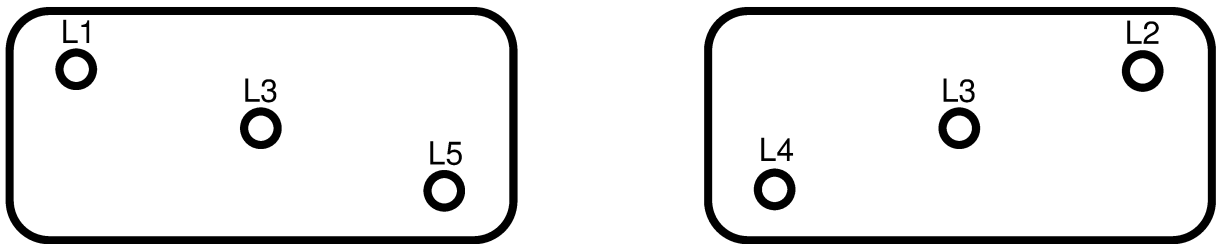
Figure 3 – List with the Combinations possible for Legs for New Table.

L1, L2, L3 (<i>L4, L5</i>)	L1, L4, L5 (<i>L2, L3</i>)
L1, L2, L4 (<i>L3, L5</i>)	L2, L3, L4 (<i>L1, L5</i>)
L1, L2, L5 (<i>L3, L4</i>)	L2, L3, L5 (<i>L1, L4</i>)
L1, L3, L4 (<i>L2, L5</i>)	L2, L4, L5 (<i>L1, L3</i>)
L1, L3, L5 (<i>L2, L4</i>)	L3, L4, L5 (<i>L1, L2</i>)

Source – From the author.

A total of ten three legged table possibilities are found; however, not every one of these possibilities are feasible. Figure 4 presents two unfeasible combinations.

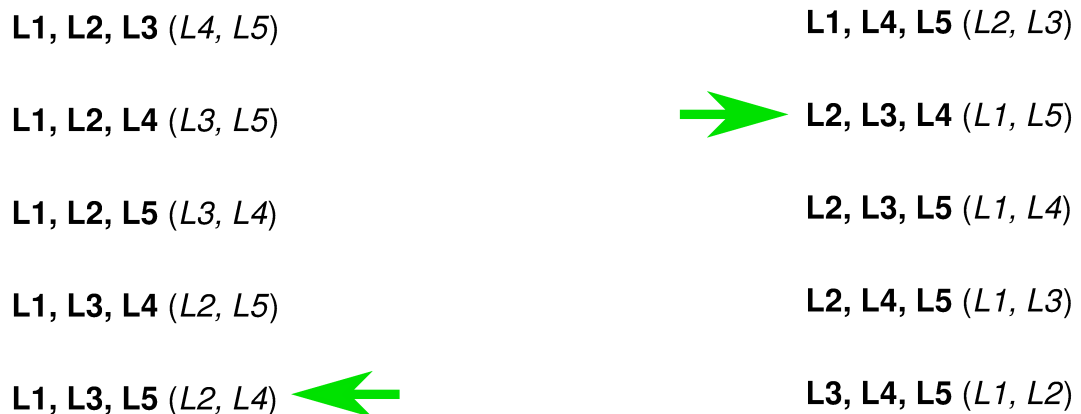
Figure 4 – Unfeasible Legs Combinations.



Source – From the author.

If either the combination of legs L1, L3 and L5 or the combination L2, L3 and L4 are chosen and a table is manufactured, the table will not be stable due to the alignment of the legs in a single line. Thus, these two combinations are removed from the list, remaining eight possible combinations for the designer to choose. The combinations L1, L3 and L5 and L2, L3 and L4 are marked on the list:

Figure 5 – Unfeasible combinations on the list.



Source – From the author.

then removed from the list, as shown in Figure 6.

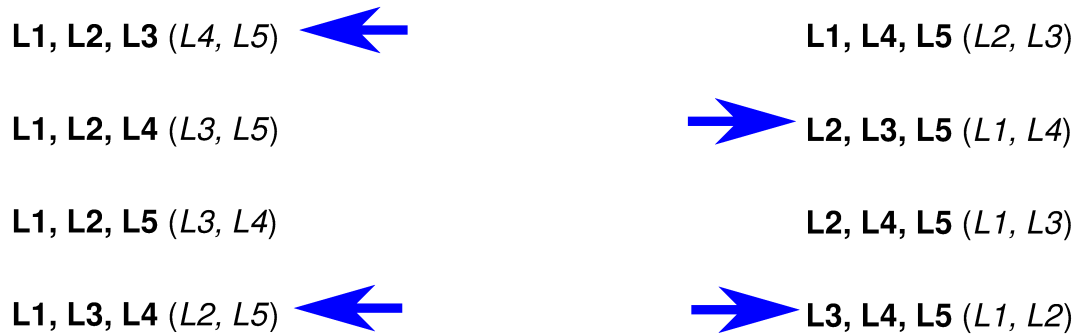
Figure 6 – New list with the combinations possible for legs for new table.



Source – From the author.

Suppose a design requirement states that the middle leg L3 must be present in the new table. The designer then reviews which combinations present the leg L3, shown in Figure 7.

Figure 7 – Combinations of Legs with L3 Present.



Source – From the author.

The combinations without L3 to the left written in bold are then removed, with only four combinations remaining as shown in Figure 8.

Figure 8 – Remaining Combinations.



Source – From the author.

Since the leg L3 will definitely be on the redesigned table and L3 is present in each of the remaining combinations, the element L3 can be removed from the combinations to simplify the list. The remaining simplified combinations are shown in Figure 9.

Figure 9 – Simplified Remaining Combinations.



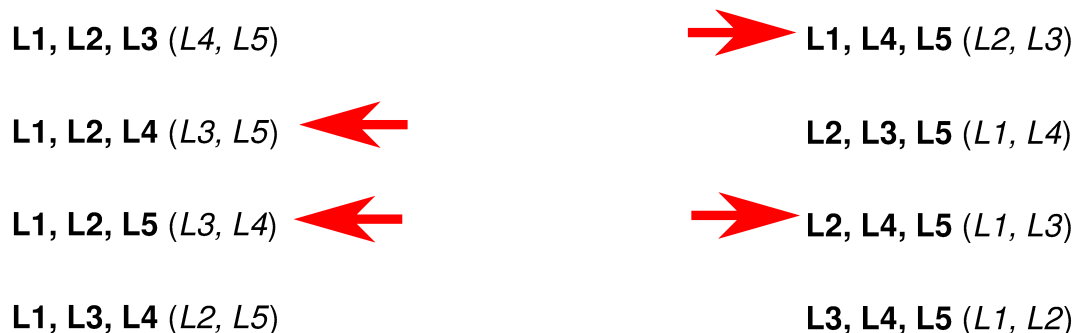
Source – From the author.

Finally, the designer has a list with only four possible combinations of three legged tables derived from the original five legged table with the leg L3 present.

Now, suppose that at the beginning of the process the designer had a different design requirement. Instead of keeping the leg L3, the designer has to remove the middle leg from the table. Recall that the elements to the right written in italic are those that must be eliminated. From the eight initial possible combinations, the designer

reviews which of these have the element L3 to the right inside the parenthesis, as demonstrated in Figure 10.

Figure 10 – Combinations of Legs with L3 Removed.



Source – From the author.

The designer then removes the combinations that do not comply with the design requirement, with the remaining combinations shown in Figure 11.

Figure 11 – Remaining Combinations.



Source – From the author.

The element L3 is present on the right side inside the parenthesis for all the remaining combinations, so the designer removes that element from the combinations to simplify the list, resulting in the combinations of Figure 12.

Figure 12 – Simplified Remaining Combinations.



Source – From the author.

After demonstrating the steps taken in the selection process used by the designer, those steps will be compared to what will be discussed later on this work.

The table with redundant legs is considered equivalent to mechanisms with redundant constraints. The redundant legs do not necessarily cause the table to have a wobble, but could create problems during the use of the table. The removal of the redundant legs will guarantee the stability of the table, but could decrease the stiffness

or the maximal load supported by the table. Analogous considerations are valid for mechanism with redundant constraints and for the removal of those constraints.

The enumeration of new mechanisms from an existing mechanism with redundant constraints is similar to the five legged table example and is achieved using matroid theory throughout this work.

The elements to the left written in bold on the list of combinations are legs present in the possible new tables, while the elements to the right written in italic are legs that must be removed.

The combinations of legs that would result in three legs in a straight line are comparable to linearly dependent constraints. If those legs aligned were chosen, the table would gain a mobility and would tumble. This is only true if the maximal set of legs are aligned, *i.e.* the alignment of two legs do not cause problems in the design. A mechanism with a set of constraints presenting linear dependence would cause the mechanism to gain an additional unwanted freedom. One of the benefits of using matroids for the enumeration of self-aligning mechanisms is the fact that it ensures the designer that every basis is a maximal set without linear dependence that would cause an unwanted freedom.

The steps taken by the designer to ensure that the leg L3 is present in the new table is similar to the matroid operation called contraction. In this work, the contraction of matroids is used to select a desirable set of constraints, similar to what the designer did, but instead of legs, constraints are chosen based on design requirements.

The dual of the contraction is the deletion. When defining that the leg L3 must be removed, the designer selected the combinations that have the element L3 present to the right inside the parenthesis. Then the element L3 was erased from the list because it was present in every remaining combination. Using the deletion of matroid elements the same property is achieved, thus selecting which constraints must be removed from the mechanism.

Comparing the lists of legs obtained in the five legged table example with matroids, the matroid bases consist of the remaining linearly independent elements which are the elements to the left of the dividing symbol. The elements to the right of the dividing symbol consist of the cobases of the matroid.

Table 1 presents a summary of the comparison between the concepts discussed in the five legged table example and the tools related to matroids and mechanisms that are used throughout this work. The column respective to the chapters indicates in which the equivalent concept is used first in this thesis.

Table 1 – Comparison between the five legged table example with matroid and mechanisms.

Table example	Matroids and mechanisms	Chapter
Table with five legs	Mechanism with redundant constraints	Chapter 2
Table with three legs	Self-Aligning Mechanism	Chapter 2
Enumeration of stable legs	Enumeration of matroid bases	Chapter 5
Combination of legs written in bold	Matroid basis	Chapter 5
Combination of legs written in italic	Matroid cobasis	Chapter 5
Selection of legs to remain	Matroid contraction	Chapter 6
Selection of legs for removal	Matroid deletion	Chapter 6

Source – From the author.

1.3 THESIS OBJECTIVES

The objective of this thesis is to provide a type synthesis methodology for generating and selecting new mechanisms.

Specific objectives are listed below.

- To provide theoretical tools that aid the design of new mechanisms.
- To provide theoretical tools that aid the enumeration and selection of mechanisms.
- To provide a new approach for the synthesis of self-aligning mechanisms.

The methods and tools presented herein are expected to assist designers when developing new mechanisms, facilitating in creating and selecting innovative mechanisms from sets of design requirements with easy to use mathematical tools employing open-source software.

1.4 THESIS CONTRIBUTIONS

This thesis contributes to the enumeration and selection of mechanisms enumerated using Davies' method and matroid theory. The self-aligning mechanisms enumeration proposed by Carboni (2015) is extended for creating mechanisms with different kinematic properties. A new method to select the enumerated self-aligning mechanisms is also proposed, improving the translation of design requirements into filters. The specific contributions are as follows.

1.4.1 Davies' Method

Based on the formulation of Davies' method, new approaches for analysing and enumerating mechanisms are proposed. The specific contributions are:

- The novel concept of Reshetov virtual joint is introduced in this thesis. This new concept allows the designer to input a rigid structure inside the multibody formulation proposed by Davies (1981). Reshetov virtual joints are used as means of generating self-aligning mechanisms, receiving different freedoms and thus creating different mechanisms.
- Using the proposed concept of Reshetov virtual joints, this work shows how redundant constraints can be removed for creating new mechanisms with different kinematic structures.

1.4.2 Self-Aligning Mechanisms Synthesis

Based on the enumeration of self-aligning mechanisms methodology by Carboni (2015), a new tool for modelling mechanisms is applied. Furthermore, a new method for selecting self-aligning mechanisms using matroid theory is discussed. The specific contributions are:

- Reshetov virtual joints are used for designing new self-aligning mechanisms. When creating new self-aligning mechanisms using Davies' method, the designer is required to modify existing joints, which could be undesirable. The new concept of Reshetov virtual joints provides the designer more flexibility for creating new mechanisms.
- Using contraction and deletion of matroids, a new method for selecting mechanisms enumerated by matroids is proposed. Unlike previous works where the methods were hard to implement or to use design requirements as filters, contraction and deletion of matroids provide a user-friendly tool for selecting mechanisms as well as a practical approach of using design requirements in the synthesis process.
- A novel notation for representing matroid operations. Based on the joint types according to Reshetov (1979), a representation of both contraction and deletion of matroids applied to mechanisms synthesis is created. Furthermore, the representation can be used as a list of matroid operations already finished.

1.5 THESIS OUTLINE

This thesis is organized in ten chapters.

Chapter 1 is an introduction to the research area of the thesis, presenting also the objectives of this work.

In Chapter 2 mechanism design methodologies are discussed. An overview of the main steps normally present in mechanisms design are discussed, focusing mainly

in the type synthesis. Different enumeration based mechanism design methodologies are introduced.

In Chapter 3 screw theory and Davies' method are briefly discussed. Both kinematic and static model of mechanism using Davies' method are demonstrated. Linear dependence of freedoms and constraints are also discussed.

The concept of Reshetov virtual joints is proposed in Chapter 4. Using this new concept, new Baranov gripper mechanisms are designed employing also Davies' method and Baranov chains.

In Chapter 5 matroid theory is introduced. Throughout this chapter, linear and graphical matroids are used to explain different properties and operations.

In Chapter 6 contraction and deletion of matroids are used to create a new method to select self-aligning mechanism enumerated. A notation for representing the matroid operations is also proposed.

In Chapter 7 a case study is presented. The anthropomorphic reconfigurable palm of the KCL/TJU hand is used as a seed mechanism for generating new self-aligning mechanisms.

In Chapter 8 a new methodology for enumerating mechanisms is proposed. In this methodology, the new mechanisms are not kinematically equivalent to the seed mechanism.

In Chapter 9 the new design methodology is used for generating hospital bed mechanisms.

The conclusions of this thesis and future work are presented in Chapter 10.

Appendix A contains a deeper discussion of matroid theory, presenting properties and formulations.

Appendix B presents a deeper demonstration of contraction and deletion of matroids.

Appendix C contains the lists of the mechanisms enumerated and selected throughout the thesis.

2 MECHANISM DESIGN TECHNIQUES AND METHODOLOGIES

The objective of this section is to introduce the different approaches of synthesis commonly applied in the mechanisms design methodology. Initially the number synthesis is presented, then the type synthesis, and finally the dimensional synthesis is discussed. In the sequence, four different mechanism design enumeration methodologies are introduced.

2.1 NUMBER SYNTHESIS

Number synthesis is the procedure to enumerate kinematic chains with given characteristics (TISCHLER et al., 1995a). In general, the mobility, screw system, number of joints, links and loops are used for the kinematic chain enumeration. The relation between these characteristics are given by the Grübler-Kutzbach formulation (GOGU, 2005a), shown in Equation 1:

$$M = \lambda(n - j - 1) + \sum_{i=1}^j f_i \quad (1)$$

where M is the mobility of the mechanism, λ is the screw system order, j is the number of joints, n is the number of links and f_i is the degrees of freedom of each kinematic pair. Equation 1 can be written differently if only joints with one degree of freedom are considered, resulting in Equation 2.

$$M = \lambda(n - j - 1) + j \quad (2)$$

Equation 2 can be modified to include the number of independent loops ν , shown in Equation 3.

$$M = j - \lambda\nu \quad (3)$$

Equation 1 is known to fail in certain conditions (MRUTHYUNJAYA, 2003; GOGU, 2005b), and thus Equations 2 and 3 also have limitations. Based on the work by Malyshv, Ozol and Shamaidenko, Reshetov (1979) presents an equation for the mobility of mechanisms considering redundant constraints:

$$M = \lambda(n - j - 1) + \sum_{i=1}^j f_i + q \quad (4)$$

where q is the number of redundant constraints present in the mechanism.

Tischler et al. (1995a) define the goal of the number synthesis as the discovery of every possible arrangement using a given number of joints and links. It is important to emphasize that the kinematic chains should be proper, as defined by Tischler et al.

(1995a), or nondegenerate, as pointed out by Simoni and Martins (2007). Kinematic chains which have $M \leq 0$ are usually not of interest in the study of mechanisms. More information and examples on number synthesis are available in Tischler et al. (1995a), Tischler et al. (1995b), Simoni and Martins (2007), Simoni et al. (2009), Martins et al. (2010). Different reviews of number synthesis are available in Mruthyunjaya (2003), Simoni et al. (2011) and Yan and Chiu (2015).

2.2 TYPE SYNTHESIS

The main objective of the type synthesis is to determine which types of joints are employable on the mechanism being developed. According to Hartenberg and Denavit (1964), at this phase the types of links and joints used in the mechanism are defined. Kong and Gosselin (2007) define type synthesis of parallel mechanisms as the procedure of finding every possibility of leg combinations for parallel mechanisms to perform a given movement.

The topological synthesis of mechanisms at times can be called the type synthesis of mechanisms, while often it is a combination of the number synthesis and type synthesis. Furthermore, depending on the author, type synthesis of mechanisms may assume different meanings. For example, Sandor and Erdman (1984) place number synthesis as a subcategory of the type synthesis. Tsai (2001) uses the term kinematic structure when generating new mechanisms with his methodology. According to Tsai, the term kinematic structure includes information from which links are connected by what types of joints, thus including number and topological synthesis in one category.

This design phase requires special attention from the designer, because several factors should be considered. More complex joints may prove troublesome in the production as they usually cost more. In addition, mechanisms that are hard to assemble considerably complicate the production and maintenance.

Considering the screw system used in a specific project, the possible types of joints are defined. For example, in the planar space, the revolute joint, the prismatic joint, cams and gears are available. Once the types of joints are defined, they are associated with the mechanism, and every possible combination is listed. This step commonly provides a high number of results, hence computer algorithms are frequently used to list and possibly filter the different combinations.

During the type synthesis, approaches such as self-aligning mechanisms are applicable. This approach enables the designer to create a mechanism easier to assemble which tolerates manufacturing flaws, facilitating also the maintenance of the device. Further information on self-aligning mechanisms or minimum constraint designs are available in Reshetov (1979), French (1985), Kamm (1990), Blanding (1999).

In this thesis, the preferred term is type synthesis when considering the phase of defining the types of joints of mechanisms, although topological and type synthesis

are considered synonyms. Furthermore, following the mechanism design procedures presented later, the number synthesis is considered independent to the type synthesis.

Ye and Li (2019) proposed a classification dividing the type synthesis of parallel mechanisms into three categories: motion-based methods, constraint-based methods and other methods. The motion based methods consists of creating a motion of the moving platform with the intersection of the allowed movements of each leg of the parallel mechanism. The constraint-based methods create mechanisms by the union of the constraints of the legs, thus forming the constraint space of the moving platform. Finally, Ye and Li (2019) present two approaches as other methods, an enumeration approach based on the Grübler-Kutzbach formula and a graph theory based method.

2.2.1 Motion-Based Methods

Ye and Li (2019) describe the motion-based methods for type synthesis of mechanisms as two different tasks. First, mathematical expressions are used to represent the movements of the bodies. Then, a correlation between the motions and the topological structures is created using mathematical tools.

Herve (1978) proposed a method for using Lie group theory in the synthesis of parallel mechanisms, and since many works have been published using this concept (KAROUIA; HERVE, 2000; LI et al., 2004, 2017; LEE; HERVE, 2010). The motion of the moving platform of a parallel mechanism using Lie group theory based method consists of the union of the motion set enabled by each leg of the mechanism. According to Ye and Li (2019), the type synthesis using Lie group theory has two advantages: a finite mobility is guaranteed and this method can be used with mixed rotations and translations whose sequence is specified. Despite these advantages, this method requires mathematical foundation of the designer.

Another motion-based type synthesis is the method based on generalized function sets G_F . The type synthesis process using the G_F sets is similar to the method based on Lie group theory. The G_F sets are composed by three translations and three rotations that represent the motion of the moving platform, and each of these elements may be a specific symbol or zero, thus representing the presence or absence of the motion. The G_F sets of the moving platform are found by the intersection of the generalized sets of every limb. More information on G_F is available in: Gao et al. (2002), Jialun Yang et al. (2011), Gao et al. (2011), Jialun Yang et al. (2012) and He et al. (2015).

Type synthesis of parallel mechanisms may also be done using the linear transformation method. This approach is based on the linear transformation from joint velocity in space to the output velocity in space. When using this method, conditions are identified so that they correspond to a desired motion, which generally is with no actuator locked and in particular cases with one actuator locked, thus guaranteeing independent mapping between input and output velocities. Then, limbs that comply with the

conditions identified are designed. Several parallel mechanisms with different degrees of freedom have been designed using this approach (GOGU, 2007, 2009, 2012).

Other synthesis methods are also available, such as finite screw method (YANG, S. et al., 2016, 2017; SUN et al., 2018; SUN; HUO, 2018) and position and orientation characteristic (POC) set method (YANG, T. et al., 2016).

2.2.2 Constraint-Based Methods

The synthesis of parallel mechanisms using constraint-based methods is established on the notion that constraints are restrictions to motions. When the moving platform has a specific constraint, that platform cannot perform a motion related to that constraint.

Screw theory is used in constraint based methods relying on reciprocal screws to design parallel mechanisms. Several different parallel mechanisms have been designed using this approach (HUANG; LI, 2002, 2003; FANG; TSAI, 2002, 2004; GUO et al., 2012; KONG; GOSSELIN, 2004a, 2004b). Ye and Li (2019) lists the steps required in this synthesis approach, placing as last step the requirement of checking the synthesised structure to ensure full mobility.

Virtual chains are also used in the type synthesis of parallel mechanisms mainly relying on screw theory. The main difference between this approach and the previous one is the description of the motion pattern of the moving platform using virtual chains (KONG; GOSSELIN, 2005a, 2005b, 2006, 2007).

A different constraint-based method used line geometry and line graphs (XIE et al., 2013, 2014), which according to Ye and Li (2019) can be regarded as a visual version of the screw theory methods.

Lastly, parallel mechanisms can also be generated using motion constraint generators, in which a limb with the desired constraints is used with several six degrees of freedom limbs. Several papers have been published demonstrating the synthesis of mechanisms using this method (ZHANG, D.; GOSSELIN, 2000; KUO; DAI, 2013; LU; HU, 2007).

2.2.3 Other Methods

Different approaches are available when considering type synthesis of general mechanisms, such as graph theory (LIU; CHOU, 1993; LU; LEINONEN, 2005; PUCHETA; CARDONA, 2007; PUCHETA et al., 2012; LU et al., 2014), Assur groups and virtual chains (CAMPOS et al., 2008; ZHANG, X. et al., 2019) and Davies' method together with matroid theory (CARBONI, 2015; CARBONI et al., 2017; BARRETO et al., 2018; ARTMANN et al., 2019a). This work focuses on the type synthesis of mechanisms using Davies' method and matroid theory.

2.2.4 Considerations About Type Synthesis

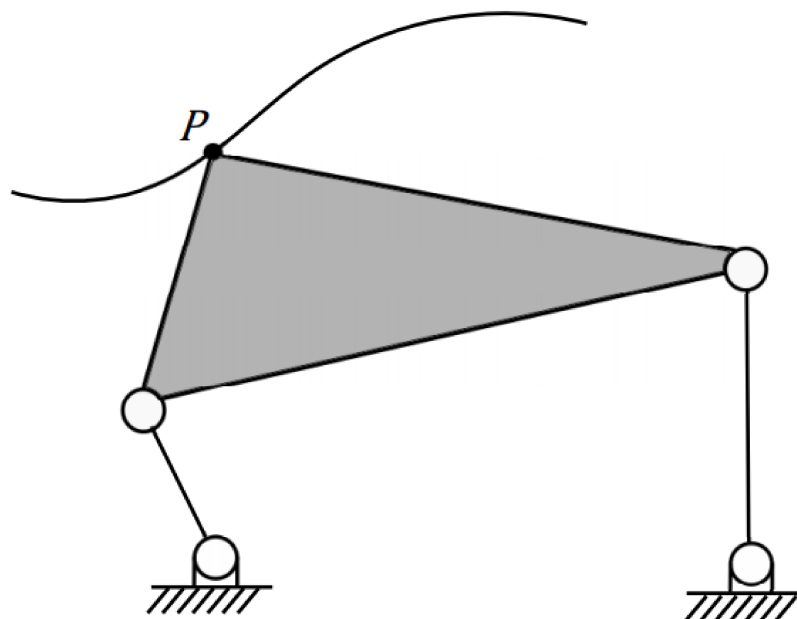
According to Ye and Li (2019), there are several methods for the type synthesis for parallel mechanisms with one moving platform and parallel limbs; however, there is a need for more methodologies that develop generalized parallel mechanisms, such as parallel mechanisms with multiple moving platforms and hybrid links.

The majority of the methods presented in the previous subsections place the synthesis of motion as the core of the synthesis methods; however, when designing special application systems, sometimes the desired motion is a complex combination of rotations and translations which increases considerably the challenge when using the methods presented. If the type synthesis is not used as means of defining the motion of the moving platform, new opportunities in synthesis methods arise. When the designer is not constrained to selecting joints aligned mainly in the coordinated axes, more complex movements are possible. Although challenging, the definition of the motions of the moving platform can be done in a later stage of the mechanism design process known as dimensional synthesis, thus making the type synthesis a phase of enumerating possibilities disregarding motion.

2.3 DIMENSIONAL SYNTHESIS

Sandor and Erdman (1984) claim that the dimensional synthesis is used to define the dimensions, starting points and conditions of pre-determined mechanisms for specific tasks and characteristics. These authors also assert that the most common tasks for the dimensional synthesis are function, path and motion. The function generation requires a correlation between the input and output links. The dimensions of the mechanism should satisfy the function that describes the relation between those links. An optimization in this case compares the desired function with the calculated correlation of the input and output of the dimensioned mechanism. The path generation uses a point on a link that is not connected to the fixed link to trace the path of that link using the fixed link as the reference. The path generation is demonstrated in Figure 13.

Figure 13 – Path generation.

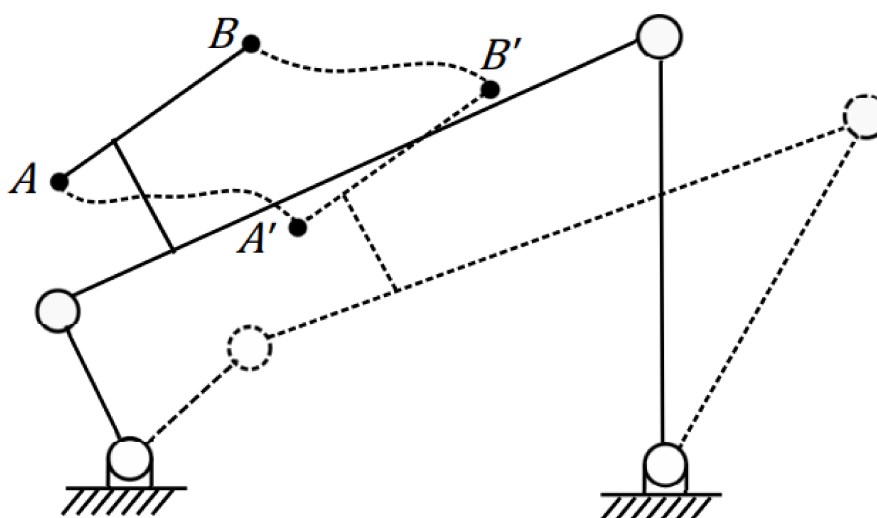


Source – (NUNEZ, 2014).

In the path generation method, the optimization is developed by comparing the desired trajectory with the trajectory performed by the dimensioned mechanism.

The motion generation in Figure 14, also called rigid-body guidance, is applicable when the whole body is required to perform a motion sequence.

Figure 14 – Motion generation.



Source – (NUNEZ, 2014).

This method differs from the path generation because it requires not only the path

desired but also the rotations of bodies. In the dimensional synthesis stage, optimization routines are essential to guarantee a satisfactory performance of the developed mechanism. In order to perform an optimization procedure, a mathematical model of the mechanism is required.

2.4 MURAI'S MECHANISM DESIGN METHODOLOGY

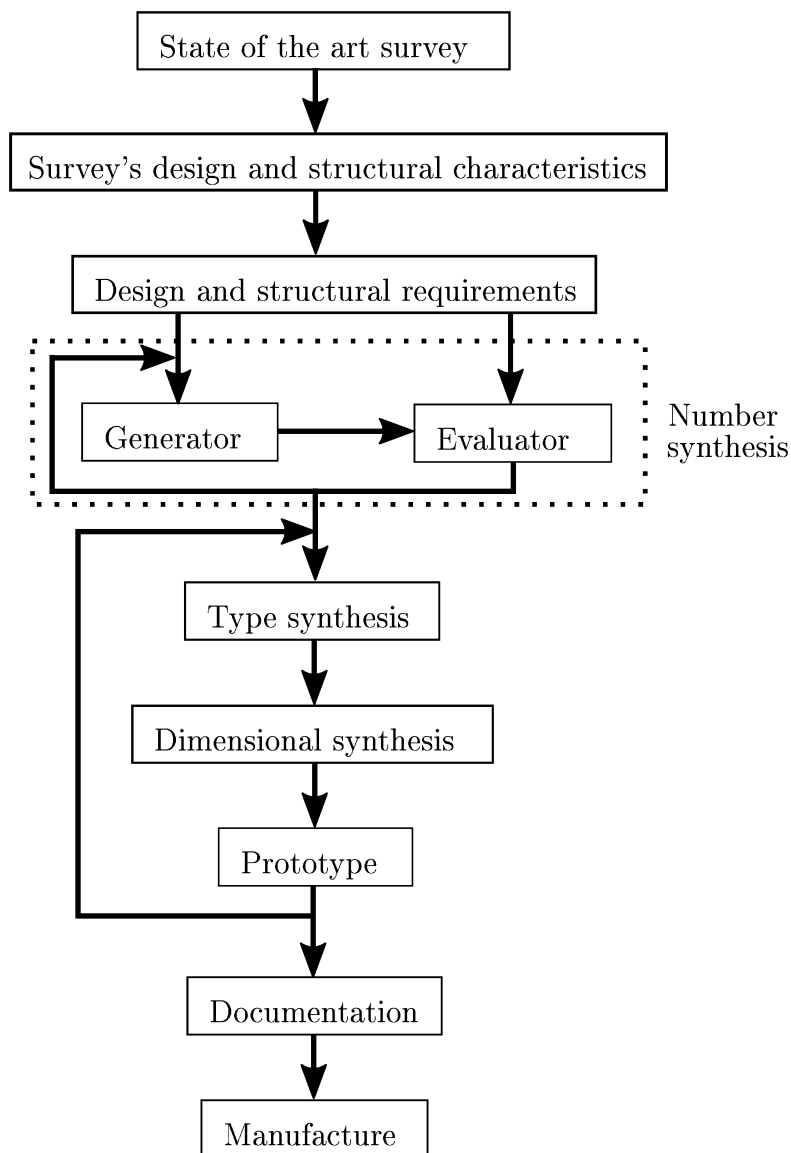
The methodology proposed by Murai (2019) combines different characteristics of the methodologies by Yan (1998) and Tsai (2001). The first step in Murai's methodology is the state of the art review, whose objective is to find existing mechanisms that fulfil the project requirements. This is done to get a wider understanding of the project area, analysing the key challenges. From the existing devices found in the state of the art review, characteristics, such as mobility and number of loops, are analyzed. The methodology presented by Murai (2019) requires a kinematic chains generator and evaluator.

Murai (2019) determines the following steps for the methodology:

- State of the art survey. Mechanisms that performs the desired or similar functions should be considered. In this step, customer requirements should also be listed.
- Identify the design and structural characteristics of the devices found in the state of the art survey.
- Determine the structural and design requirements for the project in development.
- Select structural characteristics from the requirements as inputs to the generator.
- Generate all possible kinematic chains.
- Evaluate all kinematic chains and eliminate all the kinematic chains that do not comply with the requirements. If after this step there is no kinematic chain remaining, the structural characteristics should be changed and the generator be re-evaluated.
- Perform the type synthesis for the selected kinematic chains.
- Perform the dimensional synthesis from the mechanisms developed in the previous step.
- Create a prototype.
- Develop the documentation of the prototype.
- Start the production.

A work-flow of Murai's methodology is shown in Figure 15.

Figure 15 – Methodology proposed by Murai (2019).



Source – Adapted from Murai (2019).

Other enumeration based mechanism design methodologies are also available: Yan (1998), Tsai (2001), Ding (2015). An in depth analysis and comparison of enumeration based mechanism design methodologies is available in Murai (2019).

2.5 SYNTHESIS OF SELF-ALIGNING MECHANISMS

In this Section, the synthesis of self-aligning mechanisms is discussed. Previously, self-aligning mechanisms were introduced as mechanisms without redundant

constraints, and redundant constraints were introduced using the definition by Reshetov (1979): a redundant constraint is a constraint whose removal does not change the mobility of the mechanism.

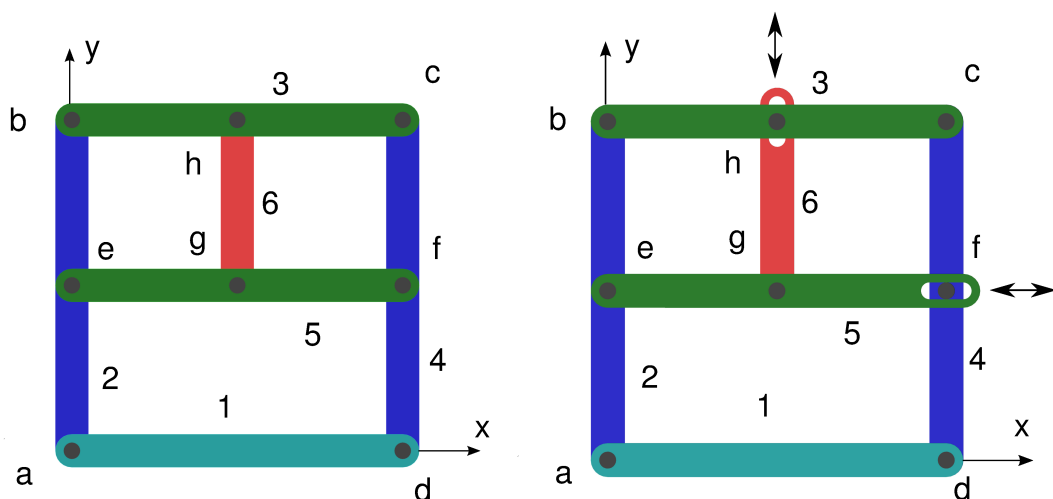
Self-aligning mechanisms can be created or analyzed in different workspaces. For example, a mechanism can be designed as self-aligning when considering it as a planar mechanism, but when considering it as a spatial mechanism there are still redundant constraints. Equation 5 (RESHETOV, 1979) presents the relation between the mobility M , the screw system order λ , the number of links n of the mechanism, the number of constraints present in the mechanism C and the number of redundant constraints q .

$$M = \lambda(n - 1) - C + q \quad (5)$$

In this work, the symbol for the number of redundant constraints used is the q , which is equivalent to the symbol C_N used by Davies in his notation.

Some examples of overconstrained mechanisms and a self-aligning counterpart will be presented. First, Figure 16 shows a mechanism with redundant constraints in the plane. This mechanism has $M = 1$, six links, eight joints and two redundant constraints. A self-aligning mechanism is derived by transforming joints f and h into pin-in-slot joints, removing a force constraint in the x-axis for joint f and a force constraint in the y-axis for joint h.

Figure 16 – Example of overconstrained and self-aligning mechanisms.

