UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO TECNOLÓGICO

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

VINÍCIUS NOAL ARTMANN

A NEW METHOD TO SELECT SELF-ALIGNING MECHANISMS ENUMERATED BY MATROID: CASE STUDIES IN HOSPITAL BEDS

FLORIANÓPOLIS

2019

Vinícius Noal Artmann

A NEW METHOD TO SELECT SELF-ALIGNING MECHANISMS ENUMERATED BY MATROID: CASE STUDIES IN HOSPITAL BEDS

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do Grau de Mestre em Engenharia Mecânica.

Orientador: Prof. Roberto Simoni, Dr. Eng.

Coorientador: Prof. Daniel Martins, Dr. Eng.

Florianópolis

2019

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

Artmann, Vinicius Noal
A NEW METHOD TO SELECT SELF-ALIGNING MECHANISMS ENUMERATED BY MATROID: CASE STUDIES IN HOSPITAL BEDS / Vinicius Noal Artmann ; orientador, Roberto Simoni, coorientador, Daniel Martins, 2019.
156 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Mecânica, Florianópolis, 2019.

Inclui referências.

1. Engenharia Mecânica. 2. Autoalinhamento. 3. Teoria de Matroide. 4. Mecanismos. 5. Método de Davies. I. Simoni, Roberto. II. Martins, Daniel. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Mecânica. IV. Título.

Vinícius Noal Artmann

A NEW METHOD TO SELECT SELF-ALIGNING MECHANISMS ENUMERATED BY MATROID: CASE STUDIES IN HOSPITAL BEDS

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

Prof. Roberto Simoni, Dr. Eng.
Presidente

Prof. Edson Roberto De Pieri, Dr. Eng.
Universidade Federal de Santa Catarina

Prof. Celso Melchiades Doria, Dr.
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 Mestre em Engenharia Mecânica.

Assinado de forma digital por Jonny Carlos da Silva:51451506449
Dados: 2019.11.06 15:27:13 -03'00'

Prof. Jonny Carlos da Silva, Dr. Eng.
Coordenador do Programa

Documento assinado digitalmente
Roberto Simoni
Data: 06/11/2019 10:59:56-0300
CPF: 033.398.969-41

Prof. Roberto Simoni, Dr. Eng.
Orientador

Documento assinado digitalmente
Daniel Martins
Data: 07/11/2019 15:54:58-0300
CPF: 711.746.219-15

Prof. Daniel Martins, Dr. Eng.
Coorientador

Florianópolis, 23 de junho de 2019.

To all special people that always come up in my life.

ACKNOWLEDGEMENTS

Principalmente aos meus pais, Vilson e Irineusa, que são os motivadores dessa caminhada, colocando os estudos como prioridade, e o trabalho sério como necessidade. Aos meus avós pela constante prontidão em me ajudar e amar, especialmente ao meu Vô Deonildo pela bondade de ter me levado desde criança à madereira onde o gosto pelas máquinas começou. À minha Vó Irani pelo exemplo de humildade, sabedoria de vida e força pra lidar com as adversidades sem perder o sorriso. Aos meus irmãos, Iéie, Tis e Ale, que não me deixam esquecer de ser criança, quero sempre ser assim com vocês.

À minha namorada, Eli, que está comigo desde antes da engenharia, já crescemos muito juntos e vamos continuar nessa caminhada, hoje você é meu alicerce e fonte de motivação.

Aos meus amigos que fazem a vida ser tão mais leve, especialmente àqueles que conheci ou pude continuar próximo durante o mestrado, muito obrigado Elias, Estevan, Fábio, Fabiola, Gregori, Julio, Luan, Leandro, Marcel, Marco, Marina, Mateus, Rodrigo, Rogério, Seu Marcos, Thais, Vini e Zen.

Aos meus orientadores, Roberto Simoni e Daniel Martins, pelos conhecimentos transmitidos, conselhos e ajudas fornecidas durante esse período. À outros professores que também se mostraram importantes durante os trabalhos, obrigado Henrique Simas, Rodrigo Vieira e Andrea Carboni.

Ao Laboratório de Robótica Aplicada (LAR) e seus membros que de fato fazem o LAR ser um lar, obrigado, e espero continuar ajudando, ainda temos muito para conquistar.