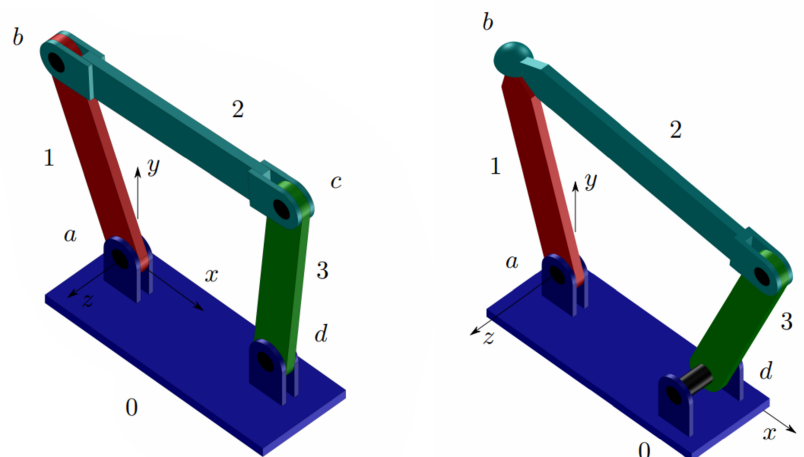


Source – Adapted from Carboni (2015).

In Figure 17, a four-bar mechanism is presented. If this mechanism is analyzed as a planar mechanism, there are no redundant constraints present; however, in the spatial screw system there are three redundant constraints. A self-aligning mechanism is derived from the overconstrained four-bar mechanism by removing two moment

constraints from joint b, thus transforming it into a spherical joint, and a force constraint in the z-axis from joint d, transforming it into a cylindrical joint.

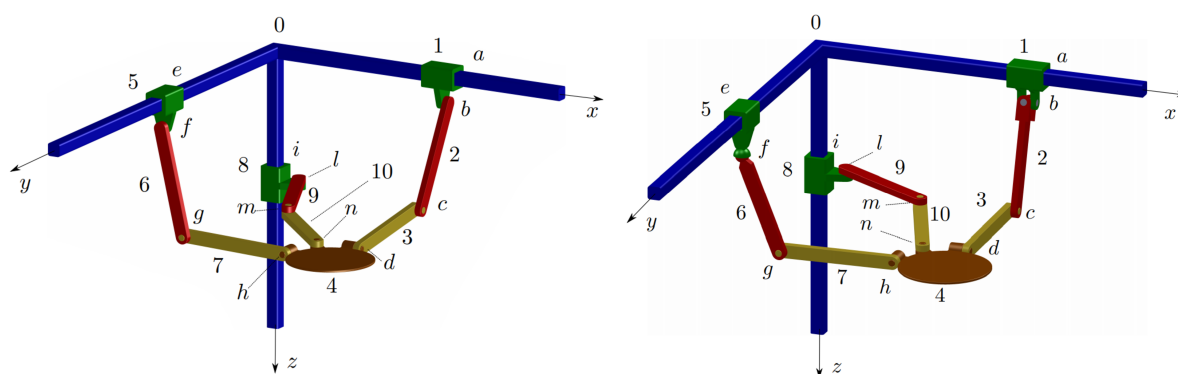
Figure 17 – Example of four-bar overconstrained and self-aligning mechanisms.



Source – Adapted from Carboni (2015).

Finally, Figure 18 presents the well known Tripteron parallel manipulator, as well as a self-aligning mechanism derived from the Tripteron. The Tripteron is a spatial mechanism with $\lambda = 6$ and it has $C = 18$ constraints, $n = 11$ links, $M = 3$ mobilities and $q = 3$ redundant constraints. The self-aligning mechanism was created by transforming joint b into a universal joint and joint f into a spherical joint.

Figure 18 – Example of four-bar overconstrained and self-aligning mechanisms.



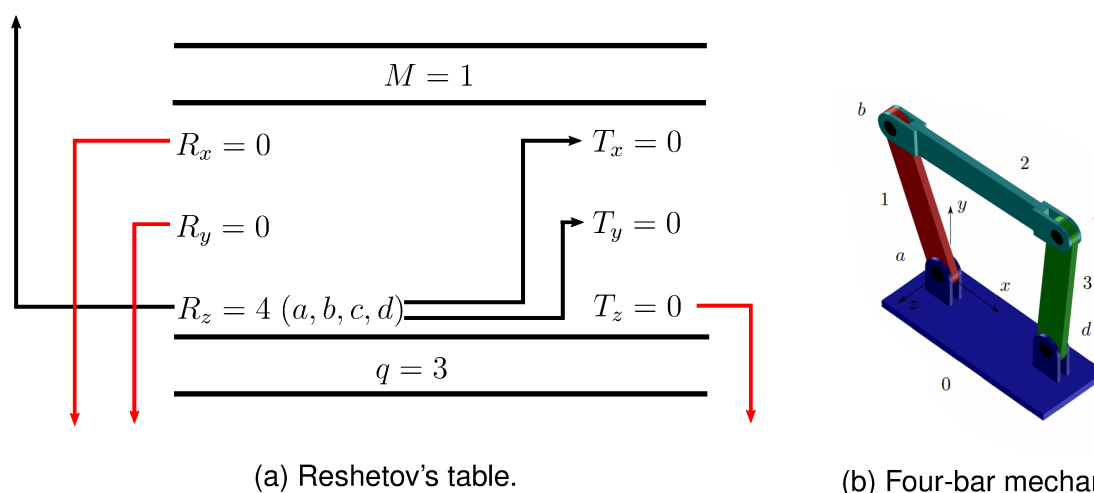
Source – Adapted from Carboni (2015).

From the examples presented, it becomes clear that the screw system order λ of the mechanism is important when evaluating redundant constraints. A self-aligning mechanism in the plane may not be self-aligning in a spatial condition.

There are different approaches for the synthesis of self-aligning mechanisms. Reshetov (1979) proposed a very practical approach of designing self-aligning mecha-

nisms based on observation of mechanisms. An overconstrained mechanism is evaluated by separating the existing freedoms enabled by the joints according to the modes of freedom, *i.e.* linear or angular freedoms, in the coordinated axes. The overconstrained four-bar mechanism of Figure 17 is used as an example for creating Reshetov's table, shown in Figure 19. The overconstrained four-bar mechanism has four revolute joints in the z-axis, thus this mechanism has $R_z = 4$. Every other freedom is equal to zero. According to Reshetov, an angular freedom can be transformed into a linear freedom on a perpendicular axis, *i.e.* an angular freedom in the z-axis can become a freedom in the x or y-axis. Therefore, two of the existing angular freedoms in the z-axis is considered a freedom in each of the other axes. The freedoms that remain null are considered redundant constraints, while remaining freedoms are considered mobilities. The arrows help to follow the freedom and redundant constraints in Figure 19, with freedom pointing upwards while redundant constraints point downwards.

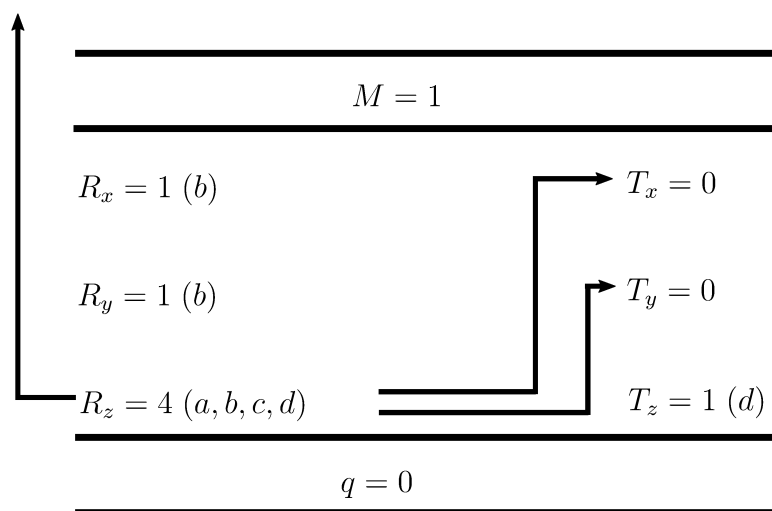
Figure 19 – Reshetov's table of the overconstrained four-bar mechanism.



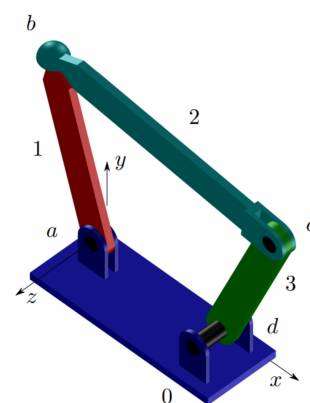
Source – Adapted from Carboni (2015).

Reshetov's table for the overconstrained four-bar mechanism has the result already discussed previously. The self-aligning four-bar mechanism from Figure 17 has a spherical joint in place of the revolute joint b while joint d is now a cylindrical joint. Reshetov's table for the self-aligning four-bar mechanism is shown in Figure 20.

Figure 20 – Reshetov’s table of the self-aligning four-bar mechanism.



(a) Reshetov’s table.



(b) Self-aligning four-bar mechanism.

Source – Adapted from Carboni (2015).

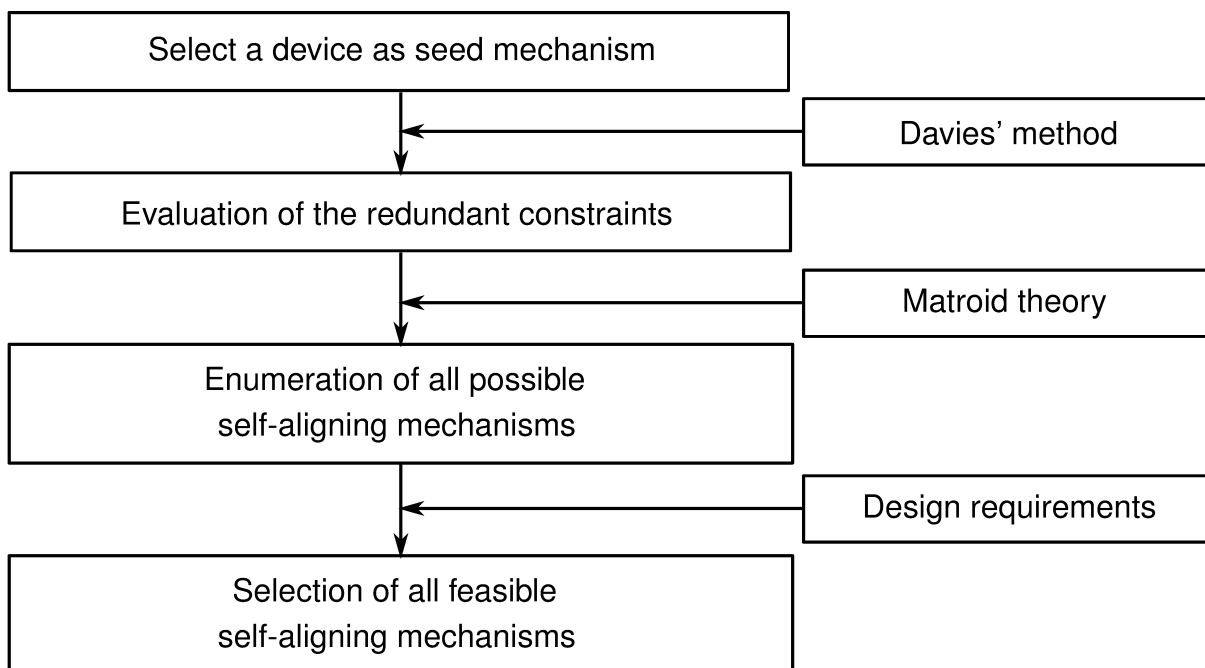
In Figure 20 there are no more arrows pointing down, thus achieving a self-aligning mechanism.

Reshetov’s table, although practical and easy to use, is not always valid. Carboni (2015) discussed the Tripteron mechanism as a counterexample of Reshetov’s approach, calculating a mechanism with $M = 2$ and $q = 2$, which is simply wrong.

Carboni (2015) proposed a method for creating self-aligning mechanism based on Davies’ method which does not have the shortcomings of Reshetov’s approach. Although Carboni’s method works well for mechanisms that Reshetov’s method does not, Carboni’s method is more complex and requires more mathematical tools.

Carboni’s method of self-aligning mechanisms enumeration use mechanisms with redundant constraints as inputs. These mechanisms with redundant constraints are henceforth called seed mechanisms. Seed mechanisms can be planar, spherical or spatial, the only requirement is the presence of redundant constraints. The process of enumerating redundant constraints is done using Davies’ method alongside matroid theory, both will be introduced in the future chapters. A summary of the method is presented in Figure 21.

Figure 21 – Flowchart of Carboni's Method for Enumerating Self-Aligning Mechanisms.



Source – Adapted from Artmann et al. (2019b).

The enumeration of mechanisms using Davies' method consists on finding every possible self-aligning mechanism based on the seed mechanism. The seed mechanism is evaluated and the network action matrix, from the static model of the mechanism according to Davies' method, is used to create a matroid. Each basis of the matroid is a new self-aligning mechanism (CARBONI, 2015). One challenge with this approach of enumeration of mechanisms is the large amount of results generated. Hence, beside the enumeration method, selection methods are also required when enumerating mechanisms using matroid theory.

The first approach to select mechanisms was proposed by Carboni (2015), using a greedy algorithm for the selection of desirable basis of the matroid. The utilization of the greedy algorithm is commonplace when considering matroid theory, but it is not without challenges. Given a matroid with n elements in the ground set, a weight must be selected to each element. The greedy algorithm will then search for an optimal basis with the highest sum of the weights and will return an independent set. When using greedy algorithm for the selection of the mechanisms enumerated by matroids, each element of the ground set is a constraint present in the seed mechanism. Therefore, the designer must choose which are the weights for each constraint before using the greedy algorithm. For an experienced designer, this task may not be difficult, but for a novice designer choosing weights is a challenge that may result in a poor selection. Finally, another problem with using greedy algorithm is the requirement of enumerating bases of the matroid. As the matroids become bigger, the number of bases increase

exponentially. If enumerating bases is required for a selection method, the analysis of complex systems may become unfeasible.

Another approach to select bases was proposed by Artmann (ARTMANN, 2019; ARTMANN et al., 2019a) called the cobases binary matrix $[N]_{\mu,C}$. After the creation of the matroid on the network action matrix, every cobasis of the matroid is enumerated. The number of rows of this matrix is the number of cobases μ of the matroid, while each column refers to a constraint of the ground set. The elements $n(i,j)$ of matrix $[N]_{\mu,C}$ are defined by:

$$n(i,j) = \begin{cases} 1 & \text{if the constraint } j \text{ is in the cobasis } i \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Using Equation 47, the presence of a constraint in the cobasis is translated into a number 1 in the column of that constraint. If the constraint is not present in the cobasis, the element is 0. Equation 7, extracted from Artmann (2019), shows an example of the $[N]_{\mu,C}$ matrix.

$$[N]_{112,20} = \begin{matrix} & \dots & \hat{\$}_{bR}^a & \hat{\$}_{bS}^a & \dots & \hat{\$}_{cW}^a & \hat{\$}_{dR}^a & \hat{\$}_{dS}^a & \hat{\$}_{dU}^a & \hat{\$}_{dV}^a & \hat{\$}_{dW}^a \\ \begin{matrix} l_1 \\ l_2 \\ l_3 \\ \vdots \end{matrix} & \begin{bmatrix} \dots & 1 & 1 & \dots & 0 & 0 & 0 & 0 & 0 & 1 \\ \dots & 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 & 0 \\ \dots & 0 & 0 & \dots & 0 & 1 & 1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \end{matrix} \quad (7)$$

Each of the l_i rows are different cobasis representing different mechanisms. The columns with 1 are constraints present in the cobasis.

By creating the cobases binary matrix, the designer has a large yet simple list of bases and constraints enumerated using matroid theory. The next step of the method proposed by Artmann is using selection criteria as means of selection process. The selection criteria are defined using project requirements, such as the requirement of using only lower kinematic pairs. Each selection criterion K_i is formed by a list of cobasis that comply with that criterion and the final group K_F of cobases that comply with all of the criterion is found by the intersection of the sets selected by each of the K_i criterion. The set K_F is the group of self-aligning mechanisms that comply with every design requirement. The cobases binary matrix is a great technique to select self-aligning mechanisms with different characteristics, as different boolean algebra operators can be used. If the designer wishes, in one criterion multiple different types of joints can be selected. The downside to this selection method is the requirement of listing every basis of the matroid, which can be very time consuming or even unfeasible.

3 INTRODUCTION TO SCREW THEORY AND DAVIES' METHOD

In this chapter a brief review of screw theory is presented. Then, Davies' method is explained and redundant constraint in the method is discussed.

3.1 SCREW THEORY

Screw theory as is known today was first introduced by Ball (1900) and has been extensively used in robotics and mechanisms (WALDRON, 1966; HUNT, 1978; DAVIDSON; HUNT, 2004; DAVIES, 2006b; KONG; GOSSELIN, 2007).

Screws can be used both in kinematic and static modelling of rigid bodies. For the kinematics, a screw represents the translation along a line coupled to a rotation around the same line, hence we can define linear and angular velocities of a rigid body in the spatial workspace using a screw.

A screw $\$$ can be defined as a line, the screw axis, with a scalar pitch h , with the pitch representing the ratio between the translational velocity and the angular velocity $h = ||\tau||/||\omega||$. The screw that represents a instantaneous motion of a rigid body is called a motion screw, which is used in kinematics. Using screw coordinates $\$^m = [r \ s \ t \ u \ v \ w]^T$, the instantaneous motion is represented by a linear component as the vector $\vec{V} = [V_x \ V_y \ V_z]^T = [u \ v \ w]^T$ and an angular component as the vector $\omega = [\omega_x \ \omega_y \ \omega_z]^T = [r \ s \ t]^T$, both relative to a fixed origin. A screw is also defined by Equation 8.

$$\$ = \begin{bmatrix} S \\ S_0 \times S + hS \end{bmatrix} \quad (8)$$

where S is the unit vector parallel to the screw axis, *i.e.* the direction of the screw. S_0 is a vector between any point of the screw axis and the fixed origin. For a screw with pitch $h = 0$, *i.e.* a pure rotation, the screw is represented by Equation 9.

$$\$ = \begin{bmatrix} S \\ S_0 \times S \end{bmatrix} \quad (9)$$

When the pitch $h \rightarrow \infty$, *i.e.* a pure translation, the screw will be represented by Equation 10.

$$\$ = \begin{bmatrix} 0 \\ S \end{bmatrix} \quad (10)$$

The static formulation using screw theory is similar to the kinematic formulation shown, with minor differences. For the static, screws are called action screws and they are used to model forces and moments acting on a rigid body. The pitch of the screw is calculated by the relation between the moment and the force components of the

screw. The screw coordinates are also different: $\$^a = [R \ S \ T \ U \ V \ W]^T$. The screw is a composition of a force vector $Q = [U \ V \ W]^T$ and a moment vector $P = [R \ S \ T]^T$. For the static modelling, we use Equation 11.

$$\$ = \begin{bmatrix} S_0 \times S + hS \\ S \end{bmatrix} \quad (11)$$

The screw used in static is inverted when considering the screw in kinematics, as for the prior is used in the axis-coordinates and the latter in the ray-coordinates (HUNT, 1978).

From the kinematics, Huang et al. (2013) discuss the concept of screw system. For a serial mechanism, the screw system is the composition of the motion screws of each link. The screw system can also be defined as the set of linearly independent screws that that can be used to describe the motion of a robot. A planar mechanism in the $x - y$ plane will present screws as combinations of t , u and v , while a spatial mechanism will present screws as combinations of r, s, t, u, v and w .

This work uses Davies' notation for constraints: $\$^a = [R \ S \ T \ U \ V \ W]^T$. Table 2 explains the Davies' notation.

Table 2 – Explanation of the notation used in this work.

Constraint	Equivalence
R	M_x - moment in the x-axis
S	M_y - moment in the y-axis
T	M_z - moment in the z-axis
U	F_x - force in the x-axis
V	F_y - force in the y-axis
W	F_z - force in the z-axis

Source – From the author.

Using Davies' constraint notation as well as the joints name, lists with many constraints are easier to distinguish, thus this combination is the preferred notation throughout this work.

In Table 3, three examples of joints are listed and their constraints are described. First, joint a is a revolute joint in the z-axis. Second, joint b is a universal joint in the x and z-axes. Finally, joint c is a cylindrical joint in the y-axis.

Table 3 – Comparison of Notations.

Joint	Type	M_x	M_y	M_z	F_x	F_y	F_z
a	revolute	a_R	a_S	–	a_U	a_V	a_W
b	universal	–	b_S	–	b_U	b_V	b_W
c	cylindrical	c_R	–	c_T	c_U	–	c_W

Source – From the author.

The constraints not present in the joints are noted using "-" in Table 3.

3.2 LINEAR DEPENDENCE OF FREEDOMS AND CONSTRAINTS

In this section, screw theory is used to evaluate the linear dependence of freedoms and constraints of joints of mechanisms. Joints are often called kinematic pairs, as they are the coupling of two bodies enabling some freedoms while imposing some constraints. Using screw theory, joints can be compared by the difference of the freedoms and constraints. Some examples are shown in Table 4.

Table 4 – Freedoms and Constraints Screws of Joints.

Joint	Freedoms	Constraints
Revolute joint in the z axis	$\$t$	$\$R, \$S, \$U, \$V, \$W$
Prismatic joint in the x axis	$\$u$	$\$R, \$S, \$T, \$V, \$W$
Universal joint in the x and y axis	$\$r, \s	$\$T, \$U, \$V, \W
Spherical joint	$\$r, \$s, \$t$	$\$U, \$V, \$W$
Pin in slot joint with a revolute freedom in the z axis and a translational freedom in the y axis	$\$t, \v	$\$R, \$S, \$U, \W
Cylindrical joint with a revolute freedom and a translational freedom in the y axis	$\$s, \v	$\$R, \$T, \$U, \W

Source – From the author.

The freedoms and constraints shown in Table 4 are based on linear independence of screws. For the revolute joint in the x axis, the screw axis, or the vector S parallel to the screw axis, is aligned with the x axis. By aligning the motion screw of the joint, the constraints can be found as shown in Table 4.

When the joints are not aligned with the x , y and z axis, the freedoms and constraints are not so easily defined. Using the revolute joints again as an example, the rotational freedom will have components in more than one axis, while maintaining only one degree of freedom. The vector S is still a unit vector, but now with components in multiple axes.

When modelling static and kinematics of robots and mechanisms, the linear dependence of screw axes are usually not a problem; however, when actually working with the linear independence of freedoms and constraints, the dependence of axes must be analyzed. Linear dependence between constraints can lead to singularities, which induces unwanted freedoms to the mechanism. Although using joints perfectly aligned with the coordinated axes facilitates the work of the designer, in real applications not always the axes are aligned and linear dependence is frequently difficult to see. Reshetov (1979), while presenting a method for enumerating self-aligning mechanisms based on linear independence, discussed about the approach required when facing joints not perfectly aligned with the coordinate axes. In his work, the mobilities of joints are related to one of the axis where the mobility is usable, *i.e.* there is a small angle between the joint axis and one of the coordinate axes. Therefore, when a joint is not aligned with the coordinate axes, the designer must choose which is the main independent axis. For example, a prismatic joint with an angle of 15° to the x- axis. This joint is considered a prismatic joint in the x-axis even if it is not aligned with that axis.

Although Reshetov (1979) does not use screw theory, another important aspect of linear dependence of freedoms is discussed. According to Reshetov (1979), a linear mobility can be replaced by an angular mobility perpendicular to the translational freedom. By rotating a link around an axis, a linear mobility freedom perpendicular to the rotating can be described.

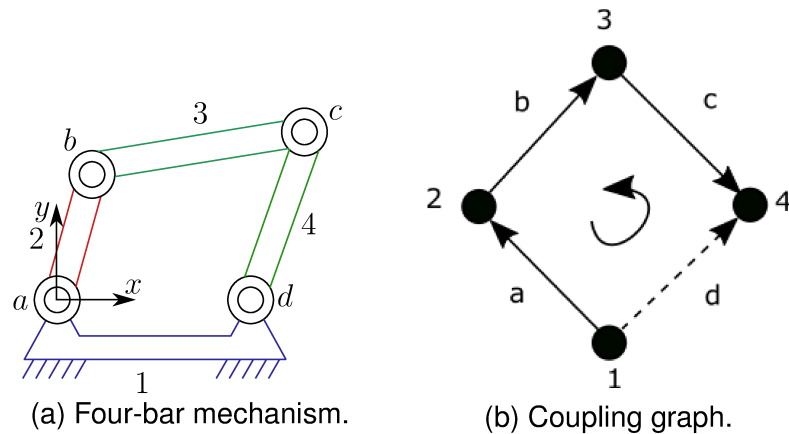
3.3 DAVIES' METHOD

The objective of this section is to briefly introduce the Davies' method. This method, proposed by Davies (DAVIES, 1981, 1995, 2006a, 2006b), consists of an adaptation of Kirchhoff's circulation and cutset laws to multibody systems. The Davies' method also uses screw theory and graph theory to model the freedoms and constraints of the joints. Several studies and applications for the Davies' method have been explored (LAUS et al., 2012; MEJIA et al., 2015; MORENO et al., 2018; TOSCANO et al., 2018).

3.3.1 Kinematic Model

The kinematic modelling in Davies' method starts by finding the coupling graph of the desired mechanism. Figure 22 shows a four-bar mechanism and its respective coupling graph G_C .

Figure 22 – Four-bar mechanism.



Source – From the author.

From the four-bar mechanism of Figure 22a, the unit motion matrix is created:

$$[M_D]_{6,4} = \begin{bmatrix} \$_{at}^m & \$_{bt}^m & \$_{ct}^m & \$_{dt}^m \end{bmatrix} \quad (12)$$

Each column of the matrix from Equation 12 represents the freedom from each joint, therefore this matrix has four columns. Although matrix $[M_D]$ has six rows, some of them can be removed. The mechanism is currently being modelled in the planar screw system, therefore the velocities not included in the planar system are always zero and can be removed. Thus, matrix $[M_D]$ has only three rows.

The vertices of the coupling graph G_C , Figure 22b, are the links of the mechanism while the edges represent the joints. The direction of the edges is assumed to go from vertices with lower identification number to vertices with higher number. In this example, there is only one circuit and edge d is chosen as the chord and thus dictates the direction of the circuit. With the circuit and its direction defined, the circuit matrix $[B_M]$ is created:

$$[B_M] = \begin{bmatrix} a_t & b_t & c_t & d_t \\ -1 & -1 & -1 & 1 \end{bmatrix} \quad (13)$$

A simplification is introduced in Equation 13, the freedoms of each joint are called by the joint name and mode of freedom, *i.e.* the revolute motion around the z -axis allowed by the joint a is now called a_t . In the circuit matrix $[B_M]$, if the edge is part of the loop it receives 1, if it is not the number input is 0. Furthermore, if the edge is in the same direction of the chord of the loop the sign is positive, otherwise it receives a negative sign. In a mechanism with multiple loops, the circuit matrix has one row for each loop.

Using the unit motion matrix (Equation 12) and the circuit matrix (Equation 13), the network unit motion matrix is created, as shown in Equation 14.

$$[M_N]_{\lambda\nu,F} = \begin{bmatrix} [M_D][\mathbf{B}_1] \\ [M_D][\mathbf{B}_2] \\ \vdots \\ [M_D][\mathbf{B}_k] \end{bmatrix} \quad (14)$$

where F is the gross degree of freedom of the system, ν is the number of loops and $[B_i]$ are diagonal matrices from each row i of the circuit matrix. The network unit motion matrix is used to create a linear homogeneous system, analogous to Kirchoff's circuit law:

$$[M_N][\phi]_F = [0]_{\lambda\nu} \quad (15)$$

where ϕ is a vector with the unknown magnitudes of velocities. For the four-bar mechanism of the example, the linear homogeneous system that represents the kinematic model is presented in Equation 16.

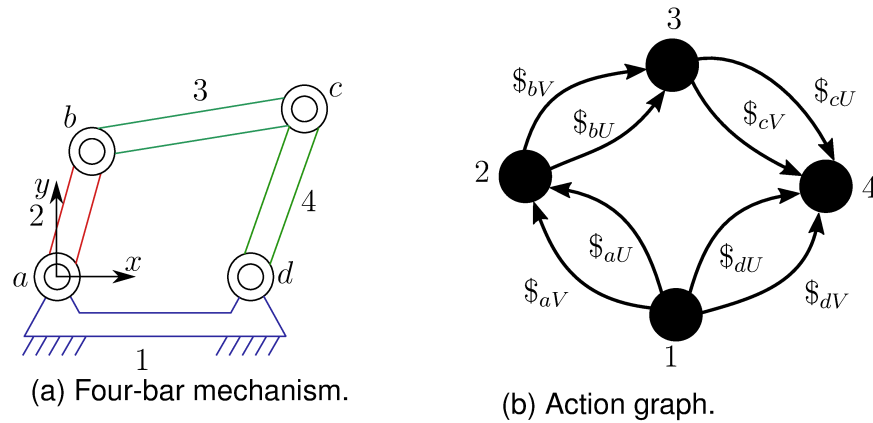
$$[M_D][B_1] \begin{bmatrix} a_t \\ b_t \\ c_t \\ d_t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (16)$$

By solving the linear system of Equation 16, the linear and angular velocities of each joint is calculated.

3.3.2 Static Model

We begin the demonstration of the static modelling by Davies' method with a four-bar mechanism, which is shown in Figure 23.

Figure 23 – Four-bar mechanism



Source – From the author.

The action graph of Figure 23 has two edges between each pair of vertices, as each revolute joint has one freedom and two constraints in the planar screw system. These constraints are arranged in action screws $\$_{i,j}^a$, each action screw related to the i coupling and the j mode of constraint. In the notation used in Davies' method, the modes of constraints are R, S and T for moment constraints around x, y and z axis, respectively, while modes U, V and W are regarding the linear constraints along the x, y and z axis. For the planar screw system, we use only the constraints T, U and V .

The action screws for the joints of the mechanism from Figure 23 are organized in the unit action matrix $[A_D]$, with each constraint as a column. This mechanism was modelled in the planar screw system.

$$[A_D]_{6,8} = [\$_{aU}^a \quad \$_{aV}^a \quad \$_{bU}^a \quad \$_{bV}^a \quad \$_{cU}^a \quad \$_{cV}^a \quad \$_{dU}^a \quad \$_{dV}^a] \quad (17)$$

Equation 17 shows all of the constraints and the respective joints of the mechanism shown in Figure 23b, which means that this matrix has eight columns. The matrix $[A_D]$ has all of the constraints in the system, however it lacks the relations between those constraints. These relations are defined using Kirchoff's cutset law applied to the action graph of the mechanism. In the action graph, shown in Figure 23b, each node represents a joint and each edge represents a constraint. In Figure 23b, the edges from joint d were chosen as chords (DAVIES, 2006b). Applying Kirchoff's cutset law to the action graph of Figure 23b, the cutset matrix $[Q_A]$ is created. Equation 18 shows the cutset matrix from graph of Figure 23b.

$$[Q_A] = \begin{bmatrix} a_U & a_V & b_U & b_V & c_U & c_V & d_U & d_V \\ \left[\begin{array}{cc|cc|cc|cc} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right] \end{bmatrix} \quad (18)$$

The columns of the $[Q_A]$ matrix are the edges of the graphs which represent each of the constraints in the coupling and each row represents a cutset. If a constraint is contained in a cutset, the element receives the number 1, otherwise the number zero is input. Furthermore, if the edge is in the same direction of the chord, then the sign is positive; otherwise, the element receives a negative sign. For a simplification, the constraints $\$_{i,j}^a$ are now called only by the joint and mode of constraint, e.g. a_U .

The matrices $[A_D]$ and $[Q_A]$ are now used to create the network unit action matrix $[A_N]$. This matrix is presented in Equation 19.

$$[A_N]_{\lambda k, C} = \begin{bmatrix} [A_D][Q_1] \\ [A_D][Q_2] \\ \vdots \\ [A_D][Q_k] \end{bmatrix} \quad (19)$$

In Equation 19, $[Q_i]$, $i = 1, 2, \dots, k$, are diagonal matrices whose elements correspond to the row i of the cutset matrix $[Q_A]$, in this system $k = 3$. The matrix $[A_N]$, as well as the matrix $[A_D]$, has eight columns ($C = 8$), as there are four joints with two constraints each. If we add external forces or actuation forces, these would be entered as additional constraints and would increase the number of columns of matrix $[A_N]$. Using the $[A_N]$, every force or moment of every joint is calculated using the fundamental cutset law:

$$[A_N][\psi]_C = [0]_{\lambda k} \quad (20)$$

where ψ is the vector with unknown magnitudes of the action screws imposed by the joints. Using Equation 20 as a linear homogeneous, system the static model of the mechanism can be solved, as shown in Equation 21.

$$\begin{bmatrix} [A_D][Q_1] \\ [A_D][Q_2] \\ [A_D][Q_3] \end{bmatrix} \begin{bmatrix} a_U \\ a_V \\ b_U \\ b_V \\ c_U \\ c_V \\ d_U \\ d_V \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (21)$$

Matrix $[A_N]$ has λk rows and C columns. For the planar four-bar mechanism, the screw system order is $\lambda = 3$ and from the action graph of Figure 23b three cutsets $k = 3$ exist, hence matrix $[A_N]$ has 9 rows. It was also shown that for the four-bar mechanism, matrix $[A_N]$ has 8 columns. Actuation forces can be used as constraints to complete the static model of the four-bar mechanism, achieving a matrix $[A_N]$ with

9 rows and 9 columns. This step can be done because adding an actuation constraint does not change the number of cutsets nor the screw system order. Furthermore, a known exterior force imposed on the mechanism can be used as the primary variable while the remaining 9 constraints (2 for each joint plus the actuation) are considered secondary variables. If the actuation force is known and the designer wishes to calculate the maximum external force supported by the mechanism, the actuation is defined as the primary variable and the external force becomes a secondary variable. If matrix $[A_N]$ is a square matrix with full rank, it does not have redundant constraints and is statically determinate.

4 RESHETOV VIRTUAL JOINTS

In this Chapter, the novel concept of Reshetov virtual joints is introduced. This new concept is presented using Baranov kinematic chains. Later, new gripper mechanisms are created using Reshetov virtual joints, Baranov kinematic chains and Davies' method.

The static model of the mechanism according to Davies' method is used herein; however, the kinematic model can also be used when evaluating redundant constraints. After creating the static model of a mechanism, the linear homogeneous system is solved using the matrix $[A_N]$. In case the mechanism is modelled and the number of columns of matrix $[A_N]$ is bigger than the rank of the same matrix, there are redundant constraints present in the mechanism. Redundant constraints in mechanisms pose some problems. For example, redundant constraints may hinder the assembly of mechanisms when there are manufacturing problems, an issue that mechanisms without redundant constraints do not have. Mechanisms without redundant constraints are called self-aligning mechanisms. Previous work already discussed redundant constraints in Davies' method: Carboni (2015), Carboni et al. (2017), Barreto et al. (2018), Artmann et al. (2019a).

The number of redundant constraints q is calculated using Equation 22.

$$q = C - \text{rank}\{[A_N]\} \quad (22)$$

where C is the number of constraints present in the mechanism.

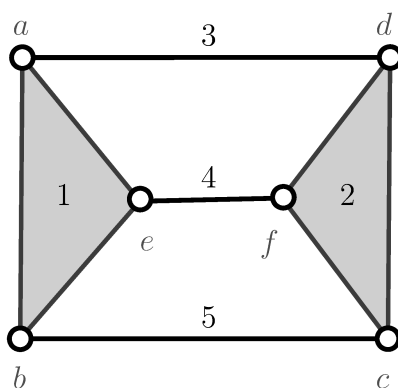
Redundant constraints can also be evaluated using the kinematic modelling by Davies' method as proposed by Davies (2006b), shown in Equation 23 for multi-loop mechanical networks:

$$q = \lambda\nu - \text{rank}\{[M_N]\} \quad (23)$$

and in Equation 24 for single loops mechanisms:

$$q = \lambda\nu - \text{rank}\{[M_D]\} \quad (24)$$

Now, Baranov kinematic chains will be used to demonstrate the relevant concepts of the chapter. Baranov kinematic chains are the chains that present mobility $M = 0$ while every independent circuit present mobility $M > 0$. The Baranov chain from Figure 24 will be used as example for the Davies' method modelling.

Figure 24 – Seed Baranov chain with $n = 5$ and $j = 6$.

Source – From the author.

This Baranov chain has $n = 5$ links and $j = 6$ joints. Using a well known formulation for the mobility of mechanisms is shown in Equation 25 (GOGU, 2005c):

$$M = \lambda(n - 1) - C + q \quad (25)$$

where C is the number of independent constraints and q is the number of redundant constraints. A mechanism constituted by the Baranov chain of Figure 24 has mobility $M = 0$ and $q = 0$. Furthermore, in the planar screw system $\lambda = 3$ the mechanism must have twelve constraints. It is important to highlight that Equation 25 is not necessarily valid for the full cycle mobility, *i.e.*, the mobility may change if the mechanism is in a singular position. For more information on full cycle mobility, please refer to (CARRICATO; ZLATANOV, 2014). This work does not focus on mechanisms in critical or singular conditions.

Now a novel concept called Reshetov virtual joints is introduced. The inspiration behind the development of the Reshetov virtual joints was the work by (RESHETOV, 1979), thus this new concept was named after him. The couplings used in the network unit action matrix normally refers to joints, *e.g.* revolute or prismatic joints; however, in this work the concept of Reshetov virtual joints is used. Reshetov virtual joints are joints without freedoms, *i.e.* formed only by λ independent constraints. In a planar system $\lambda = 3$, a Reshetov virtual joint has three constraints, while prismatic or revolute joints have two constraints and a single freedom. By removing selected constraints, the Reshetov virtual joint can be transformed into any desired joint. Table 5 shows some examples of joints derived from a Reshetov virtual joint in the spatial screw system.

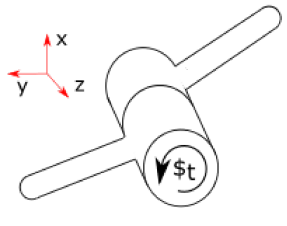
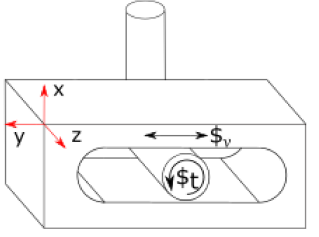
Table 5 – Examples of joints derived from a Reshetov virtual joint.

Removed constraints	Remaining constraints	Derived joint
$\$T$	$\$R, \$S, \$U, \$V, \$W$	Revolute joint in the z axis
$\$V$	$\$R, \$S, \$T, \$U, \$W$	Prismatic joint in the y axis
$\$R, \$S, \$T$	$\$U, \$V, \$W$	Spherical joint
$\$T, \U	$\$R, \$S, \$V, \W	Pin in slot joint with a revolute freedom around the z axis and a translation in the x axis
none	$\$R, \$S, \$T, \$U, \$V, \W	The Reshetov virtual joint is integrated into the link

Source – From the author.

Figure 25 shows a revolute joint and a pin-in-slot joint. Each joint is presented with the relative freedoms (indicated by the motion screws) while the constraints (expressed by the action screws) are listed to the right.

Figure 25 – Example of joints and the remaining constraints.

Joint	Representation	Constraints
Revolute		$\$R$ $\$S$ $\$U$ $\$V$ $\$W$
Pin-in-slot		$\$R$ $\$S$ $\$U$ $\$W$

Source – From the author.

When modelling a mechanism with Davies' method, Reshetov virtual joints can be introduced as means to add redundant constraints to the system; however, Reshetov virtual joints are not required as there are mechanisms that already contain redundant constraints.

A concept proposed by Reshetov (1979) that was presented in Section 3.2 stated that for joints not completely aligned with the coordinated axes, the designer chooses the axis with the smallest angle between the coordinated axis and the joint axis. Take for instance a prismatic joint with 15° to the x -axis. According to the concept proposed by Reshetov, this prismatic joint is considered a prismatic joint in the x -axis. Applying

this concept for Reshetov virtual joints, when constraints are removed the designer is not obliged to align the joint with the desired axis; instead, the designer can make adjustments to the joint axis as long as that axis is still closer to the initial coordinated axis than the other axes. For example, a Reshetov virtual joint is given a translational freedom along the x -axis. This does not mean that the derived joint must always remain aligned with the x -axis, but actually this joint is only considered with the smallest angle between the joint axis and the x -axis than the y or z axes. This concept is true not only for Reshetov virtual joints, but for regular joints as well.

Using the novel concept of Reshetov virtual joints, a mechanism constituted solely by these joints is created, with the corresponding $[A_D]$ matrix presented in Equation 26 for the Baranov chain from Figure 24. The subscripts T , U and V represents the constraints in the planar screw system, a moment around the z -axis (T) and forces in the x and y -axes, U and V respectively.

$$[A_D]_{6,52} = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc} \text{Coupling a} & & & \text{Coupling b} & & & \text{Coupling c} & & & \text{Coupling d} & & & \text{Coupling e} & & & \text{Coupling f} \\ a_T & a_U & a_V & b_T & b_U & b_V & c_T & c_U & c_V & d_T & d_U & d_V & e_T & e_U & e_V & f_T & f_U & f_V \end{array} \right] \quad (26)$$

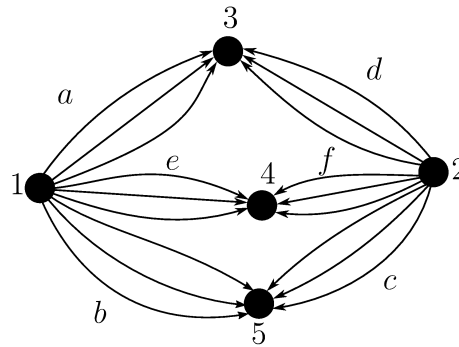
Table 6 presents the correlation between the constraints T , U and V with the forces and moment present in the Reshetov virtual joint.

Table 6 – Examples of Couplings and the Constraints Correlations.

Coupling	M_z	F_x	F_y
a	a_T	a_U	a_V
b	b_T	b_U	b_V
c	c_T	c_U	c_V

Source – From the author.

Figure 26 presents the action graph for the seed Baranov chain. There are three edges representing each joint because every joint has three constraints, each of these three constraints represent one of the constraints for the coupling from Equation 26.

Figure 26 – Action Graph for the Seed Baranov Chain with $n = 5$ and $j = 6$.

Source – From the author.

Using the $[A_D]$ matrix from Equation 26 and the action graph from Figure 26 we create the network unit action matrix $[A_N]$ according to Davies' method. When modelling a mechanism using only Reshetov virtual joints, the number of columns of the network unit action matrix $[A_N]$ is:

$$C = \lambda j \quad (27)$$

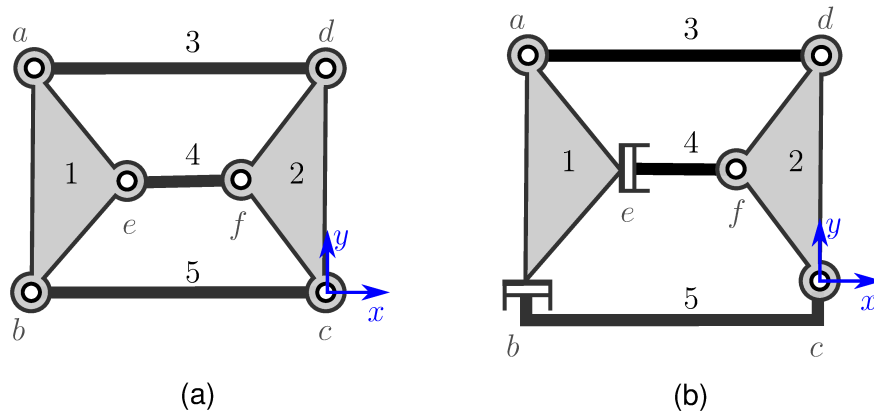
Considering a mechanism from the chain shown in Figure 24, the matrix $[A_N]$ will have eighteen columns. Recalling Equation 25, a mechanism with six joints, mobility $M = 0$ in the planar screw system must have twelve constraints. Comparing matrix $[A_N]$ with eighteen columns and the required number of constraints equal to twelve, we must remove six columns from the network action matrix to achieve the required number of constraints in the mechanism considering a Baranov chain. Thus, we derive an equation that relates the number of constraints present in the $[A_N]$ matrix to the number q of constraints that must be eliminated, shown in Equation 28.

$$q = C - [\lambda(n - 1) - M] \quad (28)$$

Recall that according to Davies' method the matrix $[A_N]$ must be square with full rank to avoid redundant constraints. Let q redundant constraints be the number of columns to be removed from the $[A_N]$ matrix of the mechanism constituted by Reshetov virtual joints to make it a square matrix.

Suppose we remove the moment constraints T in the z -axis from each of the joints of the chain of Figure 24. We will have a mechanism formed only by revolute joints shown in Figure 27a. Now suppose we remove the force constraint U in the x -axis from joint e , the force constraint V in the y -axis from joint b and the T constraints from the remaining joints, we now have a different mechanism, which is shown in Figure 27b.

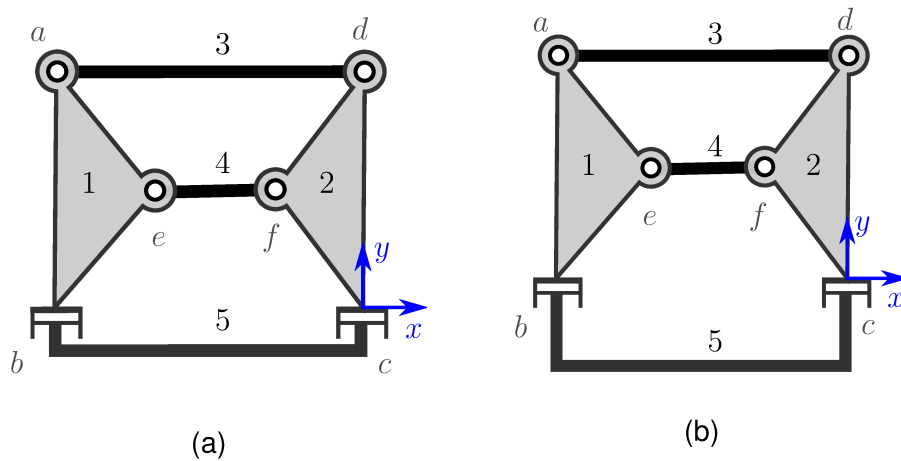
Figure 27 – Two different mechanisms generated using the proposed method.



Source – From the author.

By removing columns from the $[A_N]$ matrix arranged using Reshetov virtual joints, we can create different mechanisms; however, columns cannot be removed randomly. When removing columns, the rank of the $[A_N]$ matrix must always remain unaltered. Take for example the mechanisms from Figure 28a and Figure 28b.

Figure 28 – Mechanism with dangerous mobility.



Source – From the author.

The force constraints V in the y -axis were removed from joints b and c . Link 5 has a freedom regardless of the movements of the rest of the mechanism. Usually, this type of freedom is not desired. Moreover, if we remove the mentioned force constraints from the matrix $[A_N]$ from Equation 26, the rank of the matrix would decrease. The decrease is due to the linear dependence between the remaining constraints. Thus, we create a method to design mechanisms starting with Baranov chains by removing columns of the network unit action matrix $[A_N]$ until there are no more redundant constraints

present in that matrix. After each column removal, the rank of the $[A_N]$ matrix must be checked to verify if the rank of that matrix was affected by the column removal.

Although the method proposed in this chapter appears to be a complicated manner of choosing joint types of mechanisms, there are some important benefits. When the joint types are defined using the proposed methodology, the designer has a mathematical guarantee that there are no redundant constraints or dangerous mobilities present in the mechanism in the position analyzed. Another benefit of the proposed method is on the static model of the mechanism. After eliminating the redundant constraints present in the network unit action matrix $[A_N]$, the resulting matrix is the static model for the mechanism, which can be used to solve Equation 20. In the next section, the proposed method will be applied into the development of new gripper mechanisms.

4.1 NEW GRIPPER MECHANISMS

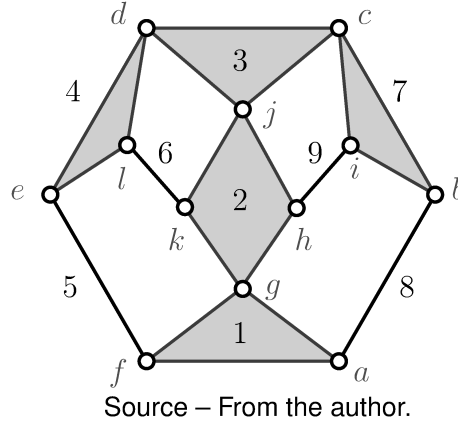
This section presents the development of new gripper mechanisms using the method presented in this Chapter. Grippers are mechanical systems designed to grasp and release objects by converting the actuator's motion into a gripping action. There are multiple ways to classify gripper mechanisms. A classification may rely, for example, on the type of the kinematic pairs, since gears, cams, screws, revolute pairs, prismatic pairs and belt-pulleys have been widely applied in the design of industrial gripping devices (CHEN, 1982).

During the development of a gripper mechanism, it is convenient for the designer to use a systematic approach to select the type of kinematic pairs, instead of randomly assembling mechanical parts. Several researchers attempted to systematize this task. In the mid-1980s, Erdman et al. (1986) outlined the importance of properly specifying gripper joint types by suggesting a type synthesis technique for gripper mechanisms based on graph theory and expert systems. The technique was applied in two examples of gripper implementations, which are derived from the Stephenson and Watt six-bar linkage. Later, Belfiore and Pennestri (1997) also proposed an algorithm for systematic enumeration of grippers. The approach proceeds in accordance with a set of prescribed structural and functional requirements, such as allowing only revolute and prismatic joints. Thus, an atlas of 64 possible topologies of 1-DoF robotic grippers with up to six links is provided. A more recent work by Xu et al. (2012) includes the type synthesis of spatial forging manipulators based on the screw theory. The systematic method was used to synthesize several configurations of non-overconstrained mechanisms composed of a gripper connected to a moving platform, called the gripper-support.

Grippers are mechanisms whose objective is to hold objects. When using Baranov chains for designing grippers, the object is considered as one of the links. As long as the object is being held by the gripper, the zero mobility is maintained; however, when the object is removed, the gripper mechanism will then possess mobility $M > 0$.

For demonstrating the method proposed herein, we choose the seed Baranov chain of Figure 29.

Figure 29 – Seed Baranov chain with $n = 9$ and $j = 12$.



Now we use Reshetov virtual joints to model the system. The unit action matrix $[A_D]$ is:

$$\begin{aligned}
 [A_D]_{6,36} = & \left[\begin{array}{cccccc}
 \overbrace{a_T \ a_U \ a_V}^{\text{Coupling a}} & \overbrace{b_T \ b_U \ b_V}^{\text{Coupling b}} & \overbrace{c_T \ c_U \ c_V}^{\text{Coupling c}} & \overbrace{d_T \ d_U \ d_V}^{\text{Coupling d}} & \overbrace{e_T \ e_U \ e_V}^{\text{Coupling e}} & \overbrace{f_T \ f_U \ f_V \dots}^{\text{Coupling f}} \\
 \dots & \overbrace{g_T \ g_U \ g_V}^{\text{Coupling g}} & \overbrace{h_T \ h_U \ h_V}^{\text{Coupling h}} & \overbrace{i_T \ i_U \ i_V}^{\text{Coupling i}} & \overbrace{j_T \ j_U \ j_V}^{\text{Coupling j}} & \overbrace{k_T \ k_U \ k_V}^{\text{Coupling k}} & \overbrace{l_T \ l_U \ l_V}^{\text{Coupling l}}
 \end{array} \right] \quad (29)
 \end{aligned}$$

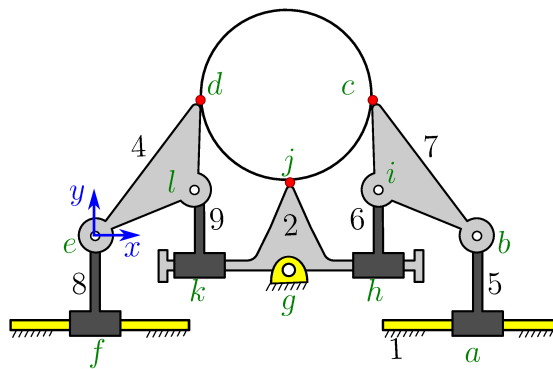
The positions for the joints used in Equation 29 and the cutset matrix are shown in Appendix A.1.

The network unit action matrix $[A_N]$ for the chain of Figure 29 formed by Reshetov virtual joints has rank $r = 24$ with $C = 36$ columns. Therefore, we must remove twelve columns and thus create new mechanisms. The contact point between the object and the finger-tip is considered a point contact with friction. This type of contact can then be considered as a revolute joint between the link representing the object and the links representing the fingers of the gripper. Thus, the joints of the contacts are defined as revolute and the moment constraint T around the z-axis must be removed from the matrix $[A_N]$.

For the first mechanisms, we consider link 3 as the object. Joints c, d and j are the contacts between the object and the finger-tips, therefore these joints are considered revolute joints and their respective moment constraints T are removed from the matrix $[A_N]$. Considering a gripper mechanism controlled by linear actuators, we considered joints a, f, h and k as prismatic joints in the x-axis. Thus, the force constraint U in the x-axis are removed from the matrix $[A_N]$. We have already removed seven columns and so far the rank of the network action matrix remains unaltered, thus we can continue. We have still five joints to decide. The remaining joints b, e, g, i and l are considered revolute

joints, therefore once again we must remove the moment constraints T respective to these joints. We finally obtain a network action matrix $[A_N]$ with twenty four columns and rank $r = 24$, which means that according to Equation 22 when the object is held the mechanisms has $q = 0$ redundant constraints and mobility $M = 0$, as we desired. The resulting mechanism is shown in Figure 30.

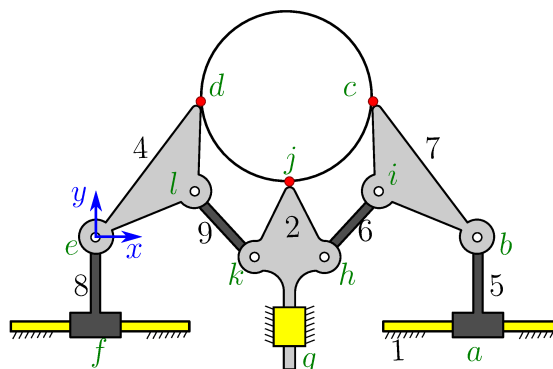
Figure 30 – First potential scheme for robot grippers based on the nine-link Baranov chain.



Source – From the author.

Another example is shown in Figure 31. In this gripper mechanism, the joints c , d and j were chosen as contact points and thus were defined as revolute joints. Joints b , e , h , i , k and l were also defined as revolute joints; therefore, the constraint T was removed from each of these joints. Joints a and f were chosen as prismatic joints in the x-axis while joint g was chosen as a prismatic joint in the y-axis. Thus, constraints U were removed from joints a and f , at the same time constraint V was removed from joint g .

Figure 31 – Second potential scheme for robot grippers based on the nine-link Baranov chain.

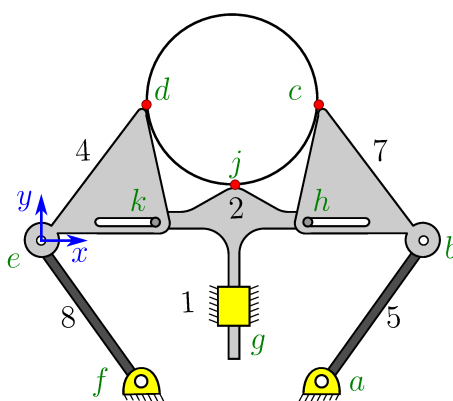


Source – From the author.

One interesting characteristic of the proposed method is shown in Figure 32.

Links 6 and 9 as well as joints i and j are no longer part of the mechanism. Joint i is contracted into joint h while joint l is integrated into joint k . This contraction means joint i and link 6 were integrated into link 7, while link 9 and joint l were integrated into link 4. As a result, joints h and k received one additional freedom each, both becoming pin-in-slot joints. In this example, pin-in-slot joints were chosen, but any type of joint can be selected, as long as the constraints present in that joint type are contained in the screw system of the modelled mechanism. For example, only joints that can be used in the planar screw system can be chosen when modelling a mechanism in the planar screw system.

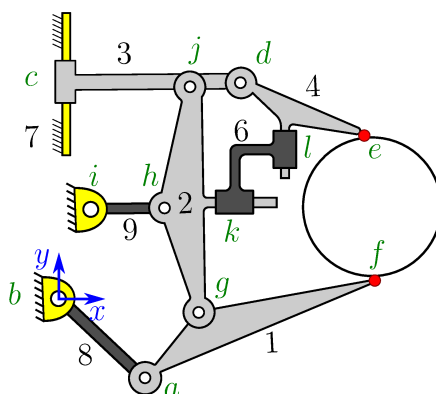
Figure 32 – Third potential scheme for robot grippers based on the nine-link Baranov chain.



Source – From the author.

In Figure 33, a Baranov gripper mechanism with two fingers is shown. Joints e and f are thus defined as the contact point, requiring us to remove the moment constraints T from the matrix $[A_N]$. Joints c , k and l are defined as prismatic joints, so we remove the force constraints V from joints c and l while removing constraints U from joint k . The remaining joints are defined as revolute joints so we remove the moment constraint T from those joints.

Figure 33 – Fourth potential scheme for robot grippers based on the nine-link Baranov chain.



Source – From the author.

The method presented in this Chapter used Reshetov virtual joints to create mechanisms with redundant constraints while modelling the mechanisms using Davies' method. Columns of matrices were removed manually in order to create new self-aligning Baranov gripper mechanisms. In the next chapters, matroid theory is introduced then used to enumerate self-aligning mechanisms without requiring to manually delete columns from the network unit action matrices.

5 BRIEF REVIEW OF MATROID THEORY

The objective of this sections is to introduce and explain the properties and characteristics of matroids relevant to this work. Matroids were first introduced by Whitney (WHITNEY, H., 1992) as an abstraction on linear independence. There are several different types of matroids, such as graph matroid, linear matroid and transversal matroids. These matroids, and other formulations, are available in the literature: Murota (2009), Neel and Neudauer (2009) and Oxley (2006).

A matroid is formed by a finite set E and a collection of subsets \mathcal{I} of E , having the following properties (RECSKI, 2013):

- (1) $\emptyset \in \mathcal{I}$ - the empty set belongs to \mathcal{I}
- (2) $J \in \mathcal{I}$, and $I \subseteq J \Rightarrow I \in \mathcal{I}$ - a subset I , contained in the subset J , is contained in the collection of subsets \mathcal{I} if the subset J is contained in \mathcal{I}
- (3) $I, J \in \mathcal{I}$, $|I| < |J| \Rightarrow (I \cup \{v\}) \in \mathcal{I}$ for some $v \in J \setminus I$ - if a subset I is smaller than the subset J , we can find an element from J that does not belong to I so that the union of this element to I is contained in \mathcal{I}

For this work, Property (2) is the most relevant, requiring that all subsets of an independent set are also independent. We are looking for the independent sets of a given matroid, so let us compare graphic and linear matroids to understand what the independent sets are.

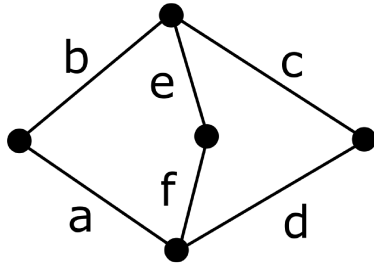
5.1 COMPARISON OF GRAPHIC AND LINEAR MATROIDS

Here we present briefly graphic and linear matroids.

Graphic Matroids: Graphic matroid is a type of matroid formed by the independent sets composed of the forest in a given undirected graph. Consider the following graph G :

Linear Matroids: Linear matroid is a type of matroid whose independent sets are composed of columns of a matrix. Consider the following matrix $[M]$:

Figure 34 – Graph example G .



Source – From the author.

$$[M]_{4,6} = \begin{matrix} & m_1 & m_2 & m_3 & m_4 & m_5 & m_6 \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{matrix} \quad (30)$$

The edges of graph G constitute the groundset $E_G = a, b, c, d, e, f$ of the matroid $\mathcal{M}(G)$. A collection of subsets \mathcal{J} of E_G is a base of matroid $\mathcal{M}(G)$ if the subset \mathcal{J} contains no circuits.

The bases from the matroid $\mathcal{M}(G)$ are: $\{a, b, c, e\}$, $\{a, b, c, f\}$, $\{a, b, d, e\}$, $\{a, b, d, f\}$, $\{a, c, d, e\}$, $\{a, c, d, f\}$, $\{a, c, e, f\}$, $\{a, d, e, f\}$, $\{b, c, d, e\}$, $\{b, c, d, f\}$, $\{b, c, e, f\}$ and $\{b, d, e, f\}$.

Each basis forms a different spanning tree of the graph G .

Comparing the graphic matroid $\mathcal{M}(G)$ and the linear matroid $\mathcal{M}(M)$ it is concluded that these matroids are isomorphic, *i.e.*, different matroids generated from different structures have the same properties. Another important characteristic of matroids is the dual bases. Every matroid has a dual matroid \mathcal{M}^* .

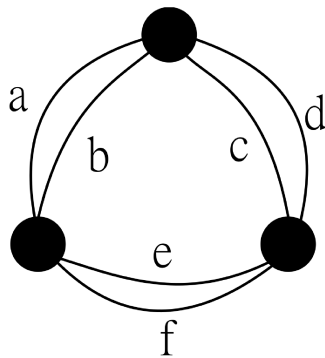
Graphic Matroids: The dual matroid $\mathcal{M}^*(G)$ can be found from the dual graph G^* (OXLEY, 2006). A graph G_1 is dual to a graph G_2 if the circuits of G_1 are cutsets of G_2 and vice versa (RECKSKI, 2013). Figure 35 presents the dual graph G^* . The bases from the dual matroid $\mathcal{M}^*(G)$ are called the cobases from the matroid $\mathcal{M}(G)$.

The columns of matrix $[M]_{3,6}$ are indexed by $E = Col(M) = m_1, m_2, m_3, m_4, m_5, m_6$. We use these indexes when analyzing the linear dependence of the columns. A subset of columns $I \subseteq E$ are independent if the vectors formed by the columns of the subset I are linearly independent. The family of all independent subsets is denoted by \mathcal{I} .

The bases from the matroid $\mathcal{M}(M)$ are: $\{m_1, m_2, m_3, m_5\}$, $\{m_1, m_2, m_3, m_6\}$, $\{m_1, m_2, m_4, m_5\}$, $\{m_1, m_2, m_4, m_6\}$, $\{m_1, m_3, m_4, m_5\}$, $\{m_1, m_3, m_4, m_6\}$, $\{m_1, m_3, m_5, m_6\}$, $\{m_1, m_4, m_5, m_6\}$, $\{m_2, m_3, m_4, m_5\}$, $\{m_2, m_3, m_4, m_6\}$, $\{m_2, m_3, m_5, m_6\}$ and $\{m_2, m_4, m_5, m_6\}$.

Linear Matroids: Each basis from the matroid $\mathcal{M}(M)$ has a complement of elements such that $X = E - Y$, where Y is the basis, E is the groundset and X is the group of elements that belong to the groundset but do not belong to the basis Y ; therefore, each basis from the matroid $\mathcal{M}(M)$ has a cobasis from the same matroid. Equation 31 represents a dual matroid $\mathcal{M}^*(M)$. Matrix $[M^*]$ is

Figure 35 – Dual graph G^* .



Source – From the author.

not equivalent to matrix $[M]$, but instead $[M^*]$ is the dual of matrix $[M]$. Each cobasis from $\mathcal{M}(M)$ is a basis from the dual matroid $\mathcal{M}^*(M)$.

$$[M^*]_{2,6} = \begin{matrix} & m_1 & m_2 & m_3 & m_4 & m_5 & m_6 \\ \begin{bmatrix} 1 & 0 & -1 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 & -1 & -1 \end{bmatrix} \end{matrix}$$

(31)

The cobases from the matroid $\mathcal{M}(G)$ (or instead the bases of matroid $\mathcal{M}(G^*)$) are: $\{d, f\}$, $\{d, e\}$, $\{c, f\}$, $\{c, e\}$, $\{b, f\}$, $\{b, e\}$, $\{b, d\}$, $\{b, c\}$, $\{a, f\}$, $\{a, e\}$, $\{a, d\}$ and $\{a, c\}$.

The matroid $\mathcal{M}(G)$ has rank $r = 4$ while the dual matroid $\mathcal{M}^*(G)$ has rank $r = 2$.

The cobases from the matroid $\mathcal{M}(M)$ (or instead the bases from matroid $\mathcal{M}(M^*)$) are: $\{m_4, m_6\}$, $\{m_4, m_5\}$, $\{m_3, m_6\}$, $\{m_3, m_5\}$, $\{m_2, m_6\}$, $\{m_2, m_5\}$, $\{m_2, m_4\}$, $\{m_2, m_3\}$, $\{m_1, m_6\}$, $\{m_1, m_5\}$, $\{m_1, m_4\}$ and $\{m_1, m_3\}$.

The matroid $\mathcal{M}(M)$ has rank $r = 4$ while the dual matroid $\mathcal{M}^*(M)$ has rank $r = 2$.

Table 7 presents a summary comparing linear and graphic matroids.

Table 7 – Comparison between linear and graphic matroids.

Matroid \mathcal{M}	Ground set E	Circuits \mathcal{C}	Independent sets \mathcal{I}	Bases \mathcal{B}
$\mathcal{M}(G)$ graphic matroid of graph G	$E(G)$, edge-set of G	edge-sets of cycles	$\{I \subseteq E(G) : I \text{ contains no cycle}\}$	For connected G : edge-sets of spanning trees
$\mathcal{M}(M)$ linear matroid of a matrix M over field F	column labels of M	minimal linearly independent multi-sets of columns	$\{I \subseteq E : I \text{ labels a linearly independent multiset of columns}\}$	Maximal linearly independent sets of columns

Source – Adapted from (GROSS et al., 2013).

5.2 COMBINATORICS PROBLEM

Given a linear matroid $\mathcal{M} = (E, \mathcal{I})$ with groundset E and independent subsets \mathcal{I} , the number of independent subsets \mathcal{I} is denoted by $|\mathcal{I}| \subseteq 2^E$. This means that increasing the groundset of the matroid, the number of independent subsets tends to

increase exponentially which can be a nuisance when dealing with complex systems. Algorithms for enumerating bases of matroids run in polynomial time $\text{poly}(x)$, where x is the cardinality of the ground set (UNO, 1999; KHACHIYAN et al., 2005).

We can evaluate this problem further using the combination equation (Equation 30):

$$\binom{k}{b} = \frac{k!}{b!(k-b)!} \quad (30)$$

This equation is used to determine the number of combinations of b sized elements from the total k elements where the order of the b elements does not matter. Using Equation 30 we can calculate the theoretical maximum possible number of bases of a linear matroid. For example, if a matrix has six columns and rank four, the number of combinations would be fifteen, which means that a linear matroid created from this matrix can have a maximum of fifteen independent bases. The example used in the previous section (Eq. 53) has six columns and rank four and a total of twelve bases. Although Equation 30 does not enable us to calculate the number of bases, it sets an upper bound. For instance, a matrix with twelve columns and rank nine will present 220 possible bases. Table 8 presents some examples of maximum possible bases for matrices with different number of columns and rank.

Table 8 – Comparison of Columns, Rank and Maximum Possible Bases for Different Matrices.

Columns	Rank	Possible Bases
6	4	15
6	3	20
12	9	220
12	7	792
18	15	816
18	12	18564
22	19	1540
22	17	170544

Source – From the author.

From Table 8 it is evident that increasing the size of matrices and the difference between the rank and the number of columns has a considerable impact in the number of possible bases of a matroid from those matrices.

When we create a linear matroid using the network unit action matrix $[A_N]$, each basis from this matroid represents the set of constraints that constitute a self-aligning mechanism. If we increase the size of the matrices and the difference between the number of columns and the rank of that matrix, the number of possible bases for the linear matroid increases considerably. This is a problem when using matroids to enumer-

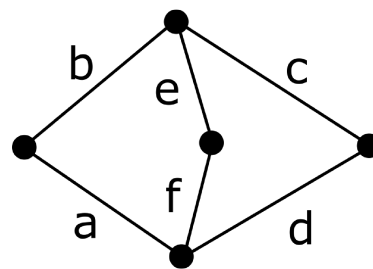
ate mechanisms using Davies' method for two reasons: firstly, more complex systems means that more computation time is needed; secondly, more possible solutions makes choosing one optimal solution hard.

In Barreto et al. (2018) the bases were filtered manually, in Carboni et al. (2017) the bases were selected using the greedy algorithm and in Artmann et al. (2019a) the bases were filtered using design requirements applied to a matrix called Cobases Binary Matrix constituted by the set of cobases. In this work, we use contraction and deletion of matroids to deal with the combinatorics problem of the number of bases increasing exponentially with the increase of the complexity of the mechanisms modelled.

5.3 CONTRACTION ($/$) AND DELETION (\setminus) OF MATROIDS

The formulation for both contraction and deletion operations is herein briefly introduced. Every calculation and comparison presented herein is done using the open source software *SageMath*, but could also be done manually. The graph operations will use the example graph G , shown again in Figure 36.

Figure 36 – Graph example G .



Source – From the author.

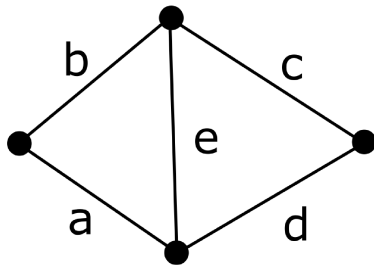
Recski (2013) defines the operation called contraction by means of the rank function of matroids. For a start, take r as the rank function of the matroid $\mathcal{M}=(E, \mathcal{B})$ and consider a subset $X \subseteq E$. The rank function R of every subset Y of the matroid \mathcal{M} is defined by Equation 31.

$$R(Y) = r(X \cup Y) - r(X) \quad (31)$$

Equation 31 is valid for any subset Y of the set $E - X$. The new matroid with rank function $R(Y)$ on the set $E - X$ is denoted by \mathcal{M}/X and is known as the contraction of \mathcal{M} to $E - X$.

Graph contraction: For the graphic matroids we will show the contraction by means of the graph. Let us contract the edge f : this step consists in merging two vertices, thus removing the desired edge from the graph. We will attain the graph G_1 :

Figure 37 – Graph G_1 - Graph G after a contraction of edge f .



Source – From the author.

The bases of the matroid $\mathcal{M}(G_1)$ are: $\{a, b, c\}$, $\{a, b, d\}$, $\{a, c, d\}$, $\{a, c, e\}$, $\{a, d, e\}$, $\{b, c, e\}$, $\{b, c, d\}$ and $\{b, d, e\}$.

The cobases of the matroid $\mathcal{M}(G_1)$ are: $\{d, e\}$, $\{c, e\}$, $\{b, e\}$, $\{b, d\}$, $\{b, c\}$, $\{a, d\}$, $\{a, e\}$ and $\{a, c\}$.

The matroid $\mathcal{M}(G_1)$ has rank $r = 3$ while the dual matroid $\mathcal{M}^*(G_1)$ has rank $r = 2$.

If with create a graphic matroid $\mathcal{M}(G_1)$ and compare it to the matroid $\mathcal{M}_1(M_1)$ we find that the matroids are still isomorphic. In both cases the rank of the matroid decreased while the rank of the dual matroid was not impacted by the contraction. Now we can demonstrate deletion.