Ao Programa de Pós-Graduação em Engenharia Mecânica da UFSC, pela oportunidade e apoio indispensável, e à CAPES pelo apoio financeiro por meio do Projeto de Tecnologia Assistiva PGPTA 59/2014.

Viver e não ter a vergonha de ser feliz,
cantar e cantar e cantar
a beleza de ser um eterno aprendiz.
(Gonzaguinha)

RESUMO

Mecanismo auto-alinhante é uma classe de mecanismos que não possuem restrições redundantes. As juntas destes mecanismos fornecem graus de liberdade para o sistema de maneira que a montagem é facilitada e livre de esforços internos causados pelas imprecisões de fabricação. Desta forma, os projetistas podem desenvolver mecanismos ou estruturas com tolerâncias mais largas, reduzindo os custos de fabricação. Este trabalho utiliza o Método de Davies e Teoria de Matroide conjuntamente. O Método de Davies é baseado na teoria de grafos e helicoides, e uma de suas matrizes é utilizada para gerar um matroide. Teoria de Matroide é um braço da matemática que avalia a independência linear em um espaço vetorial, quando aplicado para mecanismos é capaz de eliminar automaticamente as restrições redundantes. Nesse contexto, um mecanismo com uma ou mais restrições redundantes pode ser modelado estaticamente em forma matricial pelo Método de Davies, e pela Teoria de Matroide todos os mecanismos auto-alinhantes equivalentes ao original são enumerados sem restrições redundantes. O número de mecanismos auto-alinhantes enumerado por matroide pode ser alto dependendo da complexidade do mecanismo original, além de que alguns destes mecanismos enumerados não são factíveis. Atualmente, o algoritmo guloso é utilizado para escolher um mecanismo auto-alinhante entre todos os enumerados. Este trabalho propõe um novo método de seleção para ser aplicado ao conjunto de mecanismos auto-alinhantes enumerados por matroide. O método proposto seleciona um grupo destes mecanismos que atendem os requisitos de projeto determinados pelo projetista. O método proposto é então aplicado em mecanismos com restrições redundantes que estão presentes em camas hospitalares.

Palavras-chave: Auto-alinhamento, Teoria de Matroide, Mecanismos, Método de Davies.

ABSTRACT

Self-aligning mechanism is a class of mechanisms that do not have redundant constraints. The joints of these mechanisms provide degrees of freedom for the system. So the assembly is facilitated and free of internal stresses caused by manufacturing inaccuracies. In this way, designers can develop mechanisms or structures with wider tolerances, reducing manufacturing costs. This work uses the Davies Method and Matroid Theory. Davies' Method is based on Graph Theory and Screw Theory. Matroid Theory is a branch of mathematics that evaluates linear independence in a vector space, it is able to automatically eliminate redundant constraints when applied to mechanisms. In this way, a mechanism with one or more redundant constraints can be statically modeled in matrix form by Davies' Method, and creating a matroid by Matroid Theory. All self-aligning mechanisms equivalent to the original are enumerated. Depending on the complexity of the original mechanism, the number of self-aligning mechanisms enumerated by matroid can be high, and some of these mechanisms are unfeasible. Nowadays, the greedy algorithm is used to choose a self-aligning mechanism among all enumerated. This work proposes a new selection method to be applied to the set of self-aligning mechanisms enumerated by matroid. The proposed method selects a group of these mechanisms that comply with the design requirements determined by the designer. The proposed method is then applied in overconstrained mechanisms which are present in hospital beds.

Keywords: Self-aligning, Matroid Theory, Mechanisms, Davies' Method.

RESUMO EXPANDIDO

INTRODUÇÃO

Essa dissertação propõe um método para selecionar mecanismos autoalinhantes que foram enumerados por meio da Teoria de Matroides. Para aplicar o método é necessário que um mecanismo super-restrito seja modelado estaticamente pelo Método de Davies. Teoria de Matroide é então empregada para enumerar todas as possibilidades de mecanismos autoalinhantes. Estes mecanismos autoalinhantes são cinematicamente equivalentes com o mecanismo super-restrito utilizado como mecanismo original. O método proposto utiliza requisitos de projeto como ferramenta de decisão para selecionar um grupo de mecanismos autoalinhantes que satisfazem os requisitos definidos. Para isso, os requisitos de projeto são transformados em critérios de seleção. Os critérios de seleção são então aplicados em todos os mecanismos autoalinhantes enumerados criando um conjunto de mecanismos viáveis. Uma vez que o método é apresentado, ele é aplicado em dois modelos diferentes de camas hospitalares.