Take the matroid $\mathcal{M} = (E, \mathcal{B})$ and consider a subset $X \subseteq E$. We can define a new family of bases \mathcal{B}' so that an $Y \subseteq E - X$ exists so that $Y \in \mathcal{B}'$ if and only if $Y \in \mathcal{B}$. This new family \mathcal{B}' is formed by the bases of a new matroid \mathcal{M}' on $E - X$. The matroid \mathcal{M}' is denoted by $\mathcal{M} \setminus X$ and is called the deletion of X from \mathcal{M} or the restriction of \mathcal{M}

Linear matroid contraction: For the linear matroids we will contract the column m_6 using matroid $\mathcal{M}(M)$, and then we will find a matrix representation M_1 of the new matroid $\mathcal{M}_1(M_1)$. A step by step process of manually contracting columns is presented in Appendix C.

$$[M_1]_{3,5} = \begin{bmatrix} m_1 & m_2 & m_3 & m_4 & m_5 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix} \quad (32)$$

The bases of the matroid $\mathcal{M}_1(M_1)$ are: $\{m_1, m_2, m_3\}$, $\{m_1, m_2, m_4\}$, $\{m_1, m_3, m_4\}$, $\{m_1, m_3, m_5\}$, $\{m_1, m_4, m_5\}$, $\{m_2, m_3, m_4\}$, $\{m_2, m_3, m_5\}$ and $\{m_2, m_4, m_5\}$

The cobases of the matroid $\mathcal{M}_1(M_1)$ are: $\{m_4, m_5\}$, $\{m_3, m_5\}$, $\{m_2, m_5\}$, $\{m_2, m_4\}$, $\{m_2, m_3\}$, $\{m_1, m_5\}$, $\{m_1, m_4\}$ and $\{m_1, m_3\}$.

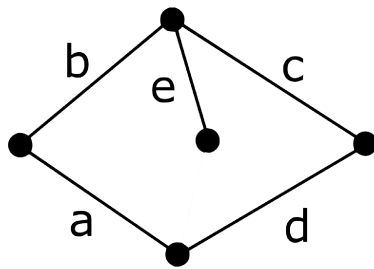
The matroid $\mathcal{M}_1(M_1)$ has rank $r = 3$ while the dual matroid $\mathcal{M}^*_1(M_1)$ has rank $r = 2$.

to $E - X$.

Graph deletion: For the graphic matroids, once again we demonstrate deletion using graph theory. Let us delete edge f : this step consists in simply removing the desired edge from the graph G . We will attain graph G_2 .

Linear matroid deletion: The demonstration of deletion once again is done using the matroid $\mathcal{M}_1(M_1)$. Deleting column m_6 using matroid $\mathcal{M}(M)$, and then finding the matrix representation M_2 of the new matroid $\mathcal{M}_2(M_2)$.

Figure 38 – Graph G_2 - Graph G after a deletion of edge f .



Source – From the author.

$$[M_2]_{4,5} = \begin{matrix} & m_1 & m_2 & m_3 & m_4 & m_5 \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix} \tag{33}$$

The bases of the matroid $\mathcal{M}(G_2)$ are: $\{a, b, c, e\}$, $\{a, b, d, e\}$, $\{a, c, d, e\}$ and $\{b, c, d, e\}$.

The bases of the matroid $\mathcal{M}_2(M_2)$ are: $\{m_1, m_2, m_3, m_5\}$, $\{m_1, m_2, m_4, m_5\}$, $\{m_1, m_3, m_4, m_5\}$ and $\{m_2, m_3, m_4, m_5\}$.

The cobases of the matroid $\mathcal{M}(G_2)$ are: $\{d\}$, $\{c\}$, $\{b\}$ and $\{a\}$.

The cobases of the matroid $\mathcal{M}_2(M_2)$ are: $\{m_4\}$, $\{m_3\}$, $\{m_2\}$ and $\{m_1\}$.

The matroid $\mathcal{M}(G_2)$ has rank $r = 4$ while the dual matroid $\mathcal{M}^*(G_2)$ has rank $r = 1$.

The matroid $\mathcal{M}_2(M_2)$ has rank $r = 4$ while the dual matroid $\mathcal{M}^*_2(M_2)$ has rank $r = 1$.

Creating a graphic matroid $\mathcal{M}(G_2)$ and comparing it to the matroid $\mathcal{M}_2(M_2)$ we find that the matroids are still isomorphic. In both cases the rank of the matroid remained unchanged while the rank of the dual matroid decreased due to the deletion.

Oxley (2006) presents the relation between contraction and deletion, given in Equation 32.

$$\mathcal{M}/T = (\mathcal{M}^* \setminus T)^* \tag{32}$$

Recski (2013) also presents another relation between contraction and deletion,

shown in Equation 33.

$$(\mathcal{M} \setminus T)^* = \mathcal{M}^* / T \quad (33)$$

These relation means that the contraction of a subset T of the dual matroid is equivalent to the dual of the deletion of the same subset T of the matroid. Due to the relations between Equations 32 and 33, Recski affirms that contraction and deletion are dual to each other; therefore, we can contract an element of the dual matroid in order to delete the same element of the original matroid.

Although presenting contraction and deletion for both graphs and matrices, in this work we use only the matrix formulation of these operations.

5.4 ANOTHER APPROACH INTO DELETION

In Section 5.3, deletion of matroids was briefly demonstrated. Using either graphs or matrices, it was shown that deleting an element from a matroid resulted in decreasing the groundset and the rank of the cobases; on the other hand, the rank of the bases remained unaltered.

Let us take this explanation a little further. Oxley (2006) states that given a matroid \mathcal{M} on a groundset E and a subset T of E , a matroid can be created such that $\mathcal{M} \setminus T$ on $E - T$ elements. In other words, we have a matroid and we are deleting T elements from that matroid. Oxley (2006) also affirms that the independent sets of the matroid $\mathcal{M} \setminus T$ are contained in $E - T$. This affirmation can be explained differently. Let us turn again to the matrix $[M]$:

$$[M]_{4,6} = \begin{array}{c} \begin{array}{cccccc} m_1 & m_2 & m_3 & m_4 & m_5 & m_6 \end{array} \\ \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{array} \right] \end{array} \quad (34)$$

The bases from the matroid $\mathcal{M}[A]$ are shown in Table 9:

Table 9 – List of bases and cobases from matroid $\mathcal{M}[M]$.

Bases	Cobases
$\{m_1, m_2, m_3, m_5\}$	$\{m_4, m_6\}$
$\{m_1, m_2, m_3, m_6\}$	$\{m_4, m_5\}$
$\{m_1, m_2, m_4, m_5\}$	$\{m_3, m_6\}$
$\{m_1, m_2, m_4, m_6\}$	$\{m_3, m_5\}$
$\{m_1, m_3, m_4, m_5\}$	$\{m_2, m_6\}$
$\{m_1, m_3, m_4, m_6\}$	$\{m_2, m_5\}$
$\{m_1, m_3, m_5, m_6\}$	$\{m_2, m_4\}$
$\{m_1, m_4, m_5, m_6\}$	$\{m_2, m_3\}$
$\{m_2, m_3, m_4, m_5\}$	$\{m_1, m_6\}$
$\{m_2, m_3, m_4, m_6\}$	$\{m_1, m_5\}$
$\{m_2, m_3, m_5, m_6\}$	$\{m_1, m_4\}$
$\{m_2, m_4, m_5, m_6\}$	$\{m_1, m_3\}$

Source – From the author.

In Table 9, on the left column we have the bases while their respective cobases are on the right column. In Table 10 we show only the cobases that include element m_5 and each respective basis.

Table 10 – List of cobases that include element m_5 and their respective bases.

Bases	Cobases
$\{m_1, m_2, m_3, m_6\}$	$\{m_4, m_5\}$
$\{m_1, m_2, m_4, m_6\}$	$\{m_3, m_5\}$
$\{m_1, m_3, m_4, m_6\}$	$\{m_2, m_5\}$
$\{m_2, m_3, m_4, m_6\}$	$\{m_1, m_5\}$

Source – From the author.

From the cobases of Table 10 we now remove the element m_5 , creating Table 11.

Table 11 – List of bases and cobases of matroid $\mathcal{M}[M \setminus m_5]$.

Bases	Cobases
$\{m_1, m_2, m_3, m_6\}$	$\{m_4\}$
$\{m_1, m_2, m_4, m_6\}$	$\{m_3\}$
$\{m_1, m_3, m_4, m_6\}$	$\{m_2\}$
$\{m_2, m_3, m_4, m_6\}$	$\{m_1\}$

Source – From the author.

After the operation done using element m_5 , the new matroid has only four bases. This is equivalent to the deletion of element m_5 of the matroid $\mathcal{M}(M)$.

5.5 ANOTHER APPROACH INTO CONTRACTION

In Section 5.3, contraction of matroids was briefly demonstrated. Using either graphs or matrices, it was shown that contracting an element from a matroid resulted in decreasing the ground set and the rank of the bases; on the other hand, the rank of the cobases remained unaltered.

Let us demonstrate contraction similarly to the demonstration from deletion. Once again we will use matrix $[M]$:

$$[M]_{4,6} = \begin{matrix} & m_1 & m_2 & m_3 & m_4 & m_5 & m_6 \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} & & & & & & \end{matrix} \quad (35)$$

The bases from the matroid $\mathcal{M}[M]$ are shown in Table 12:

Table 12 – List of bases and cobases from matroid $\mathcal{M}[M]$.

Bases	Cobases
$\{m_1, m_2, m_3, m_5\}$	$\{m_4, m_6\}$
$\{m_1, m_2, m_3, m_6\}$	$\{m_4, m_5\}$
$\{m_1, m_2, m_4, m_5\}$	$\{m_3, m_6\}$
$\{m_1, m_2, m_4, m_6\}$	$\{m_3, m_5\}$
$\{m_1, m_3, m_4, m_5\}$	$\{m_2, m_6\}$
$\{m_1, m_3, m_4, m_6\}$	$\{m_2, m_5\}$
$\{m_1, m_3, m_5, m_6\}$	$\{m_2, m_4\}$
$\{m_1, m_4, m_5, m_6\}$	$\{m_2, m_3\}$
$\{m_2, m_3, m_4, m_5\}$	$\{m_1, m_6\}$
$\{m_2, m_3, m_4, m_6\}$	$\{m_1, m_5\}$
$\{m_2, m_3, m_5, m_6\}$	$\{m_1, m_4\}$
$\{m_2, m_4, m_5, m_6\}$	$\{m_1, m_3\}$

Source – From the author.

Table 13 presents the bases from matroid $\mathcal{M}[M]$ that contain the element m_6 as well as each respective cobasis.

Table 13 – List of bases and cobases of matroid that include element m_6 and their respective cobases.

Bases	Cobases
$\{m_1, m_2, m_3, m_6\}$	$\{m_4, m_5\}$
$\{m_1, m_2, m_4, m_6\}$	$\{m_3, m_5\}$
$\{m_1, m_3, m_4, m_6\}$	$\{m_2, m_5\}$
$\{m_1, m_3, m_5, m_6\}$	$\{m_2, m_4\}$
$\{m_1, m_4, m_5, m_6\}$	$\{m_2, m_3\}$
$\{m_2, m_3, m_4, m_6\}$	$\{m_1, m_5\}$
$\{m_2, m_3, m_5, m_6\}$	$\{m_1, m_4\}$
$\{m_2, m_4, m_5, m_6\}$	$\{m_1, m_3\}$

Source – From the author.

From the bases of Table 13, we remove the element m_6 , creating Table 14.

Table 14 – List of bases and cobases of matroid $\mathcal{M}[M/m_6]$.

Bases	Cobases
$\{m_1, m_2, m_3\}$	$\{m_4, m_5\}$
$\{m_1, m_2, m_4\}$	$\{m_3, m_5\}$
$\{m_1, m_3, m_4\}$	$\{m_2, m_5\}$
$\{m_1, m_3, m_5\}$	$\{m_2, m_4\}$
$\{m_1, m_4, m_5\}$	$\{m_2, m_3\}$
$\{m_2, m_3, m_4\}$	$\{m_1, m_5\}$
$\{m_2, m_3, m_5\}$	$\{m_1, m_4\}$
$\{m_2, m_4, m_5\}$	$\{m_1, m_3\}$

Source – From the author.

The bases enumerated in Table 14 are the same as the bases enumerated using *SageMath* in Section 5.3, hence by selecting the bases as demonstrated we can contract elements from the matroid $\mathcal{M}(M)$.

In Section 3.3, the deletion of columns of an $[A_N]$ matrix with redundant constraints was used to demonstrate how to create mechanisms starting from that matrix. As we demonstrated that the deletion and contraction of matroid elements are dual to each other, we can conclude that the same principle applies to the enumeration of mechanisms starting from matrix $[A_N]$. The dual of removing a constraint from the mechanism can be described as guaranteeing that said constraint is present in the desired mechanism. Therefore, when we wish to give a freedom to a joint, we delete the constraint, while when we wish a certain constraint present in the mechanism, we contract the desired constraint. In the next chapter this idea will be formalized and applied to some examples.

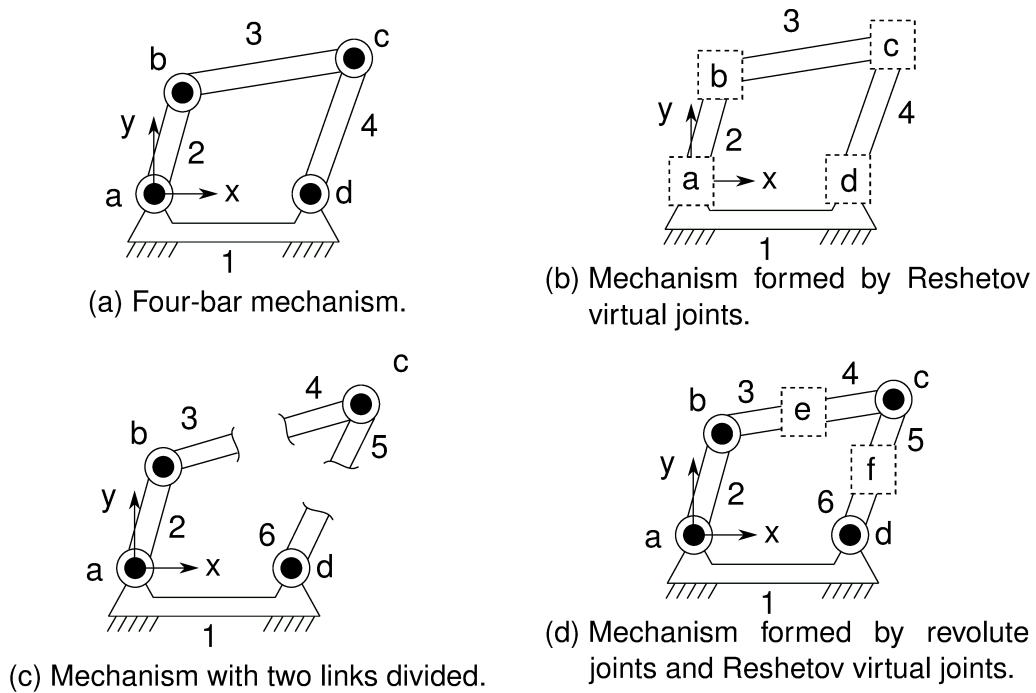
6 CONTRACTION AND DELETION APPLIED TO DAVIES' METHOD

In this chapter, contraction and deletion of matroids are applied to matrices generated according to Davies' method. After creating the network action matrix $[A_N]$, we can create the linear matroid using this matrix. The bases from the matroid are self-aligning mechanisms while the cobases are a set of redundant constraints (CARBONI et al., 2017). However, as mentioned in the previous chapter, matroids have an inherent difficulty with the size of the groundset. Systems modelled using Davies' method can pose a challenge when using matroids, therefore in this work it is proposed to use the matroid operations contraction and deletion as means to select the results.

In Chapter 3.3 Davies' method was introduced, it was discussed briefly how the network action matrix $[A_N]$ is constructed and how different mechanisms can be derived from an overconstrained mechanism. According to Carboni et al. (2017), when a matroid is created from a overconstrained mechanism from the network action matrix $[A_N]$, the bases of this matroid are the sets of constraints that form a self-aligning mechanism, while the cobases are the sets of constraints eliminated from the overconstrained mechanism. It is by the removal of the sets of redundant constraints that an overconstrained mechanism becomes a self-aligning mechanism. Carboni et al. (2017) presented a method to use matroids for enumerating every possible combination of self-aligning mechanism starting from an overconstrained mechanism.

Chapter 4 introduced the concept of Reshetov virtual joints, which are joints without freedoms. These joints were then used to design new gripper mechanisms based on Baranov chains. Reshetov virtual joints can be used in more ways than the presented method in Chapter 4, including the selection of the columns. Figure 39 presents three approaches of modelling mechanisms with redundant constraints used in this work, where Reshetov virtual joints are represented by a square with dashed lines.

Figure 39 – Different approaches used in this work to model overconstrained mechanisms.



Source – From the author.

Figure 39a shows a regular four-bar mechanism, a planar mechanism with redundant constraints when analyzed in the spatial workspace. Figure 39b shows a mechanism formed exclusively by Reshetov virtual joints. Such a mechanism in theory is a rigid structure, *i.e.* a mechanism without degrees of freedom; however, by using matroids we can derive new self-aligning mechanisms from this structure. Figure 39c shows a four-bar mechanism but the coupler and output links have been divided each into two links. Figure 39d shows a mechanism formed by revolute joints and Reshetov virtual joints. The Reshetov virtual joints are used to connect the links divided in Figure 39c. Systems formed by the combination of regular joints and Reshetov virtual joints can be used to create new joints instead of increasing the freedoms of existing joints. Regular procedures for enumerating self-aligning mechanisms distributes freedoms to existing joints, which may create joints with higher pairs. Higher pairs, such as gears and cams, frequently require critical demands of surface finish and surface treatment that is not required for lower pairs (HUNT, 1978).

Using matroids we can enumerate every possible combination of constraints that can be eliminated in order to create a self-aligning mechanism, so constraints will be removed from joints and/or Reshetov virtual joints creating new mechanisms. Although the enumerating lists only the main axes of the joints, using the concept proposed by Reshetov, the mechanisms enumerated are the starting point of a dimensional synthesis, *i.e.* the mechanisms are not required to remain aligned with the axes enumerated

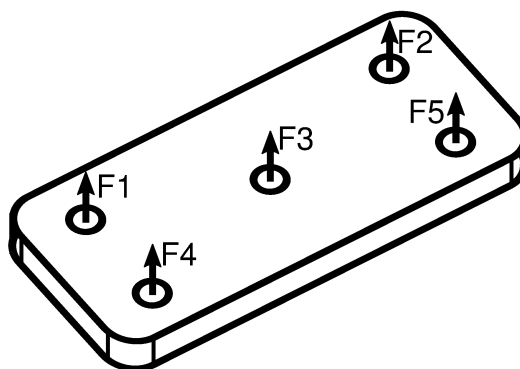
and can be repositioned as long as the main axis remains unaltered.

The challenge with enumerating self-aligning mechanism was introduced in Section 5.2, with the number of bases generated increasing exponentially with the increase of complexity of the mechanism. In this work it is proposed a method for selecting bases enumerated for the matroid created on the network action matrix $[A_N]$ using contraction and deletion. The following sections will discuss how each operation is used as means to filter the bases.

6.1 COMPARISON TO THE TABLE ANALOGY

In Section 1.2, an example of a redesign of a five legged table was used to introduce the notion of matroids, contraction and deletion. This section returns to that example using some of the mathematical tools presented in this work. Instead of using legs, consider the forces that the legs apply on the tabletop, as shown in Figure 40.

Figure 40 – Forces on the tabletop.



Source – From the author.

Table 15 presents the location of the forces on the tabletop. All of the forces are parallel to the z-axis.

Table 15 – Forces location on the table.

Force	x	y	z
F1	1	3	0
F2	5	3	0
F3	3	2	0
F4	1	1	0
F5	5	1	0

Source – From the author.

A matrix with the action screws acting on the tabletop is shown in Equation 36.

$$[F]_{6,5} = \begin{matrix} & F_1 & F_2 & F_3 & F_4 & F_5 \\ \begin{bmatrix} 3 & 3 & 2 & 1 & 1 \\ -1 & -5 & -3 & -1 & -5 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} & & & & & \end{matrix} \quad (36)$$

The rank of the matrix from Equation 36 is three, and since it has five columns, there are 2 linearly dependent columns. Employing *SageMath*, a matroid is created using the matrix with the action screws applied to the tabletop and the bases of that matroid are enumerated using the same software: $\{F_1, F_2, F_3\}$, $\{F_1, F_2, F_4\}$, $\{F_1, F_2, F_5\}$, $\{F_1, F_3, F_4\}$, $\{F_1, F_4, F_5\}$, $\{F_2, F_3, F_5\}$, $\{F_2, F_4, F_5\}$ and $\{F_3, F_4, F_5\}$. These bases are identical to the combinations enumerated in Section 1.2.

Once again using *SageMath*, the element F_3 will be contracted. The bases are: $\{F_1, F_2\}$, $\{F_1, F_4\}$, $\{F_2, F_5\}$ and $\{F_4, F_5\}$. These bases are also identical to the combinations remaining after the selection of the leg L3 to remain in the table.

Employing *SageMath* another time, the element F_3 will be deleted from the matrix $[F]$. The bases are: $\{F_1, F_2, F_4\}$, $\{F_1, F_2, F_5\}$, $\{F_1, F_4, F_5\}$ and $\{F_2, F_4, F_5\}$. Again, the bases enumerated are identical to the bases from the table example after removing the leg L3 from new table combinations.

In summary, the constraints acting on a body can be selected or eliminated in an analogous form as the selection or removal of legs from a table. Section 6.2 discusses the contraction of constraints while Section 6.3 addresses the deletion of constraints.

6.2 APPLYING CONTRACTION TO CONSTRAINTS

In Section 5.5 contraction was further explained using the effect that this operation has on the bases of a matroid. It is possible to deduce that when contraction of an element e is applied to a matroid, we are selecting a group of bases in which that element e is present. The bases where element e is not present, are removed from the group. Instead of enumerating every basis of the matroid and then selecting the bases with the desired element, using contraction the selection of bases happens without the computational cost of the enumeration. After the contraction, a new matroid is created from the original, formed by the desired bases and without the contracted element.

A selection process is derived from this logic: by selecting a group of bases with a given element, it is possible to reduce the number of existing bases to only those

bases with the desired element. Therefore, if we wish for a constraint to be present in the final self-aligning mechanism, we can contract this constraint selecting only the bases that had this constraint as viable bases, thus serving as a filter.

For example, we wish that a joint j is an actuated prismatic joint in the x axis. We then contract the constraint $\$U(j_U)$, and select only the bases which contain this constraint, removing unwanted bases. Through the contraction of the desired element, the ground set decreases making the matroid bases enumeration a simpler task while effectively imposing design constraints to the mechanism design process.

Returning to the five legged table example, the combinations with the leg selected to remain in the new table were kept while the other combinations were eliminated.

The tabletop will be stable with three forces, so two forces are redundant. The same selection process that was used for the selection of leg combinations can be done with their respective forces, thus the designer can select which forces remain in the device and which redundant constraints are eliminated. The steps a designer is required to take are: listing the linearly independent combinations of forces applied to the tabletop; selecting a force that must remain; eliminate the forces combinations that do not present the desired force; eliminate the force from the combinations.

Comparing the matroids and selection process with forces selection, the matroid is the system with five forces, the bases are the combinations, the removal of undesired combinations is the contraction. The last step was the removal of the force from the combinations as it was known that the selected force was present in every combination. For matroids, the contracted element is not present in the bases of the new matroid, but it is known that the bases had the element. When finishing the selection process, the designer must return the contracted element to the remaining bases for a full list of constraints present in the new self-aligning mechanism.

6.3 APPLYING DELETION TO CONSTRAINTS

In Section 5.4 deletion was exemplified by means of the effect this operation has on the cobases of a matroid. Using a similar logic for deletion as used previously for contraction, by deleting an element e from the cobases, we are generating only the group of cobases which contains that element e .

Using deletion, the selection process happens by selecting constraints that are not desired in the final mechanism. For example, if a joint j is desired to present a revolute motion around the z axis, we must delete the constraint $\$T(j_T)$, thus providing the desired freedom for the joint j .

By deleting a constraint from the matroid, we are removing that constraint from the mechanism, therefore we can say that we are adding a freedom to that mechanism. The deletion of the desired element, decreases the number of elements in the ground

set making the matroid bases enumeration a simpler task while using design constraints to the help in the mechanism design process.

Returning to the example of the five legged table, the contracted constraint is no longer present in the bases after contraction, however it is known by the designer that it is part of the system.

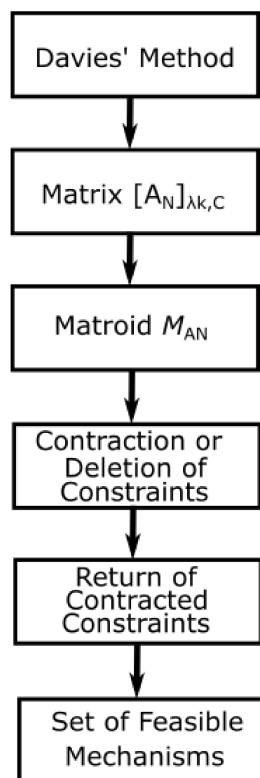
Similar to the explanation for the contracted elements of a matroid, after the deletion of constraints, the new cobases of the new matroid do not present the deleted element. After the whole filtering process, the deleted elements should return for the cobases for a full list of freedoms of the self-aligning mechanism created. The comparison between the contraction and deletion highlights the duality between these operations. Contracted elements must be returned to the bases while deleted elements must be returned to the cobases.

6.4 SUMMARY OF THE SELECTION PROCESS

After presenting how to select or remove constraints, we can summarize the selection process as contracting or deleting constraints from the matroid. When, by contraction, we remove unwanted bases, we also remove their respective cobases. When, by deletion, we remove unfeasible cobases, we are removing their respective bases as well.

After constructing a network action matrix $[A_N]$, creating a matroid using that matrix, we have deleted and contracted several constraints and we have found a set of feasible bases and cobases. The next step is to add the contracted constraints to the remaining bases. The bases formed by this addition are the bases of new self-aligning mechanisms, in the same format as proposed by Carboni (2015). The steps involved in the filtering process are shown in Figure 41.

Figure 41 – Flowchart of the Filtering Process.



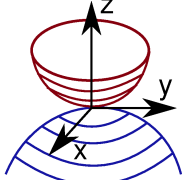
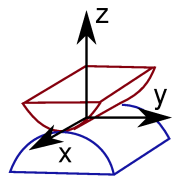
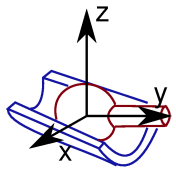
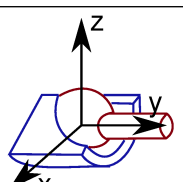
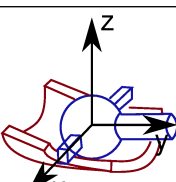
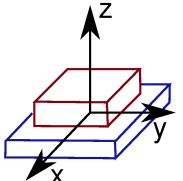
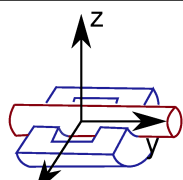
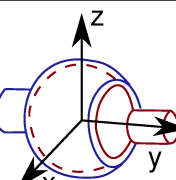
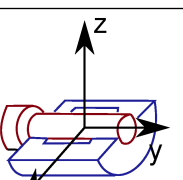
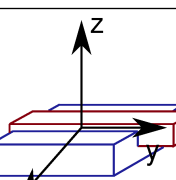
Source – From the author.

The process from Figure 41 will be used in the future chapters of this work.

6.5 PROPOSITION OF A NOTATION TO REPRESENT MATROID OPERATIONS

In order to facilitate the demonstration of each step taken using contraction and deletion of linear matroids, a notation is herein proposed. The proposed notation is based on a previous notation proposed by Reshetov (1979), in which the joints are divided into types according to the number of constraints present. Some examples of joints, the number of constraints and the type of the joint using Roman numerals are presented in Figure 42.

Figure 42 – Reshetov notation for joint types.

Type	Pair	Constraints		Pair	Constraints	
<i>I</i>	 Point pair		$\$W$			
<i>II</i>	 Line pair	$\$R$	$\$W$	 Annular pair		$\$U$ $\$W$
<i>III</i>	 Spherical pair		$\$U$ $\$V$ $\$W$	 Annular with pin pair	$\$S$	$\$U$ $\$W$
<i>III</i>	 Planar pair	$\$R$ $\$S$	$\$W$			
<i>IV</i>	 Cylindrical pair	$\$R$ $\$T$	$\$U$ $\$W$	 Spherical with pin pair	$\$S$	$\$U$ $\$V$ $\$W$
<i>V</i>	 Revolute pair	$\$R$ $\$T$	$\$U$ $\$V$ $\$W$	 Prismatic pair	$\$R$ $\$S$ $\$T$	$\$U$ $\$W$

Source – Adapted from Reshetov (1979).

Take, for instance an spherical pair. Such a joint has the three force constraints $\$U$, $\$V$ and $\$W$, thus this kinematic pair is of the type *III*. The pairs shown in Figure 42

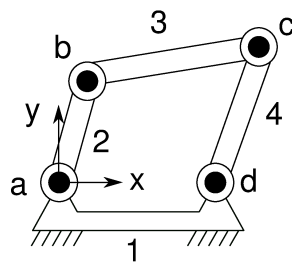
have been given arbitrary axes directions, which means that changing the direction of these axes the joints will present different constraints. Take for example the point pair shown in Figure 42, this pair presents the constraint $\$W$ as the contact is aligned with the z -axis. If a point pair was aligned with the x -axis, the constraint present would be the $\$U$. A revolute joint is a type V joint, while a cylindrical joint is a type IV joint. In this chapter the concept of Reshetov virtual joint was reviewed. This kind of structure is considered a type VI joint, as it presents every constraint.

The notation introduced in this section uses the type of joints proposed by Reshetov to represent the number of constraints present in each joint. The notation is presented in Table 16.

In summary, the contracted constraints are shown in the bottom right, such as V_R , while deleted constraints cause the type of the joint to decrease, like V to IV , and the constraint is placed in the top left, such as TIV .

The four-bar mechanism, shown in Figure 43, is used as example of the notation.

Figure 43 – Four-bars used in the example.



Source – From the author.




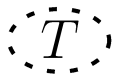
The four-bar mechanism is formed by four revolute joints, all of which are type V according to the Reshetov notation. Using Davies' method, the unit action matrix is created:

$$\begin{aligned}
 [\widehat{\mathbf{A}}_{\mathbf{D}}]_{18,20} = & \begin{bmatrix}
 \$_{aR}^a & \$_{aS}^a & \$_{aU}^a & \$_{aV}^a & \$_{aW}^a & \dots & (a) \\
 \dots & \$_{bR}^a & \$_{bS}^a & \$_{bU}^a & \$_{bV}^a & \$_{bW}^a & \dots & (b) \\
 \dots & \$_{cR}^a & \$_{cS}^a & \$_{cU}^a & \$_{cV}^a & \$_{cW}^a & \dots & (c) \\
 \dots & \$_{dR}^a & \$_{dS}^a & \$_{dU}^a & \$_{dV}^a & \$_{dW}^a & \dots & (d)
 \end{bmatrix} \quad (37)
 \end{aligned}$$

A matroid is created using the network action matrix $[A_N]$. This matroid has 112 bases and using the notation presented a selection of a basis will be shown.

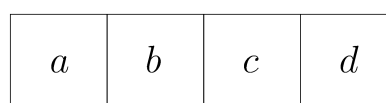
Before beginning the demonstration, the label of the joints a , b , c and d of the mechanism are organized in the top, as shown in Figure 44.

Table 16 – Explanation of the proposed notation.

Representation	Explanation
V	Representation for type V joints, such as prismatic and revolute joints
V 	The contracted constraint is shown on the bottom right of the type V joint
V 	When a moment constraint R is contracted, the letter R is shown in the indicated location
$V \rightarrow IV$	When a constraint is deleted from a joint, the number of constraints of that joint decreases, therefore a type V joint becomes a type IV joint
 IV	The deletion is shown in the top left of the joint type
 IV	Placement of the deleted constraint T
$V_{R, S, U, V, W}$	The problem with contracting many constraints is that the representation becomes large and cumbersome
V_r	Replacing groups of constraints from Table 17 with a symbol to simplify the notation

Source – From the author.

Figure 44 – Organization of the joints.



Source – From the author.

Underneath each joint label, the joint types are placed according to the seed

mechanism, shown in Figure 45.

Figure 45 – Joint types organized according to the seed mechanism.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
Seed Mechanism	<i>V</i>	<i>V</i>	<i>V</i>	<i>V</i>

Source – From the author.

The representation of the joints using the number of constraints enables the calculation of the constraints present in the mechanism. In this case, five times four is twenty, which is the number of constraints found in a spatial four-bar mechanism modelled by Davies' method.

The first step of the selection process is to define joint *a* as revolute, thus requiring the contraction of every constraint so that no deletion can happen. By using the notation proposed, the constraints *R*, *S*, *U*, *V* and *W* must be shown in the proposed place, such as $V_{R,S,U,V,W}$.

Using all five constraints as subscripts creates a big notation that may be simple to read but will become troublesome when using several joints, therefore a simplification is helpful. When considering that for a revolute joint all constraints are contracted, then only one symbol is enough to represent this operation. A similar conclusion can be found when considering most joint types, which leads to the creation of constraint sets. The constraints contracted from a revolute joint can be represented by the lower case letter *r*, thus creating a constraint set that is specific to revolute joints. As the joint is known, the freedom of the joint is also known, thus representing exactly which constraints are represented by *r* is superfluous, it is important to know that there are two moment constraint and three force constraints present in a revolute joint. Table 17 presents the constraint sets for several joints as well as for pure forces and pure moments conditions.

Table 17 – Constraint sets symbols.

Constraints set	Symbol	Number of constraints	Constraints	Observation
Forces	f	3	three forces	
Moments	m	3	three moments	
Revolute joint	r	5	two moments and three forces	
Prismatic joint	p	5	three moments and two forces	
Cylindrical joint	c	4	two moments and two forces	freedoms in the same axis
Pin-in-slot joint	n	4	two moments and two forces	freedoms in different axes

Source – From the author.

Now the letter r is used as representation of the contracted constraints of a revolute joint, such as V_r . The contraction in this case uses r as the representation of every constraint contracted from the revolute joint. Returning to the table, after the contraction of the constraints of the joint a , the first step is shown in Figure 46.

Figure 46 – First step - contraction of the constraint from joint a .

	a	b	c	d
Seed Mechanism	V	V	V	V
First Step	V_r	V	V	V

Source – From the author.

For the next step joint d is defined as a cylindrical joint. To achieve this joint type, the force constraint in the z -axis must be deleted, thus providing the required freedom. The remaining constraint must be contracted to prevent future deletions. So, in the representation, joint d becomes a type IV and receives the letter W in the top left to demonstrate the deletion and the letter c in the bottom right to demonstrate the contraction of the constraints of the cylindrical joint. The result is presented in Figure 47.

Figure 47 – Representation after deletion and contractions.

	a	b	c	d
Seed Mechanism	V	V	V	V
First step	V_r	V	V	V
Second step	V_r	V	V	${}^W IV_c$

Source – From the author.

Using the notation shown in Figure 47, the designer can keep track and represent each operation already done using matroids in a simple manner. After the contraction and deletion operations, the designer must add the contracted constraint back to achieve the desired mechanism. The constraints that must be added are easily found as they are shown in the notation.

Continuing the operation as means of example, joint b will be defined as spherical, thus requiring the deletion of both moment constraints present in the revolute joint b . The third step of the representation is shown in Figure 48.

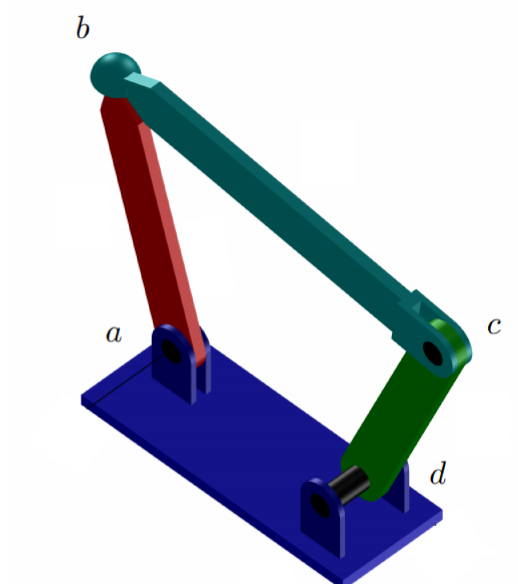
Figure 48 – Joint types organized according to the seed mechanism.

	a	b	c	d
Seed Mechanism	V	V	V	V
First step	V_r	V	V	V
Second step	V_r	V	V	${}^W IV_c$
Third step	V_r	${}^{R,S} III$	V	${}^W IV_c$

Source – From the author.

After the three steps shown, after contracting nine constraints and deleting other three constraints, the matroid will present only one basis. The self-aligning mechanism correlated to that basis is shown in Figure 49.

Figure 49 – Four-bars used in the example.



Source – Adapted from Carboni (2015).

Using matroids and the operations of contraction and deletion, one basis was selected from 112 bases. Further applications of matroids, contraction and deletion will be shown in future chapters.

7 ENUMERATION OF SELF-ALIGNING MECHANISMS USING DAVIES' METHOD AND MATROID THEORY: CASE STUDY OF THE KCL/TJU HAND

This chapter presents the enumeration of self-aligning mechanisms based on the reconfigurable palm of the anthropomorphic King's College London (KCL)/Tianjin University (TJU) hand. Preliminary results presented in this chapter were already published in Barreto et al. (2018).

The KCL/TJU anthropomorphic metamorphic robotic hand has been studied in several papers, including studies of the reconfigurable palm and applications with three, four and five fingers. Cui et al. (2009) analyzed the workspace of a metamorphic hand with three fingers; in Wei et al. (2011) the kinematic analysis of a new hand design was performed; in Wei et al. (2017) the prehension of the robotic hand was studied using opposition space model. Chunsong Zhang and Dai (2016) presented the inverse kinematics and kineto-statics of the metamorphic palm; an application was reviewed in Wei et al. (2014) where the use of the anthropomorphic hand was investigated for deboning operations in a human-robot co-working platform. Also for deboning operations, Wei et al. (2013) presented a four-fingered metamorphic hand with the same reconfigurable palm structure as the previously cited papers.

Despite the relevant contributions proposed for the KCL/TJU anthropomorphic metamorphic robotic hand, no work focused on the design of a self-aligning reconfigurable palm.

The anthropomorphic reconfigurable palm analyzed in this work was first developed by Wei et al. (2011) based on previous hands developed and patented by Dai (2008). This hand presents a reconfigurable palm, in order to increase the dexterity of the whole hand, a thumb and four fingers. The thumb has four DOF while each of the remaining fingers has three DOF. The different operation modes of the hand rely on the orientation and positioning of the links in the reconfigurable palm. A CAD model, based on Wei et al. (2011) and Wei et al. (2017), is presented in Figure 50.

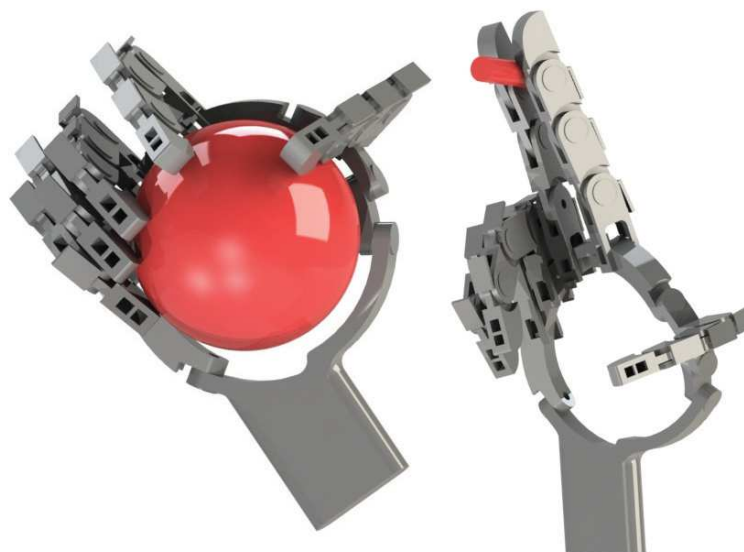
Figure 50 – CAD model of the anthropomorphic reconfigurable palm.



Source – From the author.

The anthropomorphic hand is able to perform several prehensile operations, such as operating pliers and scissors, holding a bowl or a bottle, grasping a ball or a cylinder, holding a key, pinching a coin, holding a card and even operating a comb (WEI et al., 2017). Two of these prehensile actions, grasping a ball and a chalk, are presented in Figure 51. The hand is also able to almost completely fold itself, which is useful for saving space for storage (WEI et al., 2011).

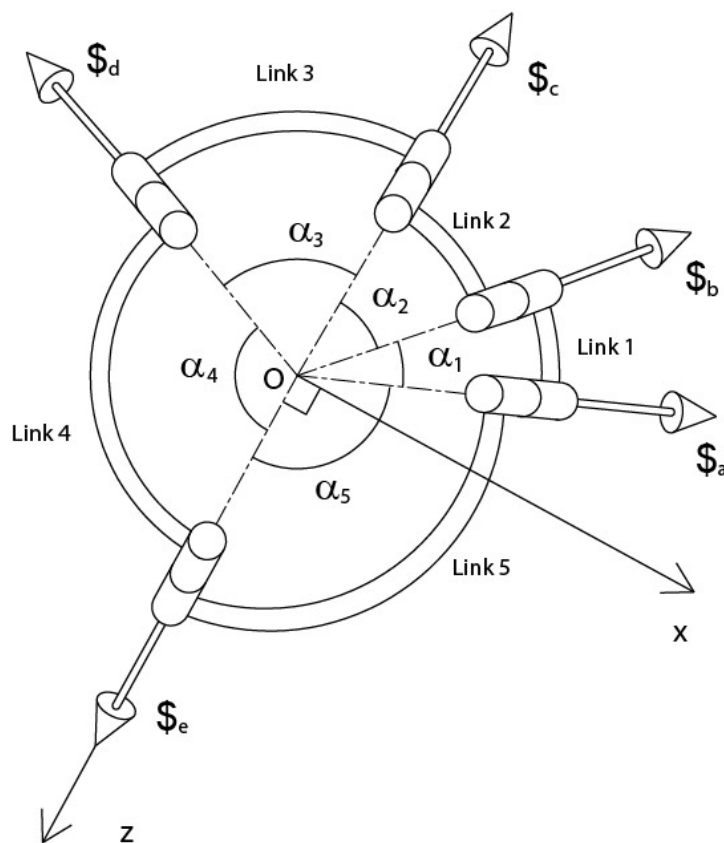
Figure 51 – Some prehensile actions of the anthropomorphic reconfigurable hand.



Source – From the author.

The screw system of the reconfigurable palm is presented in Figure 52.

Figure 52 – Screw system of the Reconfigurable Palm



Source – From the author.

This is a spherical five-bar linkage, as the revolution axis of every joint intersect each other in the same point. The angles corresponding to links 1 to 5, defined in Figure 52, are assigned to satisfy the condition (WEI et al., 2017):

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 = 2\pi \quad (38)$$

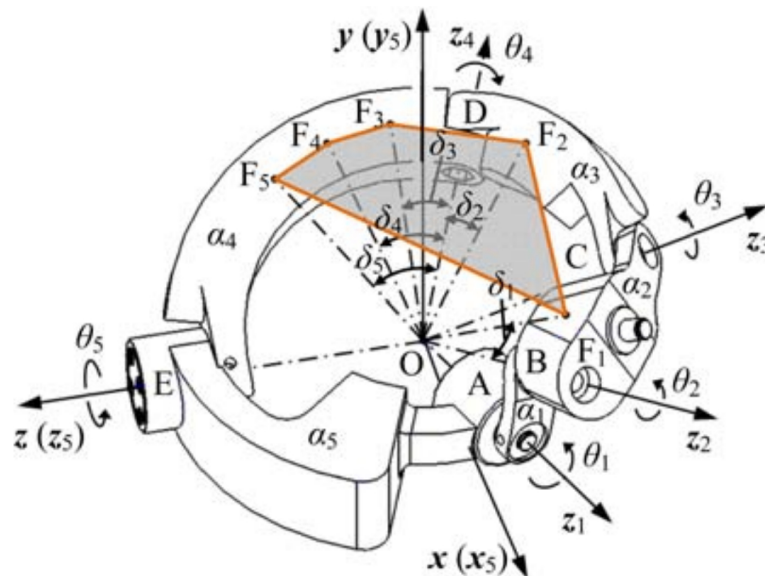
Thus, the angles of the links are assigned as: $\alpha_1 = 25^\circ$, $\alpha_2 = 40^\circ$, $\alpha_3 = 70^\circ$, $\alpha_4 = 112^\circ$ and $\alpha_5 = 113^\circ$.

The motion screws associated with the joints a, b, c, d and e can be written in the form:

$$\begin{aligned} \$^m_a &= [P_{ax} \ P_{ay} \ P_{az} \ 0 \ 0 \ 0] \\ \$^m_b &= [P_{bx} \ P_{by} \ P_{bz} \ 0 \ 0 \ 0] \\ \$^m_c &= [P_{cx} \ P_{cy} \ P_{cz} \ 0 \ 0 \ 0] \\ \$^m_d &= [P_{dx} \ P_{dy} \ P_{dz} \ 0 \ 0 \ 0] \\ \$^m_e &= [P_{ex} \ P_{ey} \ P_{ez} \ 0 \ 0 \ 0] \end{aligned} \quad (39)$$

where the parameters of the mechanism are presented in the original Figure 53, extracted from Wei et al. (2011) where the points P_a, P_b, P_c, P_d and P_e are defined.

Figure 53 – Parameters of the reconfigurable palm



Source – Wei et al. (2011)

For a single-loop mechanism, as the anthropomorphic reconfigurable palm herein analyzed, the number of redundant constraints of the mechanism can be found analyzing the unit motion matrix $[M_D] = [\$^m_a \ \$^m_b \ \$^m_c \ \$^m_d \ \$^m_e]$ (CARBONI, 2015; CARBONI et al.,

2017). Each column of this matrix represents the motion screw associated with one freedom of a specific joint.

$$[M_D] = [{}^m_a \ {}^m_b \ {}^m_c \ {}^m_d \ {}^m_e] \quad (40)$$

Rearranging the transpose of the matrix $[M_D]^T$ in the reduced row echelon form (rref), the following matrix $[M_D]^T$ is obtained:

$$[M_D]^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Recalling Equation 24 from Section 3.3, it is possible to calculate the number of redundant constraints in the palm mechanism:

$$q = \lambda\nu - \text{rank}\{[M_D]\}. \quad (41)$$

The rank of matrix $[M_D]^T$ is three, thus this mechanism has three redundant constraints. Regard that this result is correct when the anthropomorphic reconfigurable palm is not in a singular configuration. Thus, in order to obtain an equivalent self-aligning anthropomorphic reconfigurable palm, these three constraints must be eliminated.

7.1 SELF-ALIGNING APPROACHES

Either the kinematics or statics analysis may be used to find the redundant constraints of a given mechanism. However, statics analysis, based on Davies' methodology (DAVIES, 2006c) and matroid theory (CARBONI et al., 2017), is interesting as it enables the designer to quickly find all redundant constraint free mechanism possibilities, derived from the initial mechanism.

Reshetov (RESHETOV, 1979) proposed a self-aligning approach based on finding redundant constraints and adding freedom to the existing joints of the mechanism. Thus, this approach increases the freedom of a joint eliminating some redundant constraints. Some examples of this approach can be considered:

- A rotation freedom is added to a revolute joint, turning it into a universal joint;
- A translation freedom is added to a revolute joint, turning it into a pin-in-slot joint (a superior pair);
- Two distinct rotation freedoms are added to a revolute joint, turning it into a spherical joint.

The problem with this approach is that inferior kinematic pairs may be transformed into superior pairs. Even though this is sometimes desired or acceptable, these type of contacts between bodies increases the strain, decreasing the reliability and life of the mechanism.

Reshetov (1979) also presented another approach to self-aligning mechanisms. Instead of adding new freedoms to the joints and turning them into more complex joints, other joints can be added to the mechanism, in order to eliminate the redundant constraints without altering its kinematics.

Unlike the previous approach, which transforms inferior kinematic pairs into superior pairs, this approach allows the designer to add only inferior pairs to the mechanism. The self-aligning mechanism will thus present more links and joints. Despite increasing the complexity of the system, this approach helps the designer to create more reliable and lasting mechanisms (RESHETOV, 1979).

7.2 METHOD I: INCREASING THE FREEDOM OF SPECIFIC JOINTS

First, the redundant constraints of the mechanism are analyzed. The metamorphic palm is a spherical mechanism composed by five revolute joints. The network action matrix $[A_{N_{palm_I}}]$ is created, as described in Section 3.3, defined from the action screws associated with the joints of the mechanism. A simplified cutset matrix is shown in Equation 42 while the unit action matrix is shown in Equation 43.

$$[Q_{A_{leg}}] = \begin{matrix} & a & b & c & d & e \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} & & & & & \end{matrix} \quad (42)$$

$$[A_{D_{palm_I}}]_{6,25} = \begin{bmatrix} \$_{aS}^a & \$_{aT}^a & \$_{aU}^a & \$_{aV}^a & \$_{aW}^a & \dots & (a) \\ \dots & \$_{bS}^a & \$_{bT}^a & \$_{bU}^a & \$_{bV}^a & \$_{bW}^a & \dots & (b) \\ \dots & \$_{cR}^a & \$_{cS}^a & \$_{cU}^a & \$_{cV}^a & \$_{cW}^a & \dots & (c) \\ \dots & \$_{dR}^a & \$_{dT}^a & \$_{dU}^a & \$_{dV}^a & \$_{dW}^a & \dots & (d) \\ \dots & \$_{eR}^a & \$_{eS}^a & \$_{eU}^a & \$_{eV}^a & \$_{eW}^a & \dots & (e) \end{bmatrix} \quad (43)$$

In Equation 42, each column represents all of the constraints of a joint, so in order to create the complete cutset matrix each column must be repeated the number of constraints each joint has. In this case, every joint has five constraints so every column

will be repeated five times. This simplification is done to save space when representing the cutset matrix.

In Equation 43, joints a and b were defined as revolute around the x -axis; joints c and e were defined as revolute around the z -axis; joint d was defined as revolute around the y -axis. As the palm is a spherical mechanism, the axes definition is based on an initial position. When the mechanism moves, these axes may change as well; however, if the mechanism is in a singular position, the rank of the matrix $[A_{N_{palm_I}}]$ will change. A linear matroid \mathcal{M}_p over the real field \mathbb{R} is then created defined from matrix $[A_{N_{palm_I}}]$.

Each base of this matroid represents a mechanism without redundant constraint, *i.e.* a self-aligning mechanism derived from the seed mechanism. Matrix $[A_{N_{palm_I}}]$ has 25 columns and rank = 22, and 2066 bases are encountered for the matroid created using that matrix.

In other words, 2066 new mechanisms exist that are kinematically equivalent to the metamorphic palm, but with no redundant constraint. Because of the huge number of possible self-aligning mechanisms, it is interesting to filter the results. Contraction and deletion of elements of the matroid \mathcal{M}_p will then be applied as the filter, so the notation shown in the previous chapter will be used. The joint types is shown in Figure 54.

Figure 54 – Joint types of the seed mechanism.

	a	b	c	d	e
Seed Mechanism	V	V	V	V	V

Source – From the author.

By adding the number corresponding to the types of the joints, the number of constraints present in the mechanism can be calculated. In this case the joints are type five for each one of them and the sum is twenty-five.

Recall that each basis of the matroid \mathcal{M}_p consists of a group of constraints that will remain on the self-aligning mechanism. Thus when constraints are not present in a basis, that constraint was removed and became a freedom. Recall also that joints a and e are actuated joints, and thus it is better to avoid giving additional freedoms to these joints.

First, the mechanisms, *i.e.* the matroid bases, for which some freedoms have been added to the actuated joints, joints a and e , are eliminated. In this way, no change in the actuation strategy is needed for the mechanism. Remember that Section 6.2 discussed that contracting constraints from a basis meant that the contracted constraints were maintained in the self-aligning mechanisms. Therefore, the constraints of joints

a and e must be contracted. The matroid \mathcal{M}_{p_1} is created after the contractions of the constraints of joint a ($\mathcal{M}_{p_1} = \mathcal{M}_p \setminus \{a_S, a_T, a_V, a_W\}$) and matroid \mathcal{M}_{p_2} is created after the contractions applied to joint e ($\mathcal{M}_{p_2} = \mathcal{M}_{p_1} \setminus \{e_R, e_S, e_U, e_V, e_W\}$). Both operations are shown in Figure 55.

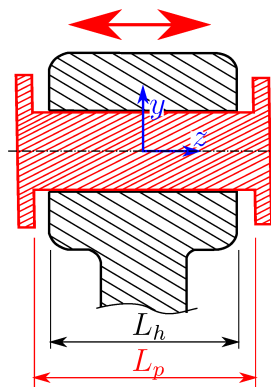
Figure 55 – Contractions of the constraint from joints a and e .

	a	b	c	d	e
Seed Mechanism	V	V	V	V	V
First Step	V_r	V	V	V	V_r

Source – From the author.

The first contraction operation decreases the number of feasible bases from 2066 to 1068, while the second operation provided a set of 429 feasible self-aligning mechanisms. A further selection in order to reduce the number of possible candidate mechanisms is desired. By analysing the palm mechanism, it is possible to deduce that joint b cannot be transformed into a more complex joint as both links connected to this joint are small. Therefore, small clearances can be used in this joint to help the self-aligning process. Figure 56 shows an axial clearance that can be applied to joint b . The application of selected small clearances to joints was already discussed in Artmann (2019) and Artmann et al. (2019b).

Figure 56 – Axial clearance applied to a revolute joint.



Source – Artmann (2019).

By providing the small axial clearance presented, joint b is transformed into a cylindrical joint. Returning to the matroid \mathcal{M}_{p_2} , defining joint b as cylindrical means deleting the constraint b_V that prevents the axial movement and contracting the remaining constraints preventing the removal of any other constraints. Now, a new matroid is

created as $\mathcal{M}_{p_3} = \mathcal{M}_{p_2}/\{b_U\}$ for the deletion operation. Later, when contracting constraints b_S, b_T, b_V, b_W , a matroid \mathcal{M}_{p_4} is created ($\mathcal{M}_{p_4} = \mathcal{M}_{p_3} \setminus \{b_S, b_T, b_V, b_W\}$). These operations are shown in Figure 57.

Figure 57 – Operation diagram for deletion of the constraint U and contractions of for Joint b .

	a	b	c	d	e
Seed Mechanism	V	V	V	V	V
First Step	V_r	V	V	V	V_r
Second Step	V_r	$^U IV_c$	V	V	V_r

Source – From the author.

Due to the deletion of the constraint U , the type of the joint decreases to IV . The remaining constraints are contracted, which is demonstrated by the letter c in the subscript. The matroid \mathcal{M}_{p_3} found after the deletion of the constraint b_U has 89 bases. Matroid \mathcal{M}_{p_4} calculated after the contraction of the remaining constraints of joint b has 45 bases. Therefore, after the operations shown in Figure 57 there are 45 feasible self-aligning mechanisms, which are shown in Appendix D.1. Table 18 presents a summary of the matroid operations done.

Table 18 – Summary of the steps in the filtering process taken in the Method I.

Joint	Operation	Matroid operation	Remaining bases
All	creation of linear matroid	$\mathcal{M}_p = ([A_{N_{palm_I}}], \mathcal{B}_p)$	2066
a	contract to guarantee revolute joint	$\mathcal{M}_{p_1} = \mathcal{M}_p / \{a_S, a_T, a_U, a_V, a_W\}$	1068
e	contract to guarantee revolute joint	$\mathcal{M}_{p_2} = \mathcal{M}_{p_1} / \{e_R, e_S, e_U, e_V, e_W\}$	429
b	delete to guarantee cylindrical joint	$\mathcal{M}_{p_3} = \mathcal{M}_{p_2} \setminus \{b_U\}$	89
b	contract to guarantee cylindrical joint	$\mathcal{M}_{p_4} = \mathcal{M}_{p_3} / \{b_S, b_T, b_V, b_W\}$	45

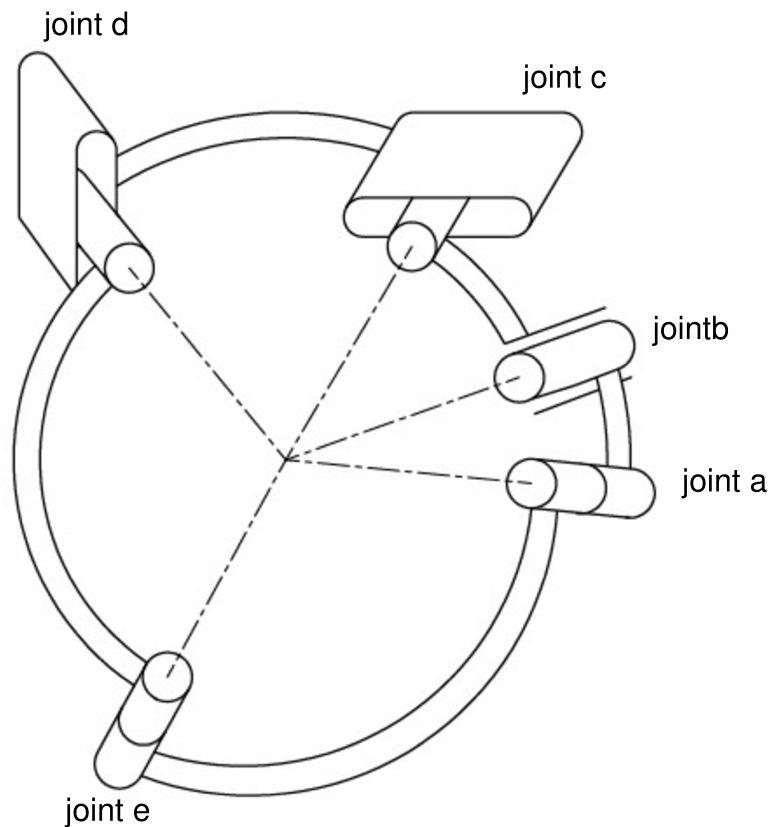
Source – From the author.

After the steps shown, only 45 mechanisms remain out of the initial 2066, representing 2,2% of the total bases. From the remaining 45 mechanisms, the designer can manually review each one of them or use other selection techniques, such as greedy algorithm or the binary cobases matrix. If the designer wishes a specific result and can ignore the remaining bases, the greedy algorithm is a good method; however, if the

designer wishes a group of self-aligning mechanisms with a set design of requirements, the binary cobases matrix is more suited.

One example of a mechanism enumerated is shown in Figure 58. Joints *c* and *d*, both revolute joints, have been replaced by pin-in-slot joints.

Figure 58 – Self-Aligning Mechanism by the Method I



Source – From the author.

Table 19 present the comparison between the network action matrices from the seed reconfigurable anthropomorphic palm and the self-aligning mechanism. Notice that the number of rows of the matrix $[A_N]$ is the same, as the screw system of the mechanism did not change. On the other hand, the number of columns is reduced, as three redundant constraints (b_T , c_R and d_S) have been eliminated. Finally, matrix $[A_N]$ corresponding to a self-aligning mechanism is a full rank matrix, as expected because no redundant constraints are present in the mechanism. The comparison between the network action matrices of the over-constrained mechanism and the self-aligning mechanism is shown in Table 19

Table 19 – Network Action Matrices Comparison Method I.

	Rows	Columns	Rank
Seed Mechanism	24	25	22
Mechanism 5	24	22	22

Source – From the author.

7.3 METHOD II: INCREASING THE NUMBER OF JOINTS

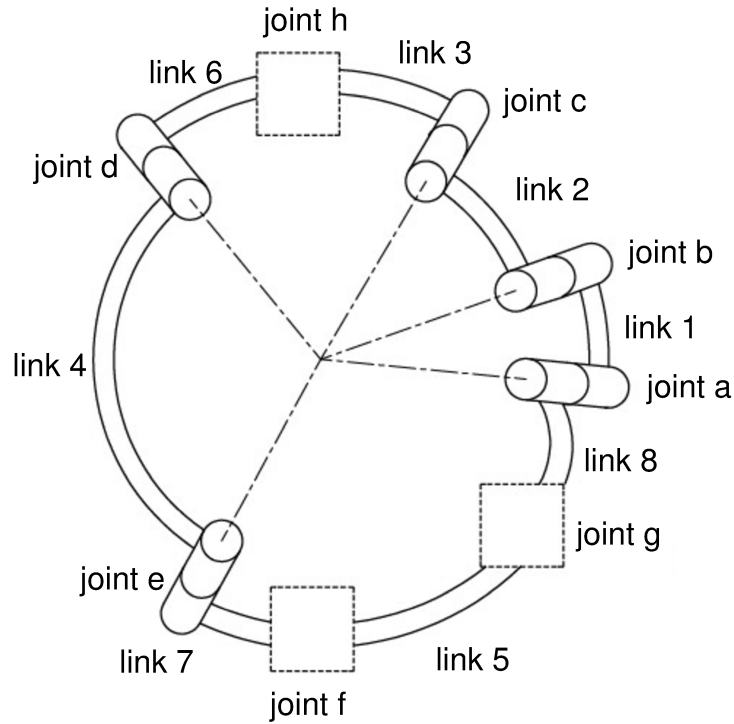
In the previous section, self-aligning mechanisms were obtained adding proper freedoms to the joints, in order to eliminate the redundant constraints.

However, joints with more than one freedom are more difficult to manufacture and are more prone to imperfections and problems with joint clearance. To deal with these issues, an alternative approach is herein proposed: adding new joints instead of replacing existing ones. In other words, new joints and links are added to the mechanism, in order to provide the missing freedoms and eliminate the redundant constraints.

First, Reshetov virtual joints, each one with six independent constraint and no freedom, are added to the mechanism. In this way, there is no change in the static analysis of the mechanism. The number of redundant constraints is the same, the only difference is the rank and number of rows and columns of the network action matrix $[A_N]$.

As the metamorphic palm has three redundant constraints, three Reshetov virtual joints can be added. In order to define the location where the new joints are placed, a further analysis of the mechanism is performed. Link 2 is very small while link 3 has the thumb actuation embedded. Link 5 has the coupling of three fingers and therefore it lacks space. With these conditions, links 1 and 4 are the best candidates to receive the additional joints. Furthermore, two joints are added to link 1 and one joint to link 4. The new joints location is presented in Figure 59.

Figure 59 – Position of the New Joints



Source – From the author.

The unit action matrix for the new system is presented in Equation 44.

$$\begin{aligned}
 [A_{D_{palm_{II}}}] = & \begin{bmatrix}
 \$_{aS}^a & \$_{aT}^a & \$_{aU}^a & \$_{aV}^a & \$_{aW}^a & \dots & (a) \\
 \dots & \$_{bS}^a & \$_{bT}^a & \$_{bU}^a & \$_{bV}^a & \$_{bW}^a & \dots & (b) \\
 \dots & \$_{cR}^a & \$_{cS}^a & \$_{cU}^a & \$_{cV}^a & \$_{cW}^a & \dots & (c) \\
 \dots & \$_{dR}^a & \$_{dT}^a & \$_{dU}^a & \$_{dV}^a & \$_{dW}^a & \dots & (d) \\
 \dots & \$_{eR}^a & \$_{eS}^a & \$_{eU}^a & \$_{eV}^a & \$_{eW}^a & \dots & (e) \\
 \dots & \$_{fR}^a & \$_{fS}^a & \$_{fT}^a & \$_{fU}^a & \$_{fV}^a & \$_{fW}^a & \dots & (f) \\
 \dots & \$_{gR}^a & \$_{gS}^a & \$_{gT}^a & \$_{gU}^a & \$_{gV}^a & \$_{gW}^a & \dots & (g) \\
 \dots & \$_{hR}^a & \$_{hS}^a & \$_{hT}^a & \$_{hU}^a & \$_{hV}^a & \$_{hW}^a & \dots & (h)
 \end{bmatrix}
 \end{aligned}
 \tag{44}$$

Then, after adding the three Reshetov virtual joints to the unit action matrix $[A_{D_{palm_{II}}}]$, the new network action matrix $[A_{N_{palm_{II}}}]$ is calculated. The comparison between the new and the original matrices is presented in Table 20 . The number of rows in the new matrix is increased from 24 to 42, due to the increase in the number of cutsets. The number of columns is increased to 43, because six columns are added for each new zero DOF joint. The rank of the matrix increased from 22 to 40, however

the number of redundant constraints remains the same. This relation is verified by subtracting the rank from the number of columns, which in both cases is three.

Table 20 – Network action matrices comparison with additional joints.

	Rows	Columns	Rank
Seed Mechanism	24	25	22
Mechanism with 3 Reshetov virtual joints	42	43	40

Source – From the author.

A matroid \mathcal{M}_{ps} created using the network action matrix of the mechanism with the Reshetov virtual joints presents 4242 bases, which will be selected using contraction and deletion. Following the procedure presented previously, Figure 60 is created to represent the joint types.

Figure 60 – Joint types of the seed mechanism.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
Seed Mechanism	<i>V</i>	<i>V</i>	<i>V</i>	<i>V</i>	<i>V</i>	<i>VI</i>	<i>VI</i>	<i>VI</i>

Source – From the author.

The Reshetov virtual joints *f*, *g* and *h* are type *VI* because each possess six constraints. As the objective of the Reshetov virtual joints is to receive freedoms, the constraints of the original revolute joints are all contracted, which is shown in Figure 61.

Figure 61 – Operation diagram for the contraction of the constraints from joints *a*, *b*, *c*, *d* and *e*.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
Seed Mechanism	<i>V</i>	<i>V</i>	<i>V</i>	<i>V</i>	<i>V</i>	<i>VI</i>	<i>VI</i>	<i>VI</i>
First Step	<i>V_r</i>	<i>V_r</i>	<i>V_r</i>	<i>V_r</i>	<i>V_r</i>	<i>VI</i>	<i>VI</i>	<i>VI</i>

Source – From the author.

The matroid \mathcal{M}_{ps_5} created after the contraction of the 25 constraints of the original joints has 460 feasible bases for self-aligning mechanisms. After designing and

assembling a certain number of spherical mechanisms, it was seen that a radial translation makes the whole assembly process easier, thus joint f is defined as a prismatic joint to enable this freedom. The matroid operation desired is a deletion of the constraint W , as the joint f has the z -axis pointed to the center of the spherical mechanism. The deletion of constraint W will transform joint f from a type VI into a type V , as the number of constraints will decrease. Also, the remaining constraints of the prismatic joint (R, S, T, U, V) must be contracted. The operations are presented in Figure 62.

Figure 62 – Operation Diagram for deletion of the constraint W and contractions of remaining constraints for Joint f .

	a	b	c	d	e	f	g	h
Seed Mechanism	V	V	V	V	V	VI	VI	VI
First Step	V_r	V_r	V_r	V_r	V_r	VI	VI	VI
Second Step	V_r	V_r	V_r	V_r	V_r	WV_p	VI	VI

Source – From the author.

Now joint f is no longer a Reshetov virtual joint, but instead it is a prismatic joint in the z -axis. After the deletion, the matroid $\mathcal{M}_{ps_6} = \mathcal{M}_{ps_5} \setminus \{f_W\}$ has 58 bases, while matroid $\mathcal{M}_{ps_7} = \mathcal{M}_{ps_6} / \{f_R, f_S, f_T, f_U, f_V\}$ has 33 bases. By these last two operations, a set of 33 feasible self-aligning mechanisms based on the seed reconfigurable palm have been listed using the method II. The list of mechanisms can be found in Appendix D.2. The set of 33 feasible self-aligning mechanisms represent 0,78% of all the self-aligning mechanisms enumerated using method II.

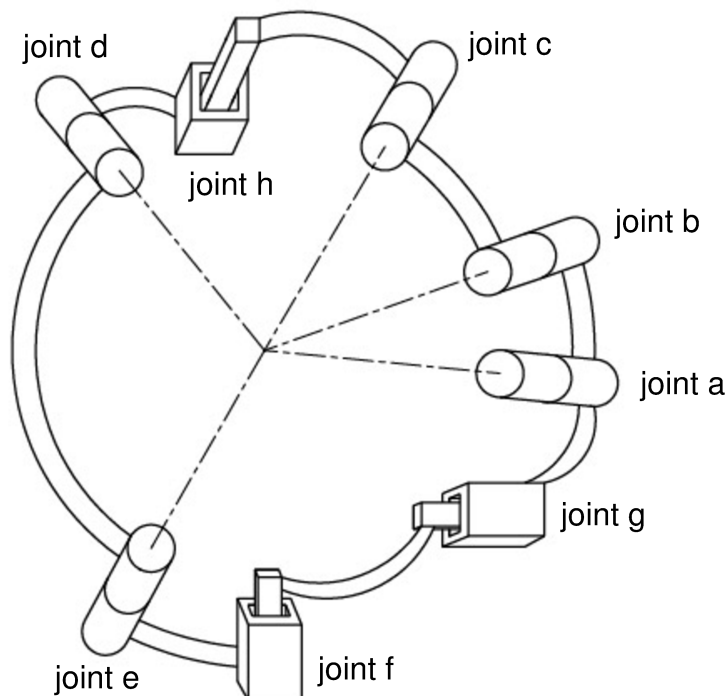
Table 21 – Summary of the steps in the filtering process taken in the Method II.

Joint	Operation	Matroid operation	Remaining bases
All	creation of linear matroid	$\mathcal{M}_{ps} = ([A_{N_{palmII}}], \mathcal{B}_p)$	4242
a	contract to guarantee revolute joint	$\mathcal{M}_{ps_1} = \mathcal{M}_{ps} / \{a_S, a_T, a_U, a_V, a_W\}$	3169
b	contract to guarantee revolute joint	$\mathcal{M}_{ps_2} = \mathcal{M}_{ps_1} / \{b_S, b_T, b_U, b_V, b_W\}$	2115
c	contract to guarantee revolute joint	$\mathcal{M}_{ps_3} = \mathcal{M}_{ps_2} / \{c_R, c_S, c_U, c_V, c_W\}$	1295
d	contract to guarantee revolute joint	$\mathcal{M}_{ps_4} = \mathcal{M}_{ps_3} / \{d_R, d_T, d_U, d_V, d_W\}$	682
e	contract to enable revolute joint	$\mathcal{M}_{ps_5} = \mathcal{M}_{ps_4} / \{e_R, e_S, e_U, e_V, e_W\}$	460
f	delete to guarantee prismatic joint	$\mathcal{M}_{ps_6} = \mathcal{M}_{ps_5} \setminus \{f_W\}$	58
f	contract to guarantee prismatic joint	$\mathcal{M}_{ps_7} = \mathcal{M}_{ps_6} / \{f_R, f_S, f_T, f_V, f_W\}$	33

Source – From the author.

As an example, a mechanism developed using method II is presented in Figure 63.

Figure 63 – Self-aligning mechanism developed using method II.



Source – From the author.

In the mechanism from Figure 63, joint *g* is now a prismatic joint in the *x*-axis while joint *h* is a prismatic joint in the *y*-axis.

In Table 22 the network matrices $[A_N]$ of the seed mechanism, the self-aligning mechanisms develop according to method I and the self-aligning mechanism with three Reshetov virtual joints added are compared. Matrix $[A_N]$ for the example mechanism has 42 lines rows and redundant constraints, as the screw system of the mechanism did not change. On the other hand, the number of columns is reduced from 43 to 40, due to the elimination of three redundant constraints. Matrix $[A_N]$ also has rank equal to the number of constraints as expected, because no redundant constraints are present.