OBJETIVOS

O principal objetivo deste trabalho é propor um método para selecionar mecanismos autoalinhantes que foram enumerados por meio da Teoria de Matroides utilizando um mecanismo super-restrito como mecanismo original. A seleção é baseada em requisitos de projeto definidos pelos projetistas. Para alcançar o objetivo principal, alguns objetivos específicos foram traçados: revisar Método de Davies e Teoria de Matroide; propor um método de seleção de mecanismos autoalinhantes; aplicar Método de Davies em mecanismos super-restritos para avaliar as restrições do sistema; aplicar Teoria de Matroide para gerar todas as possibilidades de mecanismos autoalinhantes a partir de um mecanismo super-restrito; estabelecer requisitos de projeto para os mecanismos autoalinhantes; transformar os requisitos de projeto em critérios de seleção; selecionar conjuntos de mecanismos autoalinhantes por meio do método proposto; exemplificar os mecanismos autoalinhantes selecionados.

METODOLOGIA

O procedimento adotado no desenvolvimento dessa dissertação pode ser organizado em quatro etapas, Na primeira etapa mecanismo que possui restrições redundantes é modelado estaticamente por meio do Método de Daives, assim o número de restrições redundantes, mobilidades e outras características do mecanismo pode ser avaliada. A matriz de ações em rede é então convertida em um matroide. Esse matroide é formado por um par ordenado constituído de um conjunto que possui todas as restrições do mecanismo super-restrito, e uma família de conjuntos. Esta família é formado por conjuntos de restrições que são linearmente independentes entre si, chamados de base. Um mecanismo autoalinhante é criado a partir de um conjunto de restrições linearmente independente, ou seja as restrições que estão contidas em uma base estão relacionadas com as restrições de um mecanismo autoalinhante equivalente ao mecanismo original. Desta maneira a família de bases está relacionada com todos os mecanismos autoalinhantes possíveis derivados do mecanismo original. Nessa etapa também é necessária a definição dos requisitos de projeto que os mecanismos autoalinhantes devem atender. O conjunto de mecanismos autoalinhantes enumerados por matroide e os requisitos de projeto são as entradas do método proposto, a saída do método será um conjunto de mecanismos autoalinhantes que atendem os requisitos de projeto.

Na segunda etapa as bases são organizadas de forma matricial, de maneira que cada coluna da matriz está relacionada com uma restrição do mecanismo original, enquanto as linhas da matriz estão relacionadas com as bases dos mecanismos autoalinhantes enumerados por matroide. Na terceira etapa os requisitos de projeto são transformados em critérios de seleção. Os critérios de seleção são criados de maneira a avaliar todas as linhas da matriz criada na etapa anterior. Os critérios são geralmente criados utilizando funções binárias, mas outras funções podem ser utilizadas como inequações ou álgebra booleana. Dependendo da maneira como o requisito de projeto é definido, os critérios podem avaliar uma junta apenas ou todo o sistema. Cada um dos requisitos de projeto são transformados em um critério de seleção e cada critério cria um subconjunto de bases que atendem os critérios.

Na quarta etapa é realizada a intersecção entre os subconjuntos, criando assim um outro subconjunto, chamado de subconjunto final. O subconjunto final irá conter os mecanismos autoalinhantes que atendem a todos os requisito de projeto propostos. Pode ocorrer do subconjunto ser vazio, significando que nenhum mecanismo autoalinhante cinematicamente equivalente com o original atende a todos os requisitos de projeto, nesse caso, os requisitos de projeto precisam ser reavaliados. O método proposto foi aplicado em 4 mecanismos super-restritos diferentes e os resultados são discutidos na próxima seção deste resumo expandido.