Table 22 – Network Action Matrices Comparison Method II.

	Rows	Columns	Rank
Seed Mechanism	24	25	22
Mechanism with 3 Reshetov virtual joints	42	43	40
Example Mechanism	42	40	40

Source – From the author.

A CAD model is presented for the self-aligning example mechanism described in Figure 63. This model is shown in Figure 64. The proposed self-aligning metamorphic palm model presents some differences from the seed KCL/TJU hand as it was designed for 3D prototyping.

Figure 64 – New Self-Aligned Reconfigurable Anthropomorphic Hand



Source – From the author.

New joints have been added to links 1 and 4, more precisely each of the prismatic joints added to link 1 were placed on either side of the wrist.

The main advantages of a self-aligning mechanism with respect to an over-constrained one have been addressed in the previous sections. On the other hand, a reduction in mechanism rigidity can be expected, when redundant constraints are removed. The redundant constraints usually work as a stiffening force for over-constrained mechanisms, thus mechanism specifications must be carefully examined in order to design a proper mechanism.

8 MECHANISM SYNTHESIS METHODOLOGY

The objective of this section is to present the steps required for the mechanism synthesis methodology using the Davies' method and matroids.

Carboni (2015) presented a methodology to enumerate self-aligning mechanisms using the static modelling according to Davies' method and matroid theory, where the family of bases of this matroid corresponds to every combination possible of new mechanisms created from the seed mechanism. Following the static modelling of mechanisms presented in Chapter 3.3 and from the matrix $[A_N]$ creating a matroid over that matrix, the bases of this matroid are self-aligning mechanisms. On the other hand, the co-bases from said matroid are the constraints deleted from the seed overconstrained mechanism.

In the present thesis the mechanism enumeration concept is extended, from enumerating spatial self-aligning mechanisms to enumerating mechanisms in general. Suppose a seed mechanism composed only by Reshetov virtual joints is modelled using Davies' method. Using the matroid enumeration methodology proposed by Carboni (2015), every self-aligning mechanism derived from that seed mechanism is enumerated. Differently from the methodology proposed by Carboni, using Reshetov virtual joints enables the designer to create any kind of mechanism in a given workspace, without the limitation of the seed mechanism. The difference between the method proposed in this thesis and the works of Carboni is basically the impact on the kinematics of the mechanisms enumerated. Carboni enumerates mechanisms that are kinematically equivalent to the seed mechanism; the method proposed in this thesis enables the designer to create different kinematic structures for mechanisms.

The methodology starts by modelling a mechanism formed only by Reshetov virtual joints, which means that such mechanism does not have any freedoms. By creating a mechanism without freedoms and then creating a linear matroid on the static model of that mechanism we can enumerate all the combinations of constraints that create new mechanisms. The constraints that are removed from this overconstrained mechanism are the freedoms of that mechanism. Hence, by starting with a seed mechanism without freedoms we can enumerate all of the combinations of mechanisms possible for that seed mechanism.

Each joint of the kinematic chain needs a position for the screw theory representation. If the designer knows approximately where each joint can be placed, he can use those positions, but it is not required. The designer may use random positions with the following restrictions:

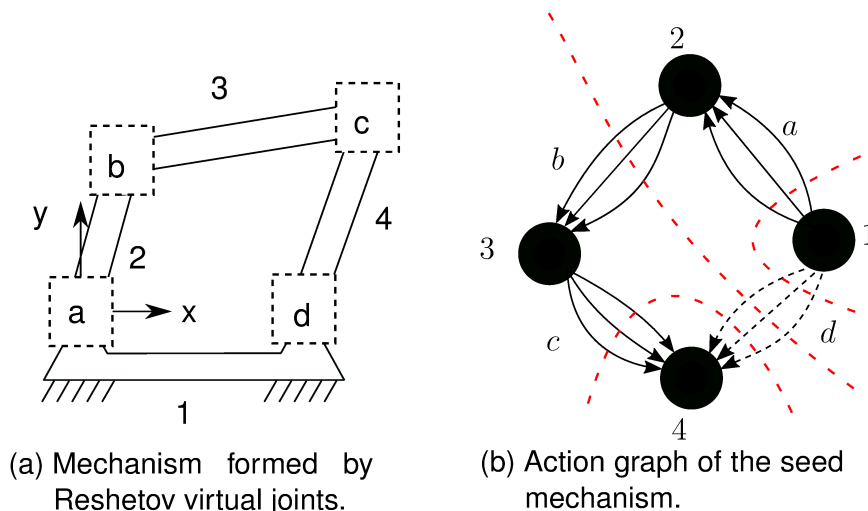
- Screw system dimensions: if the mechanism is planar, all the joints needs to be placed in the same plane;

- Avoid link alignment: the designer should also avoid choosing initial positions for the joints that enable two links with 0° or 180° between them;

The first restriction is very important for the methodology and needs to be respected always, while the other is not required and will only eliminate some possible mechanisms due to linear dependence. This linear dependence would cause the mechanism to be created in a singular position, which the bases enumerated by a matroid does not allow.

We begin the demonstration with a planar mechanism with four links and four joints. To simplify our explanation, we will consider only freedoms and constraints in the planar screw system ($\lambda = 3$). Instead of creating a mechanism with freedoms and constraints, we create a seed mechanism constituted by Reshetov virtual joints without any freedoms, Figure 65a. We then create the action graph of the seed mechanism, shown in Figure 65b.

Figure 65 – Seed mechanisms constituted by Reshetov virtual joints.



Source – From the author.

The action graph of Figure 65b has three edges between each pair of vertices, as there are three constraints in the planar screw system. The next step is creating joints and represent those joints using screw theory. Recall that the joints modelled here are joints with zero degrees of freedom, *i.e.*, only constraints and no freedoms. These constraints are arranged in action screws $\$_{i,j}^a$, each action screw related to the i coupling and the j mode of constraint. The action screws for the joints of the mechanism are organized in the unit action matrix $[A_D]$, with each constraint as a column.

$$\begin{aligned}
[A_D]_{6,12} = & \begin{bmatrix} \$_{aT}^a & \$_{aU}^a & \$_{aV}^a & \dots & (a) \\ \dots & \$_{bT}^a & \$_{bU}^a & \$_{bV}^a & \dots & (b) \\ \dots & \$_{cT}^a & \$_{cU}^a & \$_{cV}^a & \dots & (c) \\ \dots & \$_{dT}^a & \$_{dU}^a & \$_{dV}^a & \dots & (d) \end{bmatrix} \quad (45)
\end{aligned}$$

In Equation 45 is shown all of the constraints and the respective joints of the mechanism being modelled, which means that this matrix has twelve columns. The matrix $[A_D]$ has all of the constraints in the system.

In Figure 65b, the edges from joint d were chosen as chords. Applying Kirchhoff's cutset law to the action graph of 65b, the cutset matrix $[Q_A]$ is created. Using the matrices $[A_D]$ and $[Q_A]$ we create the matrix $[A_N]$.

Given the network unit matrix $[A_N]$, we can find a linear matroid \mathcal{M}_{A_N} whose groundset E is defined from the columns of matrix $[A_N]$. A basis B from the matroid \mathcal{M}_{A_N} will represent the model of a mechanism without redundant constraints. For a planar mechanism with four links and four joints, the rank of the matroid \mathcal{M}_{A_N} is nine, which means that the rank of the dual matroid $\mathcal{M}_{A_N}^*$ is three. At a first glance, the rank three of the dual matroid seems strange as there are four joints so one can deduce that the rank should also be four; however, in the Davies' method the actuation force or moment is also considered in the static analysis of the mechanism. In fact, the actuation of the joints are also constraints in Davies' method. Therefore, each of the cobases found for the dual matroid $\mathcal{M}_{A_N}^*$ will represent a different distribution of freedoms for the joints, and each will represent a different mechanism.

The matroid \mathcal{M}_{A_N} for the planar mechanism with four links and four joints has 116 different bases. A method for filtering bases is beneficial for a designer, providing assistance in choosing the best bases according to different design requirements. With filters the designer is not required to review and evaluate every basis, only those that seem viable according to a set of design requirements. For the current mechanism some requirements are proposed:

- Joint a shall have 1 freedom and that freedom is actuated;
- Joint b shall be revolute;

Now we can begin the filtering procedure which will use contraction and deletion of matroids. When contracting a constraint from matroid \mathcal{M}_{A_N} , the rank of this matroid will decrease by one. The element contracted will no longer be present in the bases, however in our mechanisms application it will mean that the contracted constraint will be present in the mechanism. The deletion is a little different. When deleting a constraint from the matroid \mathcal{M}_{A_N} , the rank of the matroid will not change, only the dual matroid

is affected by this operation. Hence, a deleted constraint will be removed from the dual matroid. This means that a deleted constraint is defined as a freedom for the mechanism. Before beginning the matroid operation, the joints and their respective types are organized in Figure 66.

Figure 66 – Joint types of the seed mechanism composed by Reshetov virtual joints.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
Seed Mechanism	<i>VI</i>	<i>VI</i>	<i>VI</i>	<i>VI</i>

Source – From the author.

The mechanism shown in Figure 65b has only type *VI* joints as they are all Reshetov virtual joints without freedoms. The mechanism selection process starts by defining joint *a* as actuated. The actuation of the joints is part of the static analysis of Davies' method, therefore the actuated freedom must be present in the basis of the matroid \mathcal{M}_{A_N} . Recalling Section 5.5, the contraction of an element of the linear matroid can be considered as selecting only the bases in which that element is present and then eliminating that element from the matroid; therefore by contracting a constraint, the designer selects only bases that have that element while decreasing the size of the ground set of the matroid. Joint *a* was defined as actuated with only one freedom, thus, for our purposes all of the constraints of this joint must be contracted. As joint *a* is now known, it no longer is a type *VI* joint, but instead a type V_r with a revolute joint with an actuated freedom, all constraints (*T*, *U* and *V*) are contracted. The representation of the contraction of the constraints and the change in joint type is shown in Figure 67. The new matroid $\mathcal{M}_{A_{N_1}}$ is created from the contraction of these constraints and has rank 6 with 46 different bases. This means that there are 46 mechanism possible with joint *a* as the actuated revolute joint.

Figure 67 – Operation diagram for contraction of the constraints of joint *a*.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
Seed Mechanism	<i>VI</i>	<i>VI</i>	<i>VI</i>	<i>VI</i>
First Step	V_r	<i>VI</i>	<i>VI</i>	<i>VI</i>

Source – From the author.

Now we will use the second requirement to continue filtering the results. Joint *b*

must be a revolute joint, which means that we need to delete the moment constraint of joint b . The deletion of an element, as discussed in Section 5.4, selects only a set of cobases in which the element is present and then eliminates that element from the cobases, thus selecting bases based on the absence of a constraint in the mechanism. By doing this operation from the original matroid \mathcal{M}_{A_N} , we arrive at a new matroid with 30 different bases, whose rank is 9 and the rank of the cobases are 2. However, let us return to matroid $\mathcal{M}_{A_{N_1}}$ that we found after contracting the constraints of joint a . If we delete the same constraint from that matroid, which is shown in Figure 68, we have a new matroid $\mathcal{M}_{A_{N_2}}$ with only 13 different bases. This means that after these two steps we managed to filter from 116 bases initially to only 13 bases. This also means that we have 13 different mechanisms that have the same structure of the seed mechanism but with different joint combinations.

Figure 68 – Operation diagram for the deletion of constraint T and contraction of the constraints U and V of joint b .

	a	b	c	d
Seed Mechanism	VI	VI	VI	VI
First Step	V_r	VI	VI	VI
Second Step	V_r	${}^T V_r$	VI	VI

Source – From the author.

Before choosing one or a few solutions from the 13 remaining bases, there is another important characteristic of the methodology to be discussed. As we use matroids to evaluate the matrix $[A_N]$, there are bases that distribute multiple freedoms to a single joint while another joint does not receive any freedoms. This characteristic enables the designer to create mechanisms constituted by joints with multiple freedoms instead of multiple single degree joints. This means that joints such as pin-in-slot can be found by the matroid bases enumeration. Using these bases or avoiding them will depend on the design requirements of the mechanism. Table 23 presents a summary of the steps taken in the synthesis.

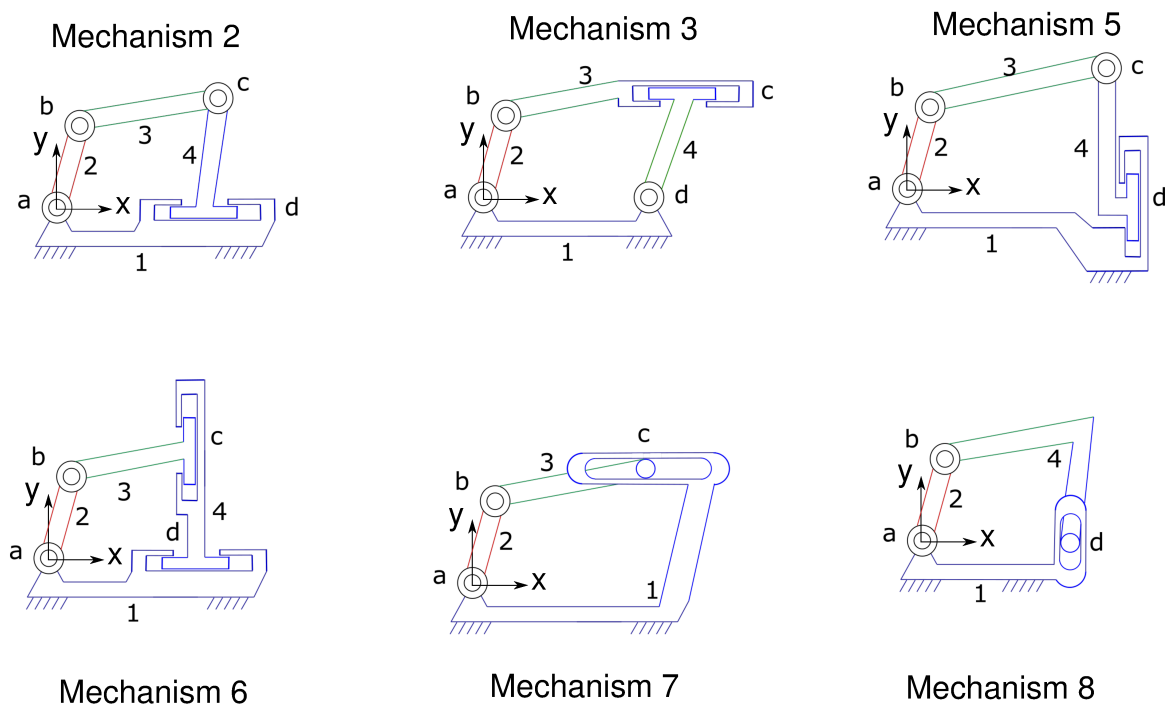
Table 23 – Summary of the Steps in the Filtering Process for the Example Mechanism.

Joint	Operation	Matroid operation	Remaining bases
All	creation of linear matroid	$\mathcal{M}_{A_N} = ([A_N], \mathcal{B}_{A_N})$	116
<i>a</i>	contract to guarantee revolute joint	$\mathcal{M}_{A_{N_1}} = \mathcal{M}_{A_N} / \{a_T, a_U, a_V\}$	46
<i>b</i>	delete to guarantee revolute joint	$\mathcal{M}_{A_{N_2}} = \mathcal{M}_{A_{N_1}} / \{b_T\}$	30
<i>b</i>	contract to guarantee revolute joint	$\mathcal{M}_{A_{N_3}} = \mathcal{M}_{A_{N_2}} / \{b_U, b_V\}$	13

Source – From the author.

Even though the mechanism's actuation does not matter in the mechanism synthesis presented in this work, for the planar mechanism with four joints and four links we defined the actuation as a revolute motor in joint *a*. Therefore, we can list all the possible mechanisms according to our synthesis methodology, which are presented in Appendix D.3. Some examples are presented in Figure 69, using the numbers from the table shown in the appendix.

Figure 69 – Examples of New Mechanisms Enumerated Using Davies' Method and Matroid Theory



Source – From the author.

Of the 13 new possible mechanisms, 7 have one mobility at each joint, while 6 have one joint with two freedoms. The examples shown consider only revolute, prismatic and pin-in-slot joints; however different types of joints can be considered, such as gear or cam pairs. Now that the designer has the combinations of joints for new mechanisms,

one should be chosen for the dimensional synthesis. It is worth to mention that although prismatic joints were defined in a specific axis, this is used only as a starting point for the dimensional synthesis process.

The methodology presented in this section will now be applied in the mechanism synthesis process of the development for new mechanisms of a hospital bed.

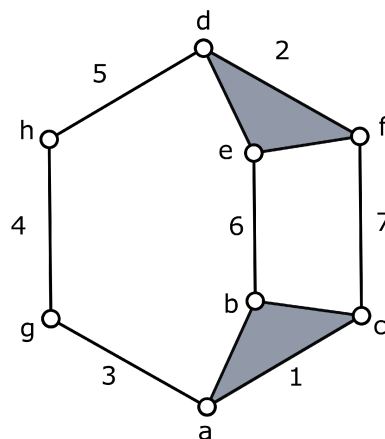
9 DEVELOPMENT OF HOSPITAL BED MECHANISMS

In this chapter, the development of new hospital bed mechanisms is presented. The work is divided into two parts, for each of the mechanism developed: leg rest mechanism and the backrest mechanism. This is done because before we can apply the mechanism synthesis methodology for finding new hospital bed mechanisms, we need to find the kinematic chains of these mechanisms and discuss the design requirements for each mechanism.

9.1 LEG REST SUPPORT MECHANISM

The kinematic chain for the leg rest mechanism, shown in Figure 70, was chosen based on a state-of-the-art survey (BARRETO et al., 2017) as well as user, design and functional requirements (MALETZ et al., 2017).

Figure 70 – Kinematic Chain for the Leg Rest Support Mechanism.



Source – From the author.

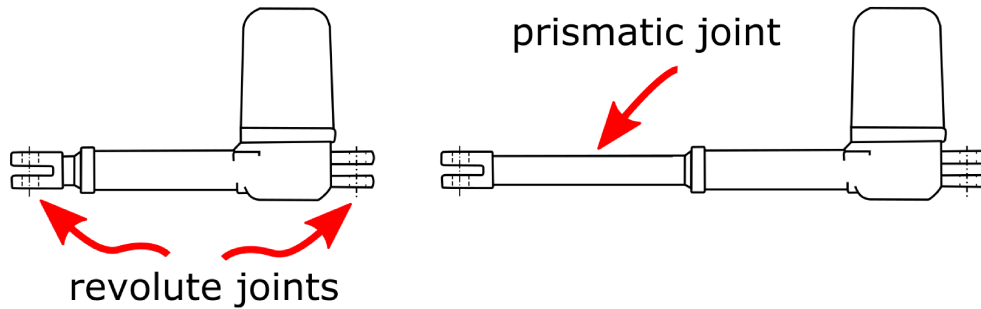
The kinematic chain from Figure 70 has two inner loops and mobility $M = 2$, $j = 8$ joints and $n = 7$ links. Using a technique called actuated degree of control (MURAI et al., 2019) and several design requirements (BARRETO et al., 2017) we define joints e and g as the actuated joints while links 2 and 5 are chosen as output links for the upper and lower leg support, respectively. The actuated degree of control calculates the actuators that affect the relative position between any 2 links of a kinematic chain with actuators. Using the same technique, link 1 was established as fixed link while joint d was established as a revolute joint. After choosing the actuated joints, output and input links, the following design requirements are created:

- Actuation by off-the-shelf commercial prismatic joints;
- Whenever possible use RPR structure with the actuator;

- Joint d must be revolute.

The first design requirement was created due to the availability of off-the-shelf commercial actuators for hospital bed applications. The general structure of these actuators is presented in Figure 71:

Figure 71 – Structure of off-the-shelf commercial actuator for hospital bed applications



Source – Adapted from Artmann (2019).

The actuator from Figure 71 is a linear actuator with two fixation points. The linear actuator is equivalent to a prismatic joint while the fixations are equivalent to revolute joints. Hence, these actuators may be considered RPR structures.

The second design requirement was created to use the actuator RPR structure when possible to facilitate the mechanism design, which means blocking the revolute joints of the fixation only when absolutely needed. The third design requirement came from actuated degree of freedom technique. After defining the design requirements, we can start the mechanism synthesis methodology.

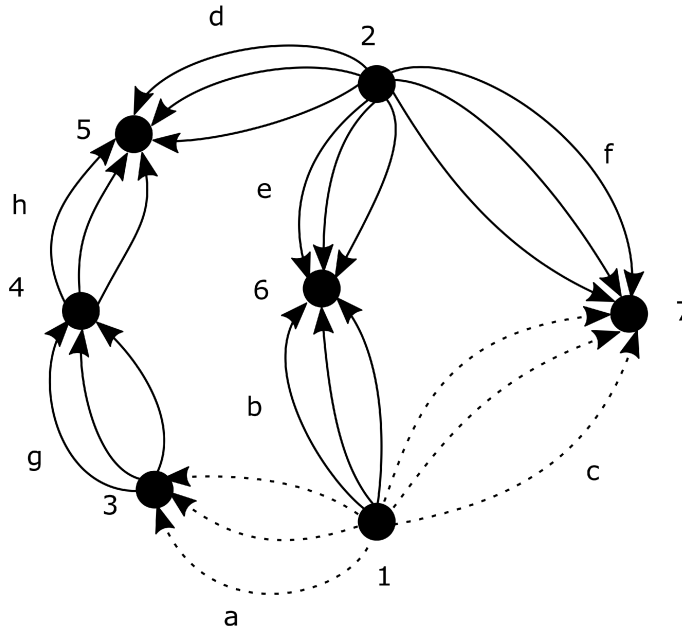
In order to begin the mechanism synthesis, we create random points for each of the zero DOF joints in the mechanism, and then we create matrix $[A_{D_{leg}}]$, shown in Equation 46.

$$\begin{aligned}
 [A_{D_{leg}}]_{6,21} = & \begin{bmatrix}
 \$_{aT}^a & \$_{aU}^a & \$_{aV}^a & \dots & (a) \\
 \dots & \$_{bT}^a & \$_{bU}^a & \$_{bV}^a & \dots & (b) \\
 \dots & \$_{cT}^a & \$_{cU}^a & \$_{cV}^a & \dots & (c) \\
 \dots & \$_{dT}^a & \$_{dU}^a & \$_{dV}^a & \dots & (d) \\
 \dots & \$_{eT}^a & \$_{eU}^a & \$_{eV}^a & \dots & (e) \\
 \dots & \$_{fT}^a & \$_{fU}^a & \$_{fV}^a & \dots & (f) \\
 \dots & \$_{gT}^a & \$_{gU}^a & \$_{gV}^a & \dots & (g) \\
 \dots & \$_{hT}^a & \$_{hU}^a & \$_{hV}^a & \dots & (h)
 \end{bmatrix}
 \end{aligned}$$

(46)

The matrix $[A_{D_{leg}}]$ is multiplied by the diagonal matrices created from the rows of the cutset matrix $[Q_{A_{leg}}]$, which is created from the cutset law applied to the action graph of Figure 72. The joints positions and cutset matrix is shown in Appendix A.2.

Figure 72 – Action graph of the seed mechanism for the leg mechanism.



Source – From the author.

In Figure 72, the edges a and c were chosen as the chords. The network unit matrix $[A_{N_{leg}}]$ created from the matrices $[A_{D_{leg}}]$ and $[Q_{A_{leg}}]$ has 24 columns. With the matrix $[A_{N_{leg}}]$ we can now create the linear matroid \mathcal{M}_l over the span created by the columns of $[A_{N_{leg}}]$.

The matroid \mathcal{M}_l has 25566 different bases, and as mentioned before, each represents a different mechanism. Before starting the filtering process, the diagram with the joint types is created and shown in Figure 73.

Figure 73 – Joint types of the seed mechanism for the leg-rest mechanism composed only by Reshetov virtual joints.

	a	b	c	d	e	f	g	h
Seed Mechanism	VI	VI	VI	VI	VI	VI	VI	VI

Source – From the author.

As mentioned in the previous section, the actuating force or moment is considered as part of the static in the Davies' method, therefore the constraints U regarding forces in the x-axis should be contracted. As the actuators are commercial items, the

constraints T and V also should be contracted since we cannot make changes to the actuators. Therefore, the constraints T, U and V for the actuated joints e and g are contracted ($\mathcal{M}_{l_1} = \mathcal{M}_l / \{e_U, e_V, e_T, g_U, g_V, g_T\}$). This first step is shown in Figure 74. This operation provides us a new matroid with 3742 bases.

Figure 74 – Operation diagram for the contraction of the constraints T, U and V of joints e and g .

	a	b	c	d	e	f	g	h
Seed Mechanism	VI	VI	VI	VI	VI	VI	VI	VI
First Step	VI	VI	VI	VI	V_p	VI	V_p	VI

Source – From the author.

The types of joints e and g are now type V_P as they were defined as actuated prismatic joints. According to the third design requirement, joint d needs to be a revolute joint, so we can delete the constraint T and contract the remaining constraints guaranteeing a purely revolute joint ($\mathcal{M}_{l_2} = \mathcal{M}_{l_1} / \{d_U, d_V\}$ and $\mathcal{M}_{l_3} = \mathcal{M}_{l_2} \setminus \{d_T\}$). The first operation shown in Figure 75 delivers a matroid with 1521 bases while the contraction, shown in Figure 76, returns a matroid with 1005 bases.

Figure 75 – Operation diagram for the deletion of constraint T of joint d .

	a	b	c	d	e	f	g	h
Seed Mechanism	VI	VI	VI	VI	VI	VI	VI	VI
First Step	VI	VI	VI	VI	V_p	VI	V_p	VI
Second Step	VI	VI	VI	TV	V_p	VI	V_p	VI

Source – From the author.

Figure 76 – Operation diagram for the contraction of the constraints U and V of joint d .

	a	b	c	d	e	f	g	h
Seed Mechanism	VI	VI	VI	VI	VI	VI	VI	VI
First Step	VI	VI	VI	VI	V_p	VI	V_p	VI
Second Step	VI	VI	VI	${}^T V$	V_p	VI	V_p	VI
Third Step	VI	VI	VI	${}^T V_r$	V_p	VI	V_p	VI

Source – From the author.

For further filtering we turn to the second design requirement, which states that whenever possible we should use RPR structures with the actuator. As we defined the actuators at joints e and g , joints a , b and h can be used in the next filter. These joints will be formed by the fixation of the commercial actuator, then we can maintain the revolute freedom of the joints, which means deleting the constraints T from those joints. We now create a new matroid $\mathcal{M}_{l_4} = \mathcal{M}_{l_3} \setminus \{a_T, b_T, h_T\}$, this step is shown in Figure 77. This matroid has only 22 bases, from those bases 7 represent mechanisms with one freedom at each joint while 15 bases represent mechanisms with joints possessing multiple freedoms.

Figure 77 – Operation diagram for the deletion of constraint T of joints a , b and h .

	a	b	c	d	e	f	g	h
Seed Mechanism	VI	VI	VI	VI	VI	VI	VI	VI
First Step	VI	VI	VI	VI	V_p	VI	V_p	VI
Second Step	VI	VI	VI	${}^T V$	V_p	VI	V_p	VI
Third Step	VI	VI	VI	${}^T V_r$	V_p	VI	V_p	VI
Fourth Step	${}^T V$	${}^T V$	VI	${}^T V_r$	V_p	VI	V_p	${}^T V$

Source – From the author.

Table 24 presents a summary of the steps taken in the bases filtering:

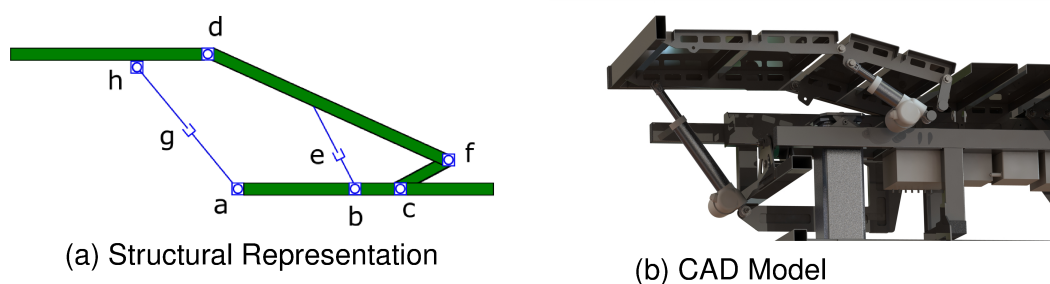
Table 24 – Summary of the steps in the filtering process for the leg-rest mechanism.

Joint	Operation	Matroid operation	Remaining bases
all	creation of linear matroid	$\mathcal{M}_l = ([A_{N_{leg}}], \mathcal{B}_l)$	25566
e and g	contract U, V and T	$\mathcal{M}_{l_1} = \mathcal{M}_l / \{e_U, u_V, e_T, g_U, g_V, g_T\}$	3742
d	delete T	$\mathcal{M}_{l_2} = \mathcal{M}_{l_1} / \{d_U, d_V\}$	1521
d	contract U and V	$\mathcal{M}_{l_3} = \mathcal{M}_{l_2} \setminus \{d_T\}$	1005
a, b and h	delete T	$\mathcal{M}_{l_4} = \mathcal{M}_{l_3} \setminus \{a_T, b_T, h_T\}$	22

Source – From the author.

The possible mechanisms found are presented in Appendix D.4. Mechanism 19 was chosen, Figure 78a, and a CAD model was created, Figure 78b.

Figure 78 – Leg Rest Mechanism Developed



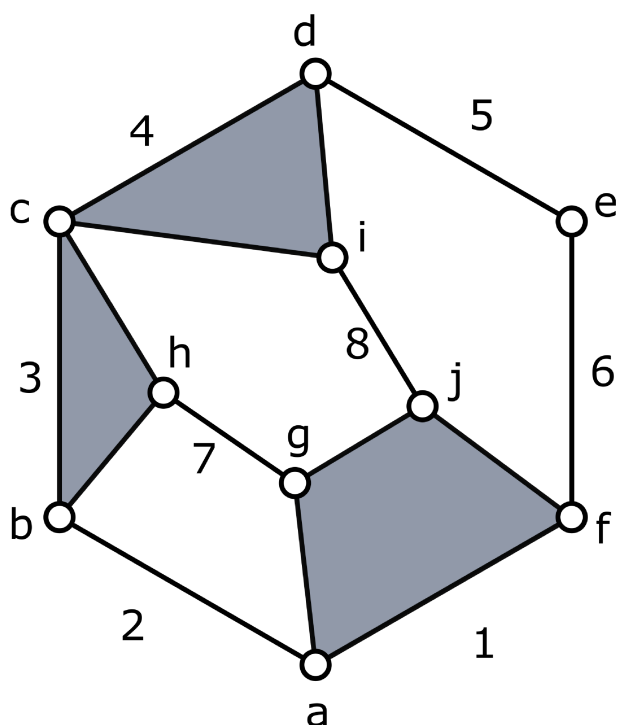
Source – From the author.

The CAD model from Figure 78b was manufactured and is present in the Laboratory of Applied Robotics' hospital bed.

9.2 BACKREST ELEVATION MECHANISM

Similar to the kinematic chain of the leg rest mechanism, the kinematic chain for the backrest elevation mechanism was chosen based on a survey of existing mechanisms and is presented in Figure 79.

Figure 79 – Kinematic Chain for the Backrest Support Mechanism.



Source – From the author.

This kinematic chain has three inner loops and mobility $M = 1$, the number of joints is $j = 10$ with $n = 8$ links. The link 1, the quaternary link, was chosen as the fixed link, while link 3, a ternary link, was chosen as the output link. For the backrest mechanism, the following requirements were used established:

- Joint e is the actuated joint;
- Actuation by off-the-shelf commercial linear actuator;
- When possible use RPR structure with the actuator;
- Joints f and d must present at least a revolute freedom to comply with the RPR structure;
- Joints a, b, g and h should be revolute - this creates a sub-circuit constituted by a four-bar mechanism, whose kinematic modeling is a well known problem;

Following the procedure presented previously, the synthesis of the backrest mechanism was performed. The cutset matrix and joints positions are shown in Appendix A.3. Table 25 presents the steps taken in filtering the bases during the synthesis of this mechanism.

Table 25 – Summary of the steps in the filtering process for the backrest mechanism.

Joint	Operation	Matroid operation	Remaining bases
all	creation of linear matroid	$\mathcal{M}_b = ([A_{N_{back}}], \mathcal{B}_b)$	1344797
e	contract U, V and T	$\mathcal{M}_{b_1} = \mathcal{M}_b / \{e_U, u_V, e_T\}$	529707
d and f	delete T	$\mathcal{M}_{b_2} = \mathcal{M}_{b_1} \setminus \{d_T, f_T\}$	81665
a, b, g and h	delete T	$\mathcal{M}_{b_3} = \mathcal{M}_{b_2} \setminus \{a_T, b_T, g_T, h_T\}$	885
a, b, g and h	contract U and V	$\mathcal{M}_{b_4} = \mathcal{M}_{b_3} / \{a_U, a_V, b_U, b_V, g_U, g_V, h_U, h_V\}$	189

Source – From the author.

The cases selection method proposed by Artmann et al. (2019a) will now be used to select a smaller number of bases from the remaining 189 bases. Two selection criteria are proposed:

- Elimination of planar joints;
- Elimination of isomorphisms of prismatic joints on the x and y-axes;
- Elimination of joints with linear freedoms on the x and y-axes.

The first criterion is defined because planar joints, joint with three freedoms in the plane, are not usually desirable in a planar mechanism. The second criterion is defined in order to remove mechanisms with isomorphic solutions but with different linear freedoms. For example, in the used method a joint with a linear freedom in the x-axis is different from a joint with linear freedom in the y-axis. Using contraction and deletion does not enable the elimination of such isomorphic results, thus the method proposed by Artmann is applied. Finally, the third criterion seeks to avoid hard to manufacture joints with two linear freedoms without planar freedoms. Recalling the co-bases selection method by Artmann:

$$n_{(i,j)} = \begin{cases} 1 & \text{if the constraint } j \text{ is in the cobasis } i \\ 0 & \text{otherwise} \end{cases} \quad (47)$$

Although some constraints were removed using contraction and deletion, the columns respective to these constraints will be considered in this step with null value when contracted and equal to one when deleted in order to simplify the column numbering. This is done to define easily which column correlates to which constraint of which joint as well as to return previously defined freedoms. As there were originally ten joints and each joint had three constraints, the j column of the co-bases binary matrix $n_{(i,j)}$ is easily correlated to a specific joint. From the contraction and deletion steps there were 189 remaining bases, each is related to a i row of the co-bases binary matrix.

The first criterion stated that planar joints should be removed, therefore in the co-bases it is not allowed three freedoms in the same joint. The criterion K_1 is thus defined:

$$\{\forall i = 1, 2, \dots, 189 | B_i^* \in K_1 \Leftrightarrow \{n(i, 3k+1) + n(i, 3k+2) + n(i, 3k+3) < 3\}, k = 0, 1, 2, \dots, 9\}. \quad (48)$$

Applying criterion K_1 returned 187 co-bases, which means there were two mechanisms with planar joints.

The second criterion is used to removed isomorphic co-bases when considering linear freedoms in the x and y-axes. The verification for the second criterion is done by comparing the $n(i, 3k + 1)$ and $n(i, 3k + 2)$ bases with every other $n(r, 3k + 1)$ and $n(r, 3k + 2)$ bases, for $r \neq i$ and $k = 0, 1, 2, \dots, 9$. If a isomorphism is noticed, that basis is removed from the criteria K_2 . After comparing the elements described, criterion K_2 has a set of 78 co-bases.

The third criterion is used to remove joints with multiple linear freedoms. The criterion K_3 is thus defined:

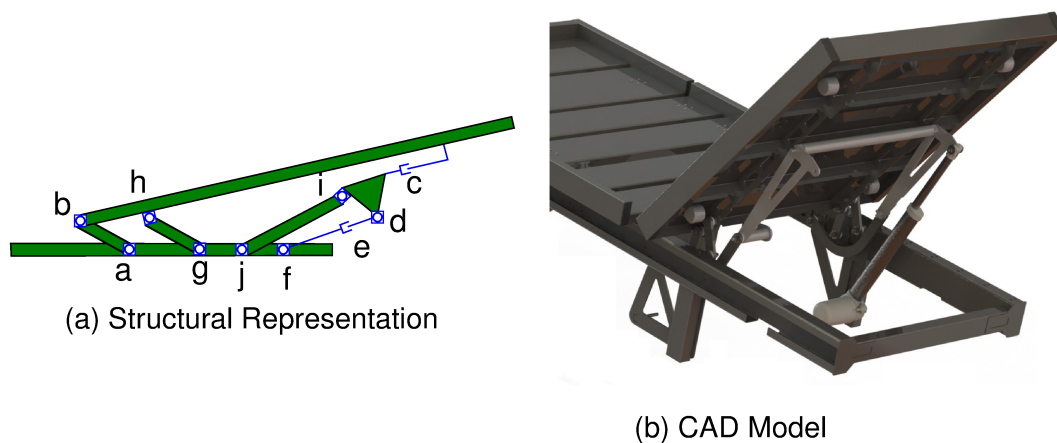
$$\{\forall i = 1, 2, \dots, 189 | B_i^* \in K_3 \Leftrightarrow \{n(i, 3k + 2) + n(i, 3k + 3) < 2\}, k = 0, 1, 2, \dots, 9\}. \quad (49)$$

A set of 60 cobases comply with the criterion K_3 .

After applying the three criteria, the next step is intersecting the three sets to check which cobases are present in both. The intersection resulted in a set of 55 mechanisms. The result means that there are 55 mechanisms which comply with both criteria. The set of mechanisms is shown in Appendix D.5.

One of the selected mechanisms using Artmann's method was chosen for further manufacturing, shown in Figure 80, with the structural representation shown in Figure 80a and the CAD model shown in Figure 80b.

Figure 80 – Backrest Mechanism Developed



Source – From the author.

The CAD model from Figure 80b was manufactured and is present in the LAR hospital bed.

10 CONCLUSION

One of the biggest challenges in mechanism design is defining the types of joints. Poor choices may lead to undesired singularities or manufacturing challenges. Moreover, when defining the types of the joints for a mechanism, the designer must also seek to develop new and patentable mechanisms; therefore, enumeration based mechanism design methodologies are useful for achieving all of these requirements.

Screw theory, a powerful mathematical tool for representing forces moments and velocities, is widely used when analysing mechanisms. Using screw theory, Davies (1981) proposed a method for analysing multibody systems. Based on the formulation by Davies, Carboni (2015) proposed a method for enumerating self-aligning mechanisms using matroid theory.

This thesis proposed a method for selecting self-aligning mechanisms enumerated using matroid theory. The proposed method is simple to use, relying on open-source software and uses only already available libraries. The hardest tasks regarding the proposed method is creating a model of a seed mechanism according to Davies' method and listing design requirements.

The proposed method of selecting self-aligning mechanisms in fact is a method of decreasing the existing possibilities by eliminating certain constraints from the matroid. By decreasing the ground set of the matroid, the problem of the huge number of results is solved, thus making solvable systems that would otherwise be impossible due to the computer time required.

This thesis proposed a concept called Reshetov virtual joints for enumerating self-aligning mechanisms. This concept was used with Davies' method for modelling a mechanism, then using matroid theory to enumerate every possible self-aligning mechanism based on the seed mechanisms. The concept of Reshetov virtual joints increases the flexibility of Davies' method, providing the designer with another tool for mechanism analysis and design.

Using Reshetov virtual joints, the self-aligning mechanisms methodology proposed by Carboni (2015) and the mechanisms selection method applying contraction and deletion of matroids, self-aligning mechanisms based on the anthropomorphic reconfigurable palm of the KCL/TJU hand were proposed. Two different methods were compared, one using the actual palm mechanism as the seed mechanism, and the other introducing Reshetov virtual joints for receiving freedoms. For the first method, initially 2066 mechanisms were enumerated and after the selection steps a set of 45 mechanisms were listed, representing only 2,2% of the original set. For the second method, there were 4242 mechanisms enumerated. The selection steps returned 33 new mechanisms, representing 0,78% of the original set.

This thesis proposed a method of enumerating self-aligning mechanisms by

using Davies' method, matroid theory and mechanisms constituted only by Reshetov virtual joints. Using these tools, the designer can create mechanisms with different kinematic structures, an improvement from the previous enumeration methodologies that used Davies' method and matroid theory.

Two hospital bed mechanisms were used as study case for the proposed self-aligning mechanism design methodology. First, the leg rest of the hospital bed was designed, initially returning 25566 mechanisms. Using the selection process, a set of 22 feasible mechanisms were found, representing 0,08% of the original set. For the backrest of the hospital bed, there were 1344797 possible mechanisms, and after the selection process a set of 189 mechanisms were found. The remaining mechanisms represent 0,014% of the original set. CAD model examples for both mechanisms were presented.

The self-aligning mechanisms enumeration method using Reshetov virtual joints does have some drawbacks. The method returns many results that are equivalent and sorting or selecting only one of these results are difficult. For improving the method, one solution is using the binary cobases matrix (ARTMANN, 2019) after a few steps of contraction and deletion. By using Artmann's method, we can try to avoid isomorphisms.

10.1 SUGGESTIONS FOR FUTURE WORK

Suggestions for future work include:

- To create an algorithm to use both matroid contraction and deletion as well as the binary cobases matrix proposed by Artmann (2019).
- To study and propose a method for selecting spatial self-aligning mechanisms designed based on the method proposed in this thesis. The challenge with this task is due to the huge number of mechanisms enumerated.
- To investigate different design requirements and how to apply these requirements in the method proposed in this thesis.
- To investigate other applications for the Reshetov virtual joints. The use of Reshetov virtual joints in Davies' method is not limited for generating new self-aligning mechanisms. One possible application is using Reshetov virtual joints for calculating forces and moments in a specific point of a rigid link.
- To investigate the matroid operation called union for mechanisms enumeration. The mechanism enumeration method proposed in this thesis can be divided into multiple analyses and then each of these analyses can be joined together using matroid union.

- To compare the kinematic and dynamic performance comparison between the seed mechanism and the self-aligning counterpart.

10.2 PUBLICATION LIST

The following papers were published in relation to the work presented in this thesis.

- ARTMANN, V. N.; BARRETO, R. L. P.; CARBONI, A. P.; SIMONI, R.; MARTINS, D. Type Synthesis of Self-Aligning Mechanisms Applied to the Leg Rest Section of an Hospital Bed. In: SPRINGER. IFTOMM World Congress on Mechanism and Machine Science. [S.l.: s.n.], 2019. p. 1413–1422.
- ARTMANN, V. N.; BARRETO, R. L. P.; MARTINS, D.; SIMONI, R. Analysis and elimination of redundant constraints in clamping devices. In: ABCM. PROCEEDINGS 25th International Congress of Mechanical Engineering - COBEM 20-25 October. [S.l.: s.n.], 2019. p. 147–161.
- BARRETO, R. L. P.; CARBONI, A. P.; SIMONI, R.; MARTINS, D., DAI, J. S. Self-aligning Analysis of the Metamorphic Palm of the KCL/TJU Metamorphic Hand. In: IEEE. 2018 International Conference on Reconfigurable Mechanisms and Robots (ReMAR). [S.l.: s.n.], 2018. p. 1–8.
- BARRETO, R. L. P.; CARBONI, A. P.; SIMONI, R.; MARTINS, D. Method for selecting self-aligning mechanisms enumerated by matroid contractions. Accepted for publication in the 17th International Symposium on Advances in Robot Kinematics, 2020.
- BARRETO, R. L. P.; MALETZ, E. R.; MOLGARO, A. L.; BRITO, J. V. F.; MARTINS, D.. A Method to Integrate Anthropometry to Davies' Method for Dimensional Synthesis of a Hospital Bed Backrest Mechanism. In: ABCM. PROCEEDINGS 25th International Congress of Mechanical Engineering - COBEM 20-25 October. [S.l.: s.n.], 2019.
- BARRETO, R. L. P.; SIMONI, R.; MARTINS, D. An Initial Assessment of Mechanisms for the Development of New Hospital Beds. In: SPRINGER. INTERNATIONAL Symposium on Multibody Systems and Mechatronics. [S.l.: s.n.], 2017. p. 485–494.
- MALETZ, E. R.; BARRETO, R. L. P.; MARTINS, D. Revisão do Estado da Arte de Camas Hospitalares: Análise de Mercado e Pesquisa de Patentes. In: 10 Congresso Brasileiro de Pesquisa e Desenvolvimento em Tecnologia Assistiva: Engenharia e Deseign. Curitiba - Brasil. [S.l.: s.n.], 2016.

- MALETZ, E. R.; SIMAS, H.; BARRETO, R. L. P.; MARTINS, D. Contribution to the design of hospital bed: systematic for surveying the design requirements and functional requirements for synthesis of mechanism. In: SPRINGER. INTERNATIONAL Conference on Applied Human Factors and Ergonomics. [S.l.: s.n.], 2017. p. 652–662.

REFERENCES

- ARTMANN, Vinicius N. **A New Method to Select Self-Aligning Mechanisms Enumerated by Matroid: Case Studies in Hospital Beds**. June 2019. MA thesis – Universidade Federal de Santa Catarina.
- ARTMANN, Vinicius N; BARRETO, Rodrigo L P; CARBONI, Andrea P; SIMONI, Roberto; MARTINS, Daniel. Type Synthesis of Self-Aligning Mechanisms Applied to the Leg Rest Section of an Hospital Bed. In: SPRINGER. IFTOMM World Congress on Mechanism and Machine Science. [S.l.: s.n.], 2019a. P. 1413–1422.
- ARTMANN, Vinicius N; BARRETO, Rodrigo L P; MARTINS, Daniel; SIMONI, Roberto. Analysis and elimination of redundant constraints in clamping devices. In: ABCM. PROCEEDINGS 25th International Congress of Mechanical Engineering - COBEM 20-25 October. [S.l.: s.n.], 2019b. P. 147–161.
- BALL, Robert Stawell. **A Treatise on the Theory of Screws**. Cambridge: Cambridge, 1900. ISBN 0521636507.
- BARRETO, Rodrigo L P; CARBONI, Andrea P; SIMONI, Roberto; DAI, Jian S; MARTINS, Daniel. Self-aligning Analysis of the Metamorphic Palm of the KCL/TJU Metamorphic Hand. In: IEEE. 2018 International Conference on Reconfigurable Mechanisms and Robots (ReMAR). [S.l.: s.n.], 2018. P. 1–8.
- BARRETO, Rodrigo L P; SIMONI, Roberto; MARTINS, Daniel. An Initial Assessment of Mechanisms for the Development of New Hospital Beds. In: SPRINGER. INTERNATIONAL Symposium on Multibody Systems and Mechatronics. [S.l.: s.n.], 2017. P. 485–494.
- BELFIORE, Nicola Pio; PENNESTRI, Ettore. An atlas of linkage-type robotic grippers. **Mechanism and Machine Theory**, Elsevier, v. 32, n. 7, p. 811–833, 1997.
- BLANDING, D. L. **Exact Constraint: Machine Design Using Kinematic Principles**. New York: ASME, 1999.
- CAMPOS, Alexandre; BUDDE, Christoph; HESSELBACH, Jurgen. A type synthesis method for hybrid robot structures. **Mechanism and Machine Theory**, Elsevier, v. 43, n. 8, p. 984–995, 2008.
- CARBONI, Andrea P. **Análise de mecanismos com restrições redundantes através da aplicação da teoria de matroides**. Outubro 2015. PhD thesis – Universidade Federal de Santa Catarina.
- CARBONI, Andrea P; SIMAS, Henrique; MARTINS, Daniel. Analysis of self-aligning mechanisms by means of matroid theory. In: SPRINGER. INTERNATIONAL Symposium on Multibody Systems and Mechatronics. [S.l.: s.n.], 2017. P. 61–73.

CARRICATO, Marco; ZLATANOV, Dimiter. Persistent screw systems. **Mechanism and Machine Theory**, Elsevier, v. 73, p. 296–313, 2014.

CHEN, Fan Yu. Gripping mechanisms for industrial robots: an overview. **Mechanism and Machine Theory**, Elsevier, v. 17, n. 5, p. 299–311, 1982.

CUI, Lei; DAI, Jian S; WANG, De Lun. Workspace analysis of a multifingered metamorphic hand. In: IEEE. RECONFIGURABLE Mechanisms and Robots, 2009. ReMAR 2009. ASME/IFTOMM International Conference on. [S.l.: s.n.], 2009. P. 589–595.

DAI, Jian S. **Robotic hand with palm section comprising several parts able to move relative to each other**. [S.l.: s.n.], Jan. 2008. US Patent App. 11/587,766.

DAVIDSON, Joseph K; HUNT, Kenneth H. **Robots and screw theory: applications of kinematics and statics to robotics**. New York: Oxford, 2004. ISBN 0198562454.

DAVIES, Trevor H. Couplings, coupling networks and their graphs. **Mechanism and Machine Theory**, Elsevier, v. 30, n. 7, p. 991–1000, 1995.

DAVIES, Trevor H. Dual coupling networks. **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science**, SAGE Publications Sage UK: London, England, v. 220, n. 8, p. 1237–1247, 2006a.

DAVIES, Trevor H. Freedom and constraint in coupling networks. **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science**, SAGE Publications Sage UK: London, England, v. 220, n. 7, p. 989–1010, 2006b.

DAVIES, Trevor H. Freedom and Constraint in coupling networks. **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science**, v. 220, n. 7, p. 989–1010, 2006c.

DAVIES, Trevor H. Kirchhoff's circulation law applied to multi-loop kinematic chains. **Mechanism and machine theory**, Elsevier, v. 16, n. 3, p. 171–183, 1981.

DING, H. **Automatic structural synthesis of planar mechanisms and its application to creative design**. 2015. PhD thesis – Universitat Duisburg-Essen.

ERDMAN, Arthur G; THOMPSON, Thomas; RILEY, Donald R. Type selection of robot and gripper kinematic topology using expert systems. **The International journal of robotics research**, Sage Publications Sage CA: Thousand Oaks, CA, v. 5, n. 2, p. 183–189, 1986.

FANG, Yuefa; TSAI, Lung-Wen. Structure synthesis of a class of 3-DOF rotational parallel manipulators. **IEEE transactions on robotics and automation**, IEEE, v. 20, n. 1, p. 117–121, 2004.

FANG, Yuefa; TSAI, Lung-Wen. Structure synthesis of a class of 4-DoF and 5-DoF parallel manipulators with identical limb structures. **The international journal of Robotics Research**, SAGE Publications Sage UK: London, England, v. 21, n. 9, p. 799–810, 2002.

FRENCH, Michael J. **Conceptual design for engineers**. [S.l.]: Springer, 1985.

GAO, Feng; LI, Weimin; ZHAO, Xianchao; JIN, Zhenlin; ZHAO, Hui. New kinematic structures for 2-, 3-, 4-, and 5-DOF parallel manipulator designs. **Mechanism and machine theory**, Elsevier, v. 37, n. 11, p. 1395–1411, 2002.

GAO, Feng; YANG, Jialun; GE, Qiaode Jeffrey. Type synthesis of parallel mechanisms having the second class GF sets and two dimensional rotations. **Journal of Mechanisms and Robotics**, American Society of Mechanical Engineers, v. 3, n. 1, p. 011003, 2011.

GOGU, Grigore. Chebychev–Grubler–Kutzbach’s criterion for mobility calculation of multi-loop mechanisms revisited via theory of linear transformations. **European Journal of Mechanics-A/Solids**, Elsevier, v. 24, n. 3, p. 427–441, 2005a.

GOGU, Grigore. Chebychev-Grübler-Kutzbach’s criterion for mobility calculation of multi-loop mechanisms revised via theory of linear transformations. **European Journal of Mechanics A/Solids**, v. 24, p. 427–411, 2005b.

GOGU, Grigore. Kinematic criteria for structural synthesis of maximally regular parallel robots with planar motion of the moving platform. In: **INTERDISCIPLINARY applications of kinematics**. [S.l.]: Springer, 2012. P. 63–81.

GOGU, Grigore. Mobility of mechanisms: a critical review. **Mechanism and machine Theory**, Elsevier, v. 40, n. 9, p. 1068–1097, 2005c.

GOGU, Grigore. Structural synthesis of fully-isotropic parallel robots with Schonflies motions via theory of linear transformations and evolutionary morphology. **European Journal of Mechanics-A/Solids**, Elsevier, v. 26, n. 2, p. 242–269, 2007.

GOGU, Grigore. Structural synthesis of maximally regular T3R2-type parallel robots via theory of linear transformations and evolutionary morphology. **Robotica**, Cambridge University Press, v. 27, n. 1, p. 79–101, 2009.

GROSS, Jonathan L; YELLEN, Jay; ZHANG, Ping. **Handbook of graph theory**. [S.l.]: Chapman and Hall/CRC, 2013.

GUO, Sheng; FANG, Yuefa; QU, Haibo. Type synthesis of 4-DOF nonoverconstrained parallel mechanisms based on screw theory. **Robotica**, Cambridge University Press, v. 30, n. 1, p. 31–37, 2012.

HARTENBERG, Richard; DENAVIT, Jacques. **Kinematic synthesis of linkages**. [S.l.]: New York: McGraw-Hill, 1964.

HE, Jun; GAO, Feng; MENG, Xiangdun; GUO, Weizhong. Type synthesis for 4-DOF parallel press mechanism using G F set theory. **Chinese Journal of Mechanical Engineering**, Springer, v. 28, n. 4, p. 851–859, 2015.

HERVE, Jacques M. Analyse structurelle des mécanismes par groupe des déplacements. **Mechanism and Machine Theory**, Elsevier, v. 13, n. 4, p. 437–450, 1978.

HUANG, Zhen; LI, Qinchuan. General methodology for type synthesis of symmetrical lower-mobility parallel manipulators and several novel manipulators. **The International Journal of Robotics Research**, SAGE Publications Sage UK: London, England, v. 21, n. 2, p. 131–145, 2002.

HUANG, Zhen; LI, Qinchuan. Type synthesis of symmetrical lower-mobility parallel mechanisms using the constraint-synthesis method. **The International Journal of Robotics Research**, SAGE Publications, v. 22, n. 1, p. 59–79, 2003.

HUANG, Zhen; LI, Qinchuan; DING, Huafeng. **Theory of Parallel Mechanisms**. Dordrecht: Springer, 2013. ISBN 978-94-007-4200-0.

HUNT, Kenneth H. **Kinematic Geometry of Mechanisms**. Oxford: Clarendon, 1978. (The Oxford engineering science series, v. 7).

KAMM, L.J. **Designing cost-efficient mechanisms: minimum constraint design, designing with commercial components, and topics in design engineering**. New York: McGraw-Hill, 1990.

KAROUIA, M; HERVE, Jacques M. A three-dof tripod for generating spherical rotation. In: **ADVANCES in robot kinematics**. [S.l.]: Springer, 2000. P. 395–402.

KHACHIYAN, Leonid; BOROS, Endre; ELBASSIONI, Khaled; GURVICH, Vladimir; MAKINO, Kazuhisa. On the complexity of some enumeration problems for matroids. **SIAM Journal on Discrete Mathematics**, v. 19, n. 4, p. 966–984, 2005. Available from: <http://rutcor.rutgers.edu/~boros/IDM/Papers/Cycles-in-matroids.pdf>.

KONG, Xianwen; GOSSELIN, Clement M. Type synthesis of 3-DOF PPR-equivalent parallel manipulators based on screw theory and the concept of virtual chain. **Journal of Mechanical Design**, American Society of Mechanical Engineers, v. 127, n. 6, p. 1113–1121, 2005a.

KONG, Xianwen; GOSSELIN, Clement M. Type synthesis of 3-DOF spherical parallel manipulators based on screw theory. **J. Mech. Des.**, v. 126, n. 1, p. 101–108, 2004a.

KONG, Xianwen; GOSSELIN, Clement M. Type synthesis of 4-DOF SP-equivalent parallel manipulators: A virtual chain approach. **Mechanism and machine theory**, Elsevier, v. 41, n. 11, p. 1306–1319, 2006.

KONG, Xianwen; GOSSELIN, Clement M. Type synthesis of 5-DOF parallel manipulators based on screw theory. **Journal of Robotic Systems**, Wiley Online Library, v. 22, n. 10, p. 535–547, 2005b.

KONG, Xianwen; GOSSELIN, Clement M. **Type synthesis of parallel mechanisms**. [S.l.]: Springer, 2007. v. 33.

KONG, Xianwen; GOSSELIN, Clement M. Type synthesis of three-degree-of-freedom spherical parallel manipulators. **The International Journal of Robotics Research**, SAGE Publications, v. 23, n. 3, p. 237–245, 2004b.

KUO, Chin-Hsing; DAI, Jian S. Task-oriented structure synthesis of a class of parallel manipulators using motion constraint generator. **Mechanism and Machine Theory**, Elsevier, v. 70, p. 394–406, 2013.

LAUS, LP; SIMAS, Henrique; MARTINS, Daniel. Efficiency of gear trains determined using graph and screw theories. **Mechanism and Machine Theory**, Elsevier, v. 52, p. 296–325, 2012.

LEE, Chung-Ching; HERVE, Jacques M. Generators of the product of two Schoenflies motion groups. **European Journal of Mechanics-A/Solids**, Elsevier, v. 29, n. 1, p. 97–108, 2010.

LI, Qinchuan; HUANG, Zhen; HERVE, Jacques M. Type synthesis of 3R2T 5-DOF parallel mechanisms using the Lie group of displacements. **IEEE transactions on robotics and automation**, IEEE, v. 20, n. 2, p. 173–180, 2004.

LI, Qinchuan; LINGMIN, XU; QIAOHONG, CHEN; WEI, YE. New family of RPR-equivalent parallel mechanisms: Design and application. **Chinese Journal of Mechanical Engineering**, Springer, v. 30, n. 2, p. 217–221, 2017.

LIU, TS; CHOU, CC. Type synthesis of vehicle planar suspension mechanism using graph theory, 1993.

LU, Yi; HU, Bo. Analyzing kinematics and solving active/constrained forces of a 3SPU+UPR parallel manipulator. **Mechanism and Machine Theory**, Elsevier, v. 42, n. 10, p. 1298–1313, 2007.

- LU, Yi; LEINONEN, Tatu. Type synthesis of unified planar–spatial mechanisms by systematic linkage and topology matrix-graph technique. **Mechanism and Machine Theory**, Elsevier, v. 40, n. 10, p. 1145–1163, 2005.
- LU, Yi; WANG, Ying; DING, Ling. Type synthesis of four-degree-of-freedom parallel mechanisms using valid arrays and topological graphs with digits. **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science**, SAGE Publications Sage UK: London, England, v. 228, n. 16, p. 3039–3053, 2014.
- MALETZ, Elias R; SIMAS, Henrique; BARRETO, Rodrigo L P, et al. Contribution to the design of hospital bed: systematic for surveying the design requirements and functional requirements for synthesis of mechanism. In: SPRINGER. INTERNATIONAL Conference on Applied Human Factors and Ergonomics. [S.l.: s.n.], 2017. P. 652–662.
- MARTINS, Daniel; SIMONI, Roberto; CARBONI, Andrea P. Fractionation in planar kinematic chains: Reconciling enumeration contradictions. **Mechanism and Machine Theory**, Elsevier, v. 45, n. 11, p. 1628–1641, 2010.
- MEJIA, L; SIMAS, Henrique; MARTINS, Daniel. Force capability in general 3 DoF planar mechanisms. **Mechanism and Machine Theory**, Elsevier, v. 91, p. 120–134, 2015.
- MORENO, Gonzalo; NICOLAZZI, Lauro Cesar; VIEIRA, Rodrigo De Souza; MARTINS, Daniel. Stability of long combination vehicles. **International journal of heavy vehicle systems**, Inderscience Publishers (IEL), v. 25, n. 1, p. 113–131, 2018.
- MRUTHYUNJAYA, TS. Kinematic structure of mechanisms revisited. **Mechanism and machine theory**, Elsevier, v. 38, n. 4, p. 279–320, 2003.
- MURAI, Estevan H. **Number synthesis methods for mechanism design: an alternative approach**. 2019. PhD thesis – Universidade Federal de Santa Catarina.
- MURAI, Estevan H.; SIMONI, Roberto; MARTINS, Daniel. Influence map: Determining the actuators? influence in kinematic chains for mechanism design. **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science**, SAGE Publications Sage UK: London, England, v. 233, n. 10, p. 3547–3556, 2019.
- MUROTA, Kazuo. **Matrices and matroids for systems analysis**. [S.l.]: Springer Science & Business Media, 2009. v. 20.
- NEEL, David L; NEUDAUER, Nancy Ann. Matroids you have known. **Mathematics magazine**, Taylor & Francis, v. 82, n. 1, p. 26–41, 2009.

NUNEZ, Neider Nadid Romero. **Síntese Estrutural e Otimização Dimensional de Mecanismos de Direção**. June 2014. MA thesis – Universidade Federal de Santa Catarina.

OXLEY, James G. **Matroid theory**. [S.l.]: Oxford University Press, USA, 2006. v. 3.

PASHKEVICH, Anatol; CHABLAT, Damien; WENGER, Philippe. Stiffness analysis of overconstrained parallel manipulators. *Mechanism and Machine Theory*, 44(5):966-982. v. 44, p. 966–982, May 2009.

PUCHETA, Martin A; BUTTI, Agostino; TAMELLINI, Valerio; CARDONA, Alberto; GHEZZI, Luca. Topological synthesis of planar metamorphic mechanisms for low-voltage circuit breakers. **Mechanics based design of structures and machines**, Taylor & Francis, v. 40, n. 4, p. 453–468, 2012.

PUCHETA, Martin A; CARDONA, Alberto. An automated method for type synthesis of planar linkages based on a constrained subgraph isomorphism detection. **Multibody System Dynamics**, Springer, v. 18, n. 2, p. 233–258, 2007.

RECSKI, Andras. **Matroid theory and its applications in electric network theory and in statics**. [S.l.]: Springer Science & Business Media, 2013. v. 6.

RESHETOV, L. **Self-Aligning Mechanism**. 2nd revised edition. Moscow: MIR, 1979. Translated from Russian by Leo M. Sachs.

RICHARD, Pierre-Luc; GOSSELIN, Clément M; KONG, Xianwen. Kinematic analysis and prototyping of a partially decoupled 4-DOF 3T1R parallel manipulator, 2006.

SANDOR, George N; ERDMAN, Arthur G. **Advanced Mechanism Design V. 2: Analysis and Synthesis**. [S.l.]: Prentice-Hall, 1984.

SIMONI, Roberto; CARBONI, Andrea P; MARTINS, Daniel. Enumeration of kinematic chains and mechanisms. **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science**, SAGE Publications Sage UK: London, England, v. 223, n. 4, p. 1017–1024, 2009.

SIMONI, Roberto; CARBONI, Andrea P; SIMAS, Henrique; MARTINS, Daniel. Enumeration of kinematic chains and mechanisms review. In: 13TH World Congress in Mechanism and Machine Science, Guanajuato, Mexico, June. [S.l.: s.n.], 2011. P. 19–25.

SIMONI, Roberto; MARTINS, Daniel. Criteria for structural synthesis and classification of mechanism. In: PROCEEDINGS of the 19th International Congress of Mechanical Engineering–COBEM. [S.l.: s.n.], 2007.

SUN, Tao; HUO, Xinming. Type synthesis of 1T2R parallel mechanisms with parasitic motions. **Mechanism and Machine Theory**, Elsevier, v. 128, p. 412–428, 2018.

SUN, Tao; YANG, Shuo-Fei; HUANG, Tian; DAI, Jian S. A finite and instantaneous screw based approach for topology design and kinematic analysis of 5-axis parallel kinematic machines. **Chinese Journal of Mechanical Engineering**, Springer, v. 31, n. 1, p. 44, 2018.

TISCHLER, CR; SAMUEL, AE; HUNT, Kenneth H. Kinematic chains for robot hands—I. Orderly number-synthesis. **Mechanism and Machine Theory**, Elsevier, v. 30, n. 8, p. 1193–1215, 1995a.

TISCHLER, CR; SAMUEL, AE; HUNT, Kenneth H. Kinematic chains for robot hands—II. Kinematic constraints, classification, connectivity, and actuation. **Mechanism and machine theory**, Elsevier, v. 30, n. 8, p. 1217–1239, 1995b.

TOSCANO, Gustavo S; SIMAS, Henrique; CASTELAN, Eugênio B; MARTINS, Daniel. A new kinetostatic model for humanoid robots using screw theory. **Robotica**, Cambridge University Press, v. 36, n. 4, p. 570–587, 2018.

TSAI, Lung-Wen. **Mechanism Design: enumeration of kinematic structures according to function**. New York: CRC, 2001.

UNO, Takeaki. A new approach for speeding up enumeration algorithms and its application for matroid bases. In: **COMPUTING and Combinatorics**. Berlin: Springer, 1999. P. 349–359. Available from:
<http://research.nii.ac.jp/~uno/papers/cocoon99web.pdf>.

WALDRON, Kenneth J. The constraint analysis of mechanisms. **Journal of Mechanisms**, v. 1, n. 2, p. 101–114, 1966.

WEI, Guowu; AMINZADEH, Vahid; EMMANOUIL, Evangelos; DAI, Jian S. Structure design, kinematics and grasp constraint of a metamorphic robotic hand for meat deboning operation. In: **AMERICAN SOCIETY OF MECHANICAL ENGINEERS. ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference**. [S.l.: s.n.], 2013.

WEI, Guowu; DAI, Jian S; WANG, Shuxin; LUO, Haifeng. Kinematic analysis and prototype of a metamorphic anthropomorphic hand with a reconfigurable palm. **International Journal of Humanoid Robotics**, World Scientific, v. 8, n. 03, p. 459–479, 2011.

WEI, Guowu; REN, Lei; DAI, Jian S. Prehension of an Anthropomorphic Metamorphic Robotic Hand Based on Opposition Space Model. In: **SPRINGER. INTERNATIONAL Conference on Intelligent Robotics and Applications**. [S.l.: s.n.], 2017. P. 71–83.

WEI, Guowu; STEPHAN, Franck; AMINZADEH, Vahid; WÜRDEMAN, Helge; WALKER, Rich; DAI, Jian S; GOGU, Grigore. Dexdeb—application of dextrous robotic hands for deboning operation. In: GEARING up and accelerating cross-fertilization between academic and industrial robotics research in Europe: [s.l.]: Springer, 2014. P. 217–235.

WHITNEY, Daniel E. **Mechanical assemblies: their design, manufacture, and role in product development**. [S.l.]: Oxford Series on advanced manufacturing, 2004. v. 1.

WHITNEY, Hassler. On the abstract properties of linear dependence. In: HASSLER Whitney Collected Papers. [S.l.]: Springer, 1992. P. 147–171.

XIE, Fugui; LIU, Xin-Jun; LI, Tiemin. Type synthesis and typical application of 1T2R-type parallel robotic mechanisms. **Mathematical Problems in Engineering**, Hindawi, v. 2013, 2013.

XIE, Fugui; LIU, Xin-Jun; YOU, Zheng; WANG, Jinsong. Type synthesis of 2T1R-type parallel kinematic mechanisms and the application in manufacturing. **Robotics and Computer-Integrated Manufacturing**, Elsevier, v. 30, n. 1, p. 1–10, 2014.

XU, Yundou; YAO, Jiantao; ZHAO, Yongsheng. Type synthesis of spatial mechanisms for forging manipulators. **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science**, SAGE Publications Sage UK: London, England, v. 226, n. 9, p. 2320–2330, 2012.

YAN, Hong-Sen. **Creative Design of Mechanical Devices**. [S.l.]: Springer, 1998.

YAN, Hong-Sen; CHIU, Yu-Ting. On the number synthesis of kinematic chains. **Mechanism and Machine Theory**, Elsevier, v. 89, p. 128–144, 2015.

YANG, Jialun; GAO, Feng; GE, Qiaode Jeffrey; ZHAO, Xianchao; GUO, Weizhong; JIN, Zhenlin. Type synthesis of parallel mechanisms having the first class G F sets and one-dimensional rotation. **Robotica**, Cambridge University Press, v. 29, n. 6, p. 895–902, 2011.

YANG, Jialun; GAO, Feng; ZHU, Kuanjun; LIU, Bin. Type synthesis of parallel mechanisms with the first class GF sets and two-dimensional rotations. **International Journal of Advanced Robotic Systems**, SAGE Publications Sage UK: London, England, v. 9, n. 3, p. 61, 2012.

YANG, Shuofei; SUN, Tao; HUANG, Tian. Type synthesis of parallel mechanisms having 3T1R motion with variable rotational axis. **Mechanism and Machine Theory**, Elsevier, v. 109, p. 220–230, 2017.

YANG, Shuofei; SUN, Tao; HUANG, Tian; LI, Qinchuan; GU, Dongbing. A finite screw approach to type synthesis of three-DOF translational parallel mechanisms.

Mechanism and Machine Theory, Elsevier, v. 104, p. 405–419, 2016.

YANG, Tingli; LIU, Anxin; SHEN, Huiping; HANG, Lubin. Topological structure synthesis of 3T1R parallel mechanism based on POC equations. In: SPRINGER. INTERNATIONAL Conference on Intelligent Robotics and Applications. [S.l.: s.n.], 2016. P. 147–161.

YE, Wei; LI, Qinchuan. Type synthesis of lower mobility parallel mechanisms: a review.

Chinese Journal of Mechanical Engineering, Springer, v. 32, n. 1, p. 38, 2019.

ZHANG, Chunsong; DAI, Jian S. Inverse Kinematics and Kineto-Statics of Metamorphic Palm of the KCL/TJU Metamorphic Hand. In: ADVANCES in Reconfigurable Mechanisms and Robots II. [S.l.]: Springer, 2016. P. 127–141.

ZHANG, Dan; GOSSELIN, Clement M. Kinetostatic modeling of N-DOF parallel mechanisms with a passive constraining leg and prismatic actuators. **J. Mech. Des.**, v. 123, n. 3, p. 375–381, 2000.

ZHANG, Xing; MU, Dejun; ZHANG, Yitong; YOU, Henghao; WANG, Hongrui. Type Synthesis of Multi-Loop Spatial Mechanisms With Three Translational Output Parameters Based on Virtual-Loop Theory and Assur Groups. **Robotica**, Cambridge University Press, v. 37, n. 6, p. 1104–1119, 2019.

APPENDIX A – INPUTS USED IN DAVIES' METHOD

This Appendix presents the information needed to reproduce the models created according to Davies' method throughout this work.

A.1 GRIPPER MECHANISMS

Table 26 shows the position for each joint of the gripper mechanisms.

Table 26 – Position for the joints for the Synthesis of the Gripper Mechanisms.

Joint	x	y
a	10	1
b	12	4
c	9	10
d	5	10
e	2	4
f	4	1
g	7	2
h	8	3
i	9	5
j	7	6
k	6	3
l	5	5

Source – From the author.

The simplified cutset matrix for the gripper mechanisms is shown in Equation 50.

$$[Q_{gripper}]_{8,12} = \begin{matrix} & a & b & c & d & e & f & g & h & i & j & k & l \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} & \end{matrix} \quad (50)$$

A.2 LEG-REST MECHANISM

Table 27 shows the position for each joint of the leg-rest mechanisms.

Table 27 – Position for the joints for the Synthesis of the Leg-rest Mechanisms.

Joint	x	y
a	3	1
b	4	2
c	6	0
d	2	7
e	5	5
f	7	8
g	0	3
h	1	4

Source – From the author.