RESULTADOS E DISCUSSÃO

O método proposto foi aplicado a quatro estudos de casos, esses estudos de casos estão relacionados com mecanismos de camas hospitalares. Os mecanismos de ajuste de encosto e e os mecanismos de ajustes das pernas de dois modelos foram usados. O primeiro modelo é de uma cama hospitalar comercial produzida pela empresa Linet, enquanto o segundo modelo é uma cama hospitalar desenvolvida pelo Laboratório de Robótica Aplicada (LAR) da UFSC. O primeiro estudo de caso é referente ao mecanismo de ajuste das costas do primeiro modelo de camas, este mecanismo possui originalmente 10 juntas, cada uma das juntas possui cinco restrições, totalizando 50 restrições no mecanismo original. Após a aplicação do Método de Davies foi descoberto que esse mecanismo possui um grau de liberdade e nove restrições redundantes. Um matroide foi criado a partir da matriz de ações em rede respectiva. Esse matroide listou 838,451 bases, ou seja 838,451 mecanismos autoalinhantes cinematicamente equivalente ao original. Três requisitos de projeto foram definidos para esse estudo de caso, que foram transformados em critérios de seleção. Os criteŕios de seleção criaram três subconjuntos de bases que atendem os critérios. O subconjunto final foi criado a partir da intersecção entre os três subconjuntos, este é constituído por 10 bases, ou seja 10 mecanismos autoalinhantes cinematicamente equivalentes ao original e que atendem os requisitos de projeto, um exemplo desses 10 mecanismos é mostrado.

O segundo estudo de caso é referente ao mecanismo de ajuste das pernas da cama hospitalar produzida pela empresa Linet. Esse mecanismo possui originalmente oito juntas, cada uma com cinco restrições, totalizando 40 restrições no mecanismo original. Após a aplicação do Método de Davies foi descoberto que esse mecanismo possui dois graus de liberdade e seis restrições redundantes. Um matroide foi criado a partir da matriz de ações em rede respectiva. Esse matroide listou 15,704 bases, ou seja 15,704 mecanismos autoalinhantes cinematicamente equivalente ao original. Três requisitos de projeto foram definidos para esse estudo de caso, que foram transformados em critérios de seleção. Três critérios de seleção foram criados a partir dos requisitos de projeto, assim três subconjuntos que atendem cada uma dos requisitos de projeto foram criados. O subconjunto final foi criado a partir da intersecção entre os três subconjuntos, este é constituído por 78 bases, ou seja 78 mecanismos autoalinhantes cinematicamente equivalentes ao original. Esses 78 mecanismos também atendem todos os requisitos de projeto. Um exemplo de desses 78 mecanismos é mostrado.

O terceiro estudo de caso é referente ao mecanismo de ajuste das costas do modelo de cama hospitalar desenvolvido no LAR. . Esse mecanismo possui originalmente dez juntas, cada uma com cinco restrições, totalizando 50 restrições no mecanismo original. Após a aplicação do Método de Davies foi descoberto que esse mecanismo possui um grau de liberdade e nove restrições redundantes. Um matroide foi criado a partir da matriz de ações em rede respectiva. Esse matroide listou 773,212 bases, ou seja 773,212 mecanismos autoalinhantes cinematicamente equivalente ao original. Três requisitos de projeto foram definidos para esse estudo de caso, que foram transformados em critérios de seleção. Três critérios de seleção foram criados a partir dos requisitos de projeto, assim três subconjuntos que atendem cada uma dos requisitos de projeto foram criados. O subconjunto final foi criado a partir da intersecção entre os três subconjuntos, este é constituído por oito bases, ou seja oito mecanismos autoalinhantes cinematicamente equivalentes ao original. Esses oito mecanismos também atendem todos os requisitos de projeto. Um exemplo de desses oito mecanismos é mostrado.

O quarto estudo de caso é referente ao mecanismo de ajuste das pernas do modelo de cama hospitalar desenvolvido no LAR. . Esse mecanismo possui originalmente oito juntas, cada uma com cinco restrições, totalizando 40 restrições no mecanismo original. Após a aplicação do Método de Davies foi descoberto que esse mecanismo possui dois grau de liberdade e seis restrições redundantes. Um matroide foi criado a partir da matriz de ações em rede respectiva. Esse matroide listou 21,988 bases, ou seja 21,988 mecanismos autoalinhantes cinematicamente equivalente ao original. Três requisitos de projeto foram definidos para esse estudo de caso, que foram transformados em critérios de seleção. Três critérios de seleção foram criados a partir dos requisitos de projeto, assim três subconjuntos que atendem cada uma dos requisitos de projeto foram criados. O subconjunto final foi criado a partir da intersecção entre os três subconjuntos, este é constituído por quatro bases, ou seja quatro mecanismos autoalinhantes cinematicamente equivalentes ao original. Esses quatro mecanismos também atendem todos os requisitos de projeto. Um exemplo de desses quatro mecanismos é mostrado.

CONSIDERAÇÕES FINAIS

A principal contribuição deste trabalho é o método de seleção proposto para mecanismos autoalinhantes enumerados por matroide. O método se mostrou efetivo para selecionar os mecanismos de acordo com os requisitos de projeto, visto que o conjunto final possui

uma quantidade muito menor de mecanismos quando comparado com a quantidade total enumerada. Alguns exemplos de requisitos de projeto foram utilizados, como por exemplo, juntas atuadas que não podem ser modificadas, modificação das juntas por meio de imposição de folgas. Outros requisitos de projetos e critérios de seleção podem ser criados.

A rigidez de mecanismos autoalinhantes pode ser afetada, isto que os mecanismos autoalinhantes possuem menos restrições do que mecanismos super-restritos equivalentes. Então uma análise de rigidez pode ser feita para os mecanismos autoalinhantes selecionados pelo método proposto, assim o mecanismo autoalinhante com maior rigidez pode ser escolhido.

Em alguns estudos de casos, restrições de uma junta foram substituídas por graus de liberdade por meio de adição de folgas. Portanto pode ser realizada uma análise de erro de posição de acordo com as folgas entre os mecanismos selecionados pelo método proposto, assim o mecanismo autoalinhante com menor erro de posição pode ser escolhido.

Palavras-chave: Auto-alinhamento, Teoria de Matroide, Mecanismos, Método de Davies.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\$$	Screw
$\$^m$	Twist or Motion Screw
ω	Angular Velocity Vector
ω_x	Angular Velocity around x-axis
ω_y	Angular Velocity around y-axis
ω_z	Angular Velocity around z-axis
V_p	Linear Velocity Vector
V_{px}	Linear Velocity along x-axis
V_{py}	Linear Velocity along y-axis
V_{pz}	Linear Velocity along z-axis
r	Translational Freedom along x-axis
s	Translational Freedom along y-axis
t	Translational Freedom along z-axis
u	Rotational Freedom around x-axis
v	Rotational Freedom around y-axis
w	Rotational Freedom around z-axis
τ	Velocity Parallel to the Screw axis
h	Pitch of the Screw
S_o	Position Vector
$\widehat{\$}^m$	Normalized Twist
Ψ	Magnitude of the Screw
$\$^a$	Wrench or Action Screw
R	Force Vector
R_x	Force along x-axis
R_y	Force along y-axis
R_z	Force along z-axis
T_p	Moment Vector
T_{px}	Moment around x-axis
T_{py}	Moment around y-axis
T_{pz}	Moment around z-axis
O	Origin System
R	Force constraint along x-axis
S	Force constraint along y-axis
T	Force constraint along z-axis
U	Moment constraint around x-axis
V	Moment constraint around y-axis
W	Moment constraint around z-axis
G_C	Coupling Graph
$[B]_{\nu,e}$	Circuit Matrix
ν	Number of Circuits of the Coupling Graph
e	Number of Edges of the Coupling Graph
$b(i,j)$	Elements of the Circuit Matrix
$[Q]_{k,e}$	Cut-set Matrix
k	Number of Cut-sets
$q(i,j)$	Elements of the Cut-set Matrix

$[\widehat{M}_D]_{(\lambda,F)}$	Unit Motion Matrix
$[\widehat{A}_D]_{(\lambda,C)}$	Unit Action Matrix
λ	Dimension of Screw System
F	Gross Degree of Freedom of a Coupling Network
f_i	Degree of Freedom of the Coupling i
C	Gross Degree of Constraint of Coupling Network
c_i	Degree of Constraint of the Coupling i
ca_i	Degree of Actuated Constraint
$[\widehat{M}_N]_{(\lambda\nu,F)}$	Network Unit Motion Matrix
$[A_N]_{\lambda k,C}$	Network Unit Action Matrix
$[B_i]_{F,F}$	Diagonal Matrix Corresponding to Row i of the Circuit Matrix
$[Q_i]_{(C,C)}$	Diagonal Matrix Corresponding to Row i of the Cut-set Matrix $[Q]_{(k,C)}$
F_N	Degree of Freedom of a Coupling Graph
C_N	Degree of Redundant Constraint of a Coupling Network
G	Graph
E	Edge Set of Graph G
X	Subset of E or T_c
T_c	Column Set of a Matrix
S_M	Ground Set of a Matroid $\mathcal{M}$
F_M	Subset of S_M Linearly Independent
$\mathcal{M}$	Matroid
$\mathcal{B}$	Bases of Matroid $\mathcal{M}$
$\mathcal{M}^*$	Dual Matroid
$\mathcal{B}^*$	Cobases of Matroid $\mathcal{M}$
$\mathcal{M}_{AN}$	Matroid derived from Matrix $[A_N]_{\lambda k,C}$
$\mathcal{M}^*_{AN}$	Dual Matroid derived from Matrix $[A_N]_{\lambda k,C}$
w_i	Weight relative to any constraint i
C'	Exact Gross Degree of Constraint
$[N]_{\mu,C}$	Cobases Binary Matrix
μ	Number of Cobases created by the Matroid $\mathcal{M}_{AN}$
$n(i,j)$	Elements of the Matrix $[N]_{\mu,C}$
l_i	Line i of Matrix $[N]_{\mu,C}$
α	Criterion α applied to Matrix $[N]_{\mu,C}$
K_α	Subset of Cobases which comply with Criterion α
β	Binary Condition
K_F	Final Subset

CONTENTS

1 INTRODUCTION

This dissertation proposes a method to select self-aligning mechanisms enumerated by Matroid theory. To apply the method, a seed mechanism with redundant constraints must be statically modeled by Davies' Method. Then Matroid Theory is employed to enumerate all possibilities of self-aligning mechanisms. The proposed method uses design requirements as deciding instruments. The design requirements are transformed in selection criteria. The selection criteria are then applied to all enumerated self-aligning mechanism, creating sets of feasible mechanisms. Once the method is presented, it is applied in two models of hospital beds.

This introduction contextualizes how the method can be applied in a design methodology. Also, some reasons are presented for designers to adopt self-aligning mechanisms. Finally, the work objectives and structure are presented.

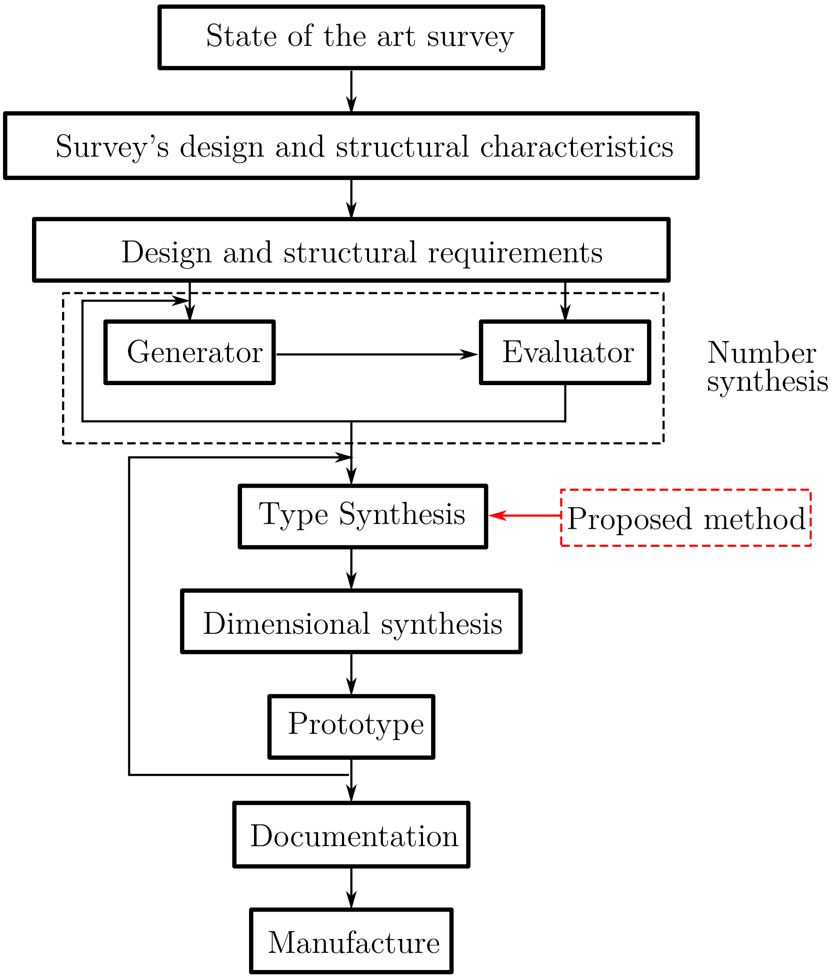
1.1 CONTEXTUALIZATION

The design of a machine is more than creating parts and putting them together, all the design processes are important and influenced by design requirements, manufacturing, logistics, and other external factors. Therefore it is important to elaborate and to improve methods to facilitate the designing processes because effective methods decrease the possibility of failure.

Machine design consists of movable mechanisms with supports for transmitting motions and forces, while mechanism design is concerned mainly with the generation or selection of a particular type of mechanisms (YAN, 1998) . Thus, machine design includes mechanism design. Several methodologies were created to guide the design process of mechanisms, such as the methodologies developed by Hartenberg and Denavit (1964), Yan (1998), Tsai (2000) and Murai (2013).

Figure 1 illustrates a flowchart which represents the methodology structure proposed by Murai (2013) . In this methodology, the state of art survey guides the definition of the design and structural requirements of the mechanism. Then all possible mechanisms are generated and evaluated, the unfeasible are eliminated. The stage of generation and evaluation is named as Number Synthesis. The output of the Number Synthesis is the structure of the mechanism which is the input for the Type Synthesis. Type synthesis is the stage where the type of each kinematic pair is determined.

Figure 1 – Murai's methodology.



Adapted from Murai (2013).

Aiming to create self-aligning mechanisms, the method proposed in this work can be applied during Type Synthesis phase. After Type Synthesis, the dimensions are determined by Dimensional Synthesis, that is important to the mechanism perform the motions according to the design requirements. Then, the mechanism is prototyped. If the prototype is satisfactory, the mechanisms can be documented and manufactured (MURAI, 2013).

During the type synthesis phase, the type of joints are established, hence the freedoms and constraints of each joint are determined. The type synthesis can result in overconstrained or self-aligning mechanisms. In a few words, an overconstrained mechanism has unnecessaries constraints while a self-aligning mechanism has the exact number of necessary constraints to determine the mechanism statics (RESHETOV, 1982).

Although self-aligning mechanisms have the exact number of constraints, these constraints cannot be randomly chosen. In case of a new mechanism, the constraints must allow the motions required in the design requirements. When the designer desires to transform an overconstrained (seed) in a self-aligning mechanism, the last must allow the same motions of the seed mechanism.

For a mechanism be classified as self-aligning, the designer must guarantee redundant constraint are not present in the constraint system. But to define if a constraint is redundant is not easy because it depends on the other constraints and the distance among the joints. Thus using self-aligning mechanisms is still obscure to the majority of designers, as well as the special characteristics of these mechanisms.

Some machines have a complex assembly, therefore the parts must be manufactured by accurate processes and even so the assembly may not be executed or it can create internal stresses decreasing the lifespan of parts. The main strength of self-aligning mechanisms is the assembly process because until the last joint is assembled, the links will have six degrees of freedom allowing to assembly the last part without creating internal loads. Therefore, the assembly will be completed even if the parts are manufactured with dimensional errors (RESHETOV, 1982).

Given an overconstrained mechanism modeled statically by Davies' method (CAZANGI, 2008), all possibilities of self-aligning mechanisms derived from the overconstrained mechanism can be enumerated employing Matroid theory (CARBONI, 2015). Matroid theory finds sets of constraints which are not redundant among them, and each set is related to a self-aligning mechanism. Some self-aligning mechanisms are not feasible because they do not comply with the design requirements and they must be discarded. The method proposed in this work is useful to select the group of these self-aligning mechanisms which are complying with design requirements.

1.2 OBJECTIVES

The main objective of this work is to propose a method to select self-aligning mechanisms enumerated by Matroid theory using an overconstrained mechanism as seed mechanism. The selection is based on design requirements established by the designers. To achieve the main objective, some specific objectives are determined:

- to review Davies' method and Matroid theory;

- to propose a method of selecting self-aligning mechanisms;

- to apply Davies' method in overconstrained mechanisms to evaluate the constraints systems;

- to apply Matroid theory to generate all possibilities of self-aligning mechanisms from the overconstrained mechanisms;

- to establish design requirements for the self-aligning mechanisms;

- to transform design requirements in selection criteria;

- to select sets of self-aligning