The simplified cutset matrix for the leg-rest mechanisms is shown in Equation 51.

$$[Q_{leg}]_{6,8} = \begin{matrix} & a & b & c & d & e & f & g & h \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} & & & & & & & & \end{matrix} \quad (51)$$

A.3 BACKREST MECHANISM

Table 28 shows the position for each joint of the backrest mechanisms.

Table 28 – Position for the joints for the Synthesis of the Backrest Mechanisms.

Joint	x	y
a	11,7	10,1
b	2,9	13,3
c	3,6	21,2
d	27,1	41,8
e	19,7	44,2
f	-11,3	32,8
g	-31,3	51,7
h	71,9	92,1
i	-31,7	51,2
j	-71,4	51,1

Source – From the author.

The simplified cutset matrix for the backrest mechanisms is shown in Equation 52.

$$[Q_{back}]_{7,10} = \begin{matrix} & a & b & c & d & e & f & g & h & i & j \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix} \quad (52)$$

APPENDIX B – INTRODUCTION TO MATROID THEORY

In this appendix, a review about matroid theory is presented. Let us begin by reviewing what is a matroid, what are the family of bases, what is the rank of a matroid and what is contraction and deletion of matroid elements. Take, for instance, the matrix $[M]_{3,6}$ from Equation 53:

$$[M]_{3,6} = \begin{array}{c} \begin{array}{ccc|ccc} & a & b & c & d & e & f \\ \hline 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{array} \end{array} \quad (53)$$

Matrix M has six columns, indexed by $E = \text{Col}(M) = a, b, c, d, e, f$. From this matrix we create a linear matroid, Equation 54.

$$\mathcal{M} = (E, \mathcal{B}) \quad (54)$$

The linear matroid \mathcal{M} is created from matrix M on the groundset E . \mathcal{B} is the family of bases from the matroid \mathcal{M} , presented in Equation 55.

$$\mathcal{B} = \{\{a, b, c\}, \{a, b, d\}, \{a, c, e\}, \{a, d, e\}, \\ \{b, c, e\}, \{b, c, f\}, \{b, d, e\}, \{b, d, f\}, \{c, e, f\}, \{d, e, f\}\} \quad (55)$$

The bases from matroid \mathcal{M} have three elements each, *i.e.*, the rank of the bases is three. This means that for matroid \mathcal{M} there are a maximum of 3 linearly independent elements.

Another important characteristic of matroids is the dual bases. Every matroid has a dual matroid $\mathcal{M}^* = (E, \mathcal{B}^*)$. Each basis from the family \mathcal{B}^* is a complement of another from the family \mathcal{B} . The basis $\{a, b, c\}$ from the matroid \mathcal{M} has a dual basis $\{d, e, f\} \in \mathcal{B}^*$. The bases \mathcal{B}^* are shown in Equation 56.

$$\mathcal{B}^* = \{\{d, e, f\}, \{c, e, f\}, \{b, d, f\}, \{b, c, f\}, \\ \{a, d, f\}, \{a, d, e\}, \{a, c, f\}, \{a, c, e\}, \{a, b, d\}, \{a, b, c\}\} \quad (56)$$

Now we can introduce the operations called contraction and deletion, which follows the formulation by Recski.

Take the matroid $\mathcal{M} = (E, \mathcal{B})$ and consider a $X \subseteq E$. We can define a new family of bases \mathcal{B}' so that an $Y \subseteq E - X$ exists so that $Y \in \mathcal{B}'$ if and only if $Y \in \mathcal{B}$. This new family \mathcal{B}' is the bases of a new matroid \mathcal{M}' on $E - X$. The matroid \mathcal{M}' is denoted by $\mathcal{M} \setminus X$ and is called the deletion of X from \mathcal{M} or the restriction of \mathcal{M} to $E - X$.

Let us use matroid \mathcal{M} from matrix $[M]_{3,6}$ for an example. We wish to delete the column d from the matroid. The new matroid \mathcal{M}' is defined as $\mathcal{M} \setminus d$, and the family of bases is:

$$\mathcal{B}' = \{\{a, b, c\}, \{a, c, e\}, \{b, c, e\}, \{b, c, f\}, \{b, e, f\}\} \quad (57)$$

The deletion of a single element of matroid \mathcal{M} also means that the cardinality, or rank, of the dual bases of matroid \mathcal{M}'^* decreases by one when compared to the cardinality of the original dual bases. Thus, the bases from the matroid \mathcal{M}'^* are:

$$\mathcal{B}'^* = \{\{e, f\}, \{b, f\}, \{a, f\}, \{a, e\}, \{a, b\}\} \quad (58)$$

Recski defines the operation called contraction by means of the rank function of matroids. For a start, take r as the rank function of the matroid $\mathcal{M}=(E, \mathcal{B})$ and consider a subset $X \subseteq E$. The rank function R of every subset Y of the matroid \mathcal{M} is defined by Equation 59.

$$R(Y) = r(X \cup Y) - r(X) \quad (59)$$

Equation 59 is valid for any subset Y of the set $E - X$. The new matroid with rank function $R(Y)$ on the set $E - X$ is denoted by \mathcal{M}/X and is known as the contraction of \mathcal{M} to $E - X$.

Let us turn again to matrix M from Equation 53. Element d now will be contracted from the matroid $\mathcal{M} = (E, \mathcal{B})$. The new matroid $\mathcal{M}'' = (E, \mathcal{B})$ is defined as \mathcal{M}/d , and the family of bases is:

$$\mathcal{B}'' = \{\{a, b\}, \{a, e\}, \{b, e\}, \{b, f\}, \{e, f\}\} \quad (60)$$

The family of bases of the dual matroid \mathcal{M}''^* is:

$$\mathcal{B}''^* = \{\{c, e, f\}, \{b, c, f\}, \{a, c, f\}, \{a, c, e\}, \{a, b, c\}\} \quad (61)$$

By contracting a single element from matroid \mathcal{M} , the rank of the bases will decrease by one, while the rank of the dual bases will remain unaltered.

B.1 OTHER MATROID FORMULATIONS

The matroid formulation was already shown in Chapter 5 is based on subsets and their linear independence. In this case, a matroid formed by a finite set E and a collection of subsets \mathcal{I} of E and has the following properties (RECSKI, 2013):

- (I1) $\emptyset \in \mathcal{I}$ - the empty set belongs to \mathcal{I}

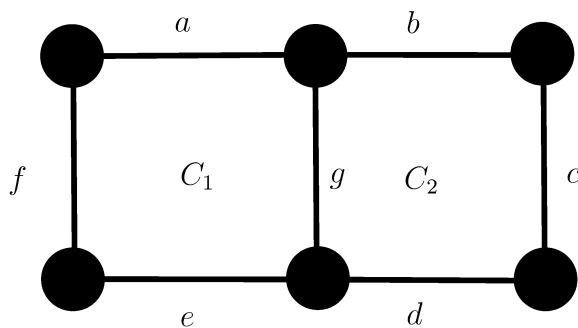
- (I2) $J \in \mathcal{I}$, and $I \subseteq J \Rightarrow I \in \mathcal{I}$ - a subset I , contained in the subset J , is contained in the collection of subsets \mathcal{I} if the subset J is contained in \mathcal{I}
- (I3) $I, J \in \mathcal{I}, |I| < |J| \Rightarrow (I \cup \{v\}) \in \mathcal{I}$ for some $v \in J \setminus I$ - if a subset I is smaller than the subset J , we can find an element from J that does not belong to I so that the union of this element to I is contained in \mathcal{I}

Besides the formulation from subsets, other formulations are available, such as the formulation using circuits. While the matroid formulation using sets is easily seen using matrices, the formulation using circuits are easier reviewed using graphs. A circuit is defined as minimal dependent set in an arbitrary matroid (OXLEY, 2006) at a graph, a circuit is a bi-connected closed loop. A set of circuits of a matroid is denoted by \mathcal{C} while a single circuit is denoted by C_i . Matroids created using circuits have the following properties (OXLEY, 2006):

- (C1) $\emptyset \notin \mathcal{C}$.
- (C2) If C_1 and C_2 are members of \mathcal{C} and $C_1 \subseteq C_2$ then $C_1 = C_2$.
- (C3) If C_1 and C_2 are distinct members of \mathcal{C} and $e \in C_1 \cap C_2$, then there is a circuit C_3 such that $C_3 \subseteq (C_1 \cup C_2) - e$.

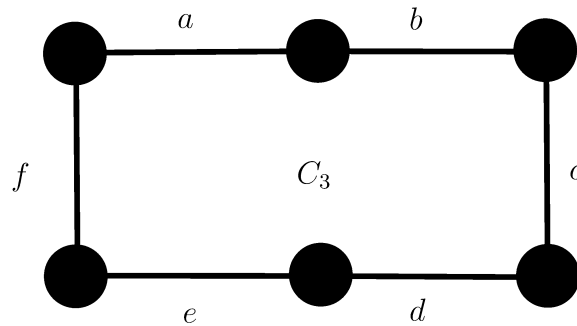
For the first property, as a circuit must be a minimal dependent set, the empty set is not included in \mathcal{C} , as it belongs to \mathcal{I} . As for the second property, considering a circuit as a minimal dependent set, it is not possible to have a minimal dependent set inside another minimal dependent set, thus both circuits must be the same. Property 3 is better shown in Figure 81:

Figure 81 – Graph with circuits C_1 and C_2 .



Source – From the author.

By removing the edge g , circuit C_3 is shown:

Figure 82 – Graph with circuit C_3 .

Source – From the author.

From Figure 82 it is clear that circuit C_3 is the union of circuits C_1 and C_2 minus the edge g , exactly as property 3 stated.

Recall matrix $[M]$ from the previous section:

$$[M]_{3,6} = \left[\begin{array}{ccc|ccc} a & b & c & d & e & f \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{array} \right] \quad (62)$$

The circuits of the matroid created using this matrix are:

$$\mathcal{C} = \{\{a, f\}, \{c, d\}, \{a, b, e\}, \{b, e, f\}\} \quad (63)$$

A set of bases can also be used as a matroid formulation. Consider a set \mathcal{B} of bases of a set E , set \mathcal{B} is a collection of bases of a matroid if the set fulfils the following properties (OXLEY, 2006):

- (B1) \mathcal{B} is non-empty.
- (B2) If B_1 and B_2 are members of \mathcal{B} and $x \in B_1 - B_2$, then there is an element y of $B_2 - B_1$ such that $(B_1 - x) \cup y \in \mathcal{B}$.
- (B3) If B_1 and B_2 are included in \mathcal{B} and $B_1 \subseteq B_2$, then $B_1 = B_2$.

Throughout the thesis, the rank of matroids was frequently used. The rank r of a set X is said to be size of a basis B from the matroid over the set X . The rank r of a matroid has the following properties:

- (R1) If $X \subseteq E$, then $0 \leq r(X) \leq |X|$.
- (R2) If $X \subseteq Y \subseteq E$, then $r(X) \leq r(Y)$.

(R3) If X and Y are subsets of E , then $r(X \cup Y) + r(X \cap Y) \leq r(X) + r(Y)$.

Closure operators are also valuable when working with matroids. Let cl be the function from 2^E into 2^E defined for every $X \subseteq E$ by:

$$cl(X) = \{x \in E : r(X \cup x) = r(X)\}. \quad (64)$$

From Equation 64, the closure operator from a matroid can be defined as the elements that can be added to a set without increasing the rank of the set. The closure operator of a matroid presents the following properties (OXLEY, 2006):

(CL1) If $X \subseteq E$, then $X \subseteq cl(X)$.

(CL2) If $X \subseteq Y \subseteq E$, then $cl(X) \subseteq cl(Y)$.

(CL3) If $X \subseteq Y \subseteq E$, then $cl(cl(X)) = cl(X)$.

(CL4) If $X \subseteq E$, $x \in E$, and $y \in cl(X \cup x) - cl(X)$, then $x \in cl(X \cup y)$.

The mathematical proof of all the properties presented can be seen in Oxley (2006).

APPENDIX C – CONTRACTION AND DELETION OF MATROIDS

Present here the step by step of contraction and deletion of matroids. Follow Oxley's method to present contraction and deletion step by step, independently as well as contraction by means of deletion.

C.1 CONTRACTION AND DELETION OF LINEAR MATROIDS

In this section we will demonstrate how to apply contraction and deletion to linear matroids using their matrix representation. The matrix representation of the matroid \mathcal{M} , demonstrated by Oxley, is shown in Equation 65, while the dual matroid is represented by the matrix in Equation 66.

$$[A] = \left[\begin{array}{cccc|cccc} e_1 & e_2 & \dots & e_r & e_{r+1} & e_{r+1} & \dots & e_n \\ & & & & & & & \\ & & & & & & & \\ & I_r & & & & D & & \\ & & & & & & & \end{array} \right] \quad (65)$$

$$[A^*] = \left[\begin{array}{cccc|cccc} e_1 & e_2 & \dots & e_r & e_{r+1} & e_{r+1} & \dots & e_n \\ & & & & & & & \\ & & & & & & & \\ & -D^T & & & & I_{n-r} & & \\ & & & & & & & \end{array} \right] \quad (66)$$

In Equations 65 and 66, I represents an identity matrix while D is a $r \times (n - r)$ matrix over the groundset of $[A]$. Using these matrices, we will demonstrate how to create a matrix representation of matroids by means of contraction and deletion. Let us turn to matrix $[A_1]$, which is used by Oxley in some examples:

$$[A_1]_{4,8} = \left[\begin{array}{cccc|cccc} a & b & c & d & e & f & g & h \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 1 & 1 & 0 \end{array} \right] \quad (67)$$

The matroid created from matrix $[A_1]$ has rank 4 and 64 bases. The dual matroid also has rank 4 and 64 bases.

We will start by demonstrating the deletion. From matrix $[A_1]$ we choose to delete element h (column 8). To attain the matroid $\mathcal{M} \setminus (h)$ all we are required is to eliminate

the chosen column from matrix $[A_1]$. Hence, we obtain matrix $[A_1 \setminus h]$, Equation 68.

$$[A_1 \setminus h]_{4,7} = \begin{array}{cccc|ccc} & a & b & c & d & e & f & g \\ \hline & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ & 0 & 0 & 0 & 1 & -1 & 1 & 1 \end{array} \quad (68)$$

The matroid represented by matrix $[A_1 \setminus h]$ has rank 4 and 31 bases. The dual matroid instead has rank 3, which previously also had rank 4. Hence, we can conclude that the deletion of a single element decreases the rank of the dual matroid, making no changes to the original matroid, only reducing the number of bases.

After presenting deletion, now we can move to contraction. Oxley (2006) states that while deletion consists into removing a column of the matrix used to create the matrix, contraction can be considered as removing a row from the matrix. Unlike deletion, contraction requires some work with the matrix prior to removing the desired elements. Let us return to the matrix A from the example. The element h must be contracted from the matroid, therefore some row operations are required such that only one element of the column is 1 while the remaining elements are 0. In this example, the element of the third row was chosen as a pivot, so the remaining elements must be transformed into zero. Row 3 is added to row 1 and row 3 is subtracted from row 2, achieving the following matrix:

$$[A_1]_{4,8} = \begin{array}{cccc|cccc} & a & b & c & d & e & f & g & h \\ \hline & 1 & 0 & 1 & 0 & 1 & 2 & 1 & 0 \\ & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 \\ & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ & 0 & 0 & 0 & 1 & -1 & 1 & 1 & 0 \end{array} \quad (69)$$

Now row 3 and column h are removed from the matrix, resulting in a new matrix $[A_1/h]$.

$$[A_1/h]_{3,7} = \begin{array}{cccc|ccc} & a & b & c & d & e & f & g \\ \hline & 1 & 0 & 1 & 0 & 1 & 2 & 1 \\ & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ & 0 & 0 & 0 & 1 & -1 & 1 & 1 \end{array} \quad (70)$$

The matroid represented by matrix $[A_1/h]$ has 33 bases and rank 3, while it's dual matroid has the same number of bases with rank 4. We can conclude that contracting an element from a matroid decreases the rank of that matroid at the same time it

reduces the number of bases. As for the dual matroid, contraction only reduces the number of bases.

As mentioned in the previous section, deletion and contraction are dual to each other. To demonstrate this operation, we need to turn once more to the relations between contraction and deletion, shown again in Equations 71 and 72. Hence, we can contract an element from the matroid from matrix $[A_1]$, we can delete the same element in the dual matroid from that matrix.

$$\mathcal{M}/T = (\mathcal{M}^*\setminus T)^* \quad (71)$$

Recski (2013) also presents another relation between contraction and deletion, shown in Equation 72.

$$(\mathcal{M}\setminus T)^* = \mathcal{M}^*/T \quad (72)$$

From matrix $[A_1]$ we can create the representation from the dual matroid of this matrix by using equations 65 and 66. This new matrix $[A_1^*]$ is shown in Equation 73.

$$[A_1^*]_{4,8} = \begin{array}{c} a \quad b \quad c \quad d \quad e \quad f \quad g \quad h \\ \left[\begin{array}{cccc|cccc} 0 & -1 & -1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & 0 & 1 & 0 & 0 \\ -1 & -1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \end{array} \quad (73)$$

We want to contract element h from the matroid of matrix $[A_1]$, therefore we need to delete the same element from matrix $[A_1^*]$. After this operation we obtain matrix $[A_1^*\setminus h]$:

$$[A_1^*\setminus h]_{4,7} = \begin{array}{c} a \quad b \quad c \quad d \quad e \quad f \quad g \\ \left[\begin{array}{cccc|ccc} 0 & -1 & -1 & 1 & 1 & 0 & 0 \\ -1 & 0 & -1 & -1 & 0 & 1 & 0 \\ -1 & -1 & 0 & -1 & 0 & 0 & 1 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 \end{array} \right] \end{array} \quad (74)$$

After attaining matrix $[A_1^*\setminus h]$, we use Equations 65 and 66 to find the dual of the matroid represented by $[A_1^*\setminus h]$, which in turn is matrix $[A_1/h]$:

$$[A_1/h]_{3,7} = \begin{array}{c} a \quad b \quad c \quad d \quad e \quad f \quad g \\ \left[\begin{array}{cccc|ccc} 1 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 1 & 1 \end{array} \right] \end{array} \quad (75)$$

The contraction of an element from matrix $[A]$ shown in Equation 70 and the contraction shown using deletion of an element of the dual in Equation 75 are identical. Table 29 presents a summary of the impact contraction and deletion has on the rank and number of bases of the matroids shown.

Table 29 – Summary of the contraction and deletion of linear matroids.

	\mathcal{M} rank	\mathcal{M}^* rank	Number of bases
Matrix $[A]$	4	4	64
Matrix $[A]$ with column h deleted	4	3	31
Matrix $[A]$ with column h contracted	3	4	33

Source – From the author.

We have some final remarks about the methods and operations presented in this section. All of the operations were done by matrices and later double checked using Sagemath, verifying that the original matroids were indeed equal to the matroids we obtained through matrices operations. When using matrices for deletion and contraction, one must keep in mind that the matrices may suffer from errors due to the round-off of some elements. This rounding error may create new linear dependencies between columns be created and hence create a new matroid.

C.2 CONTRACTION AND DELETION OF GRAPHS AND GRAPHICAL MATROIDS

In this section we will show how to use deletion and contraction applied to graphs and graphical matroids. To start, we will define the graph G that will be used in the examples. This graph is shown in Figure 83.

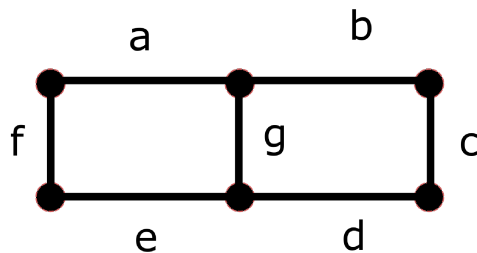


Figure 83 – Graph for the examples.

The graph from Figure 83 has 7 edges and 6 vertices. If we create a matroid $\mathcal{M}(G)$ from this graph, the bases of this matroid will consist of every possible combination of spanning trees for graph G . The matroid $\mathcal{M}(G)$ has rank 5 and 15 bases. The dual matroid $\mathcal{M}^*(G)$ has rank 2 and 15 bases.

The deletion in graphs consists of eliminating an edge of the graph. Figure 84 presents graph G with edge c deleted.

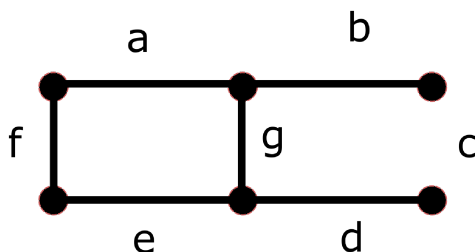


Figure 84 – Graph with edge c deleted.

A new matroid $\mathcal{M} \setminus (c)$ from the graph of Figure 84 has now rank 5 and 5 bases. The dual matroid $(\mathcal{M} \setminus (c))^*$ also has 5 bases but instead has rank 1. When comparing the matroids $\mathcal{M}(G)$ and $\mathcal{M} \setminus (c)$, we can see the rank of the matroid did not change, while the rank of the dual matroid decreased by one. Another change is in the number of bases, which decreased from 15 to 5.

Returning now to graph G we will demonstrate contraction. This operation consists in contracting an edge from a graph. Figure 85 presents the graph G with edge c contracted.

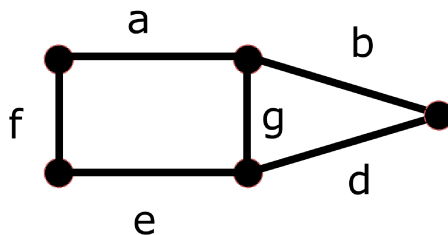


Figure 85 – Graph with edge c contracted.

The new matroid $\mathcal{M} / (c)$ of Figure 85 has rank 4 and 11 bases. The dual matroid $(\mathcal{M} / (c))^*$ has rank 2 and 11 bases. The difference between matroids $\mathcal{M}(G)$ and $\mathcal{M} / (c)$ is the decrease of the rank of the matroid and of the number of bases. The rank of the dual matroid did not change after the contraction, only reducing the number of bases.

Table 30 presents a summary of the results presented in this section.

Table 30 – Summary of the contraction and deletion of graphical matroids.

	\mathcal{M} rank	\mathcal{M}^* rank	Number of bases
Graph G	5	2	15
Graph G with edge c deleted	5	1	5
Graph G with edge c contracted	4	2	11

Source – From the author.

APPENDIX D – SETS OF MECHANISMS ENUMERATED

In this appendix the sets of mechanisms enumerated throughout the thesis are presented.

D.1 RECONFIGURABLE PALM MECHANISMS ENUMERATED BY METHOD I

The self-aligning mechanisms based on the KCL/TJU reconfigurable palm enumerated using method I are presented in this section. The method I eliminates redundant constraints of the joints of a seed mechanism. Joints a and e were defined to remain as revolute joints so no constraints are eliminated from these joints. Joint b was defined as a cylindrical joint, so only the constraint U was eliminated.

Table 31 – Eliminated constraints of the mechanisms enumerated by method I.

	Joint <i>b</i>	Joint <i>c</i>	Joint <i>d</i>
Mechanism 1	<i>U</i>	<i>U, V</i>	none
Mechanism 2	<i>U</i>	<i>U, W</i>	none
Mechanism 3	<i>U</i>	<i>R, U</i>	none
Mechanism 4	<i>U</i>	<i>S, U</i>	none
Mechanism 5	<i>U</i>	<i>U</i>	<i>U</i>
Mechanism 6	<i>U</i>	<i>U</i>	<i>V</i>
Mechanism 7	<i>U</i>	<i>U</i>	<i>W</i>
Mechanism 8	<i>U</i>	<i>U</i>	<i>S</i>
Mechanism 9	<i>U</i>	<i>U</i>	<i>T</i>
Mechanism 10	<i>U</i>	<i>V, W</i>	none
Mechanism 11	<i>U</i>	<i>R, V</i>	none
Mechanism 12	<i>U</i>	<i>S, V</i>	none
Mechanism 13	<i>U</i>	<i>V</i>	<i>U</i>
Mechanism 14	<i>U</i>	<i>V</i>	<i>V</i>
Mechanism 15	<i>U</i>	<i>V</i>	<i>W</i>
Mechanism 16	<i>U</i>	<i>V</i>	<i>R</i>
Mechanism 17	<i>U</i>	<i>V</i>	<i>T</i>
Mechanism 18	<i>U</i>	<i>R, W</i>	none
Mechanism 19	<i>U</i>	<i>S, W</i>	none
Mechanism 20	<i>U</i>	<i>W</i>	<i>U</i>
Mechanism 21	<i>U</i>	<i>W</i>	<i>V</i>
Mechanism 22	<i>U</i>	<i>W</i>	<i>W</i>
Mechanism 23	<i>U</i>	<i>W</i>	<i>R</i>
Mechanism 24	<i>U</i>	<i>W</i>	<i>T</i>
Mechanism 25	<i>U</i>	<i>R, S</i>	none
Mechanism 26	<i>U</i>	<i>R</i>	<i>U</i>
Mechanism 27	<i>U</i>	<i>R</i>	<i>V</i>
Mechanism 28	<i>U</i>	<i>R</i>	<i>W</i>
Mechanism 29	<i>U</i>	<i>R</i>	<i>R</i>
Mechanism 30	<i>U</i>	<i>R</i>	<i>T</i>
Mechanism 31	<i>U</i>	<i>S</i>	<i>U</i>
Mechanism 32	<i>U</i>	<i>S</i>	<i>V</i>
Mechanism 33	<i>U</i>	<i>S</i>	<i>W</i>
Mechanism 34	<i>U</i>	<i>S</i>	<i>R</i>
Mechanism 35	<i>U</i>	<i>S</i>	<i>T</i>
Mechanism 36	<i>U</i>	none	<i>U, V</i>
Mechanism 37	<i>U</i>	none	<i>U, W</i>
Mechanism 38	<i>U</i>	none	<i>R, U</i>
Mechanism 39	<i>U</i>	none	<i>T, U</i>
Mechanism 40	<i>U</i>	none	<i>V, W</i>
Mechanism 41	<i>U</i>	none	<i>R, V</i>
Mechanism 42	<i>U</i>	none	<i>T, V</i>
Mechanism 43	<i>U</i>	none	<i>R, W</i>
Mechanism 44	<i>U</i>	none	<i>T, W</i>
Mechanism 45	<i>U</i>	none	<i>R, T</i>

Source – From the author.

D.2 RECONFIGURABLE PALM MECHANISMS ENUMERATED BY METHOD II

The self-aligning mechanisms based on the KCL/TJU reconfigurable palm enumerated using method II are presented in this section. The method II creates Reshetov virtual joints and uses these joints to create new self-aligning mechanisms, thus only the Reshetov virtual joints f , g and h are shown, as the original five joints suffered no impact on the selection process. Table 32 presents the constraint eliminated from the Reshetov virtual joints for each of the 33 enumerated mechanisms. Reshetov virtual joints that do not receive freedoms will be integrated back into the link. Joint f was defined as a prismatic joint in the z -axis, therefore only W constraints are eliminated from these joints.

Table 32 – Eliminated constraints of the mechanisms enumerated by method II.

	Joint f	Joint g	Joint h
Mechanism 1	W	U, V	none
Mechanism 2	W	R, U	none
Mechanism 3	W	T, U	none
Mechanism 4	W	U	V
Mechanism 5	W	U	R
Mechanism 6	W	U	T
Mechanism 7	W	S, V	none
Mechanism 8	W	V, V	none
Mechanism 9	W	V	U
Mechanism 10	W	V	S
Mechanism 11	W	V	T
Mechanism 12	W	R, S	none
Mechanism 13	W	R, T	none
Mechanism 14	W	R	U
Mechanism 15	W	R	S
Mechanism 16	W	R	T
Mechanism 17	W	S, T	none
Mechanism 18	W	S	V
Mechanism 19	W	S	R
Mechanism 20	W	S	T
Mechanism 21	W	T	U
Mechanism 22	W	T	V
Mechanism 23	W	T	R
Mechanism 24	W	T	S
Mechanism 25	W	T	T
Mechanism 26	W	none	U, V
Mechanism 27	W	none	R, U
Mechanism 28	W	none	T, U
Mechanism 29	W	none	S, V
Mechanism 30	W	none	T, V
Mechanism 31	W	none	R, S
Mechanism 32	W	none	R, T
Mechanism 33	W	none	S, T

Source – From the author.

D.3 MECHANISMS ENUMERATED BY THE TYPE SYNTHESIS METHOD

Using the method proposed in Chapter 8, a seed mechanism with four links and four Reshetov virtual joints was used to enumerate new example mechanisms. Table 33 presents the enumerated mechanisms.

Table 33 – Eliminated constraints of the example mechanism enumerated by the type synthesis method.

	Joint <i>a</i>	Joint <i>b</i>	Joint <i>c</i>	Joint <i>d</i>
Mechanism 1	<i>T</i>	<i>T</i>	<i>U, V</i>	none
Mechanism 2	<i>T</i>	<i>T</i>	<i>T</i>	<i>U</i>
Mechanism 3	<i>T</i>	<i>T</i>	<i>U</i>	<i>T</i>
Mechanism 4	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>
Mechanism 5	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>
Mechanism 6	<i>T</i>	<i>T</i>	<i>V</i>	<i>U</i>
Mechanism 7	<i>T</i>	<i>T</i>	<i>U, T</i>	none
Mechanism 8	<i>T</i>	<i>T</i>	none	<i>V, T</i>
Mechanism 9	<i>T</i>	<i>T</i>	<i>V, T</i>	none
Mechanism 10	<i>T</i>	<i>T</i>	none	<i>U, V</i>
Mechanism 11	<i>T</i>	<i>T</i>	<i>U</i>	<i>V</i>
Mechanism 12	<i>T</i>	<i>T</i>	none	<i>U, T</i>
Mechanism 13	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>

Source – From the author.

D.4 LEG-REST MECHANISMS ENUMERATED BY THE TYPE SYNTHESIS METHOD

Using the method proposed in Chapter 8, a seed mechanism with seven links and eight Reshetov virtual joints was used to enumerate new leg-rest mechanisms. Table 34 presents the enumerated mechanisms.

Table 34 – Eliminated constraints of the leg-rest mechanisms enumerated by the type synthesis method.

	Joint <i>a</i>	Joint <i>b</i>	Joint <i>c</i>	Joint <i>d</i>	Joint <i>f</i>	Joint <i>h</i>
Mechanism 1	<i>T</i>	<i>T, U, V</i>	none	<i>T</i>	none	<i>T</i>
Mechanism 2	<i>T</i>	<i>T, U</i>	<i>V</i>	<i>T</i>	none	<i>T</i>
Mechanism 3	<i>T</i>	<i>T, U</i>	<i>T</i>	<i>T</i>	none	<i>T</i>
Mechanism 4	<i>T</i>	<i>T, U</i>	none	<i>T</i>	<i>V</i>	<i>T</i>
Mechanism 5	<i>T</i>	<i>T, U</i>	none	<i>T</i>	<i>T</i>	<i>T</i>
Mechanism 6	<i>T</i>	<i>T, V</i>	<i>U</i>	<i>T</i>	none	<i>T</i>
Mechanism 7	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	none	<i>T</i>
Mechanism 8	<i>T</i>	<i>T, V</i>	none	<i>T</i>	<i>U</i>	<i>T</i>
Mechanism 9	<i>T</i>	<i>T, V</i>	none	<i>T</i>	<i>T</i>	<i>T</i>
Mechanism 10	<i>T</i>	<i>T</i>	<i>U, V</i>	<i>T</i>	none	<i>T</i>
Mechanism 11	<i>T</i>	<i>T</i>	<i>T, U</i>	<i>T</i>	none	<i>T</i>
Mechanism 12	<i>T</i>	<i>T</i>	<i>U</i>	<i>T</i>	<i>V</i>	<i>T</i>
Mechanism 13	<i>T</i>	<i>T</i>	<i>U</i>	<i>T</i>	<i>T</i>	<i>T</i>
Mechanism 14	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	none	<i>T</i>
Mechanism 15	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>U</i>	<i>T</i>
Mechanism 16	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>T</i>	<i>T</i>
Mechanism 17	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>U</i>	<i>T</i>
Mechanism 18	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>
Mechanism 19	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>
Mechanism 20	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>U, V</i>	<i>T</i>
Mechanism 21	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>T, U</i>	<i>T</i>
Mechanism 22	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>T, V</i>	<i>T</i>

Source – From the author.

D.5 BACKREST MECHANISMS ENUMERATED BY THE TYPE SYNTHESIS METHOD

Using the method proposed in Chapter 8, a seed mechanism with eight links and ten Reshetov virtual joints was used to enumerate new leg-rest mechanisms. Table 35 presents the enumerated mechanisms.

Table 35 – Eliminated constraints of the backrest mechanisms enumerated by the type synthesis method.

	Joint <i>a</i>	Joint <i>b</i>	Joint <i>c</i>	Joint <i>d</i>	Joint <i>f</i>	Joint <i>h</i>	Joint <i>h</i>	Joint <i>i</i>	Joint <i>j</i>
Mech 1	<i>T</i>	<i>T</i>	<i>U</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>	none
Mech 2	<i>T</i>	<i>T</i>	<i>U</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	none	<i>V</i>
Mech 3	<i>T</i>	<i>T</i>	<i>U</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>V</i>	none
Mech 4	<i>T</i>	<i>T</i>	<i>U</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	none	<i>V</i>
Mech 5	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	none	none
Mech 6	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	none	none
Mech 7	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>	none
Mech 8	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	none
Mech 9	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	none	<i>V</i>
Mech 10	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	none	<i>T</i>
Mech 11	<i>T</i>	<i>T</i>	<i>V</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>U</i>	none
Mech 12	<i>T</i>	<i>T</i>	<i>V</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	none
Mech 13	<i>T</i>	<i>T</i>	<i>V</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	none	<i>U</i>
Mech 14	<i>T</i>	<i>T</i>	<i>V</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	none	<i>T</i>
Mech 15	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>U</i>	none
Mech 16	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	none
Mech 17	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	none	<i>U</i>
Mech 18	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	none	<i>T</i>
Mech 19	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>U</i>	<i>V</i>
Mech 20	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>
Mech 21	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>	<i>U</i>
Mech 22	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>
Mech 23	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>
Mech 24	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>
Mech 25	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	none	<i>T, V</i>
Mech 26	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>	none
Mech 27	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	none
Mech 28	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	none	<i>V</i>
Mech 29	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	none	<i>T</i>
Mech 30	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>V</i>	none

continued in the next page...

	Joint <i>a</i>	Joint <i>b</i>	Joint <i>c</i>	Joint <i>d</i>	Joint <i>f</i>	Joint <i>h</i>	Joint <i>h</i>	Joint <i>i</i>	Joint <i>j</i>
Mech 31	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	none
Mech 32	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	none	<i>V</i>
Mech 33	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	none	<i>T</i>
Mech 34	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>U</i>	<i>V</i>
Mech 35	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, V</i>	none
Mech 36	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>
Mech 37	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>
Mech 38	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>
Mech 39	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	none	<i>T, V</i>
Mech 40	<i>T</i>	<i>T</i>	none	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>U</i>	<i>V</i>
Mech 41	<i>T</i>	<i>T</i>	none	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, V</i>	none
Mech 42	<i>T</i>	<i>T</i>	none	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>
Mech 43	<i>T</i>	<i>T</i>	none	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>V</i>
Mech 44	<i>T</i>	<i>T</i>	none	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>
Mech 45	<i>T</i>	<i>T</i>	none	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	none	<i>T, V</i>
Mech 46	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>U</i>	<i>V</i>
Mech 47	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T, V</i>	none
Mech 48	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>V</i>	<i>T</i>
Mech 49	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>U</i>
Mech 50	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>
Mech 51	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>T, V</i>	<i>T</i>	<i>T</i>	none	<i>T, V</i>
Mech 52	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, U</i>	<i>V</i>
Mech 53	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>U</i>	<i>T, V</i>
Mech 54	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, V</i>	<i>T</i>
Mech 55	<i>T</i>	<i>T</i>	none	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T</i>	<i>T, V</i>

Source – From the author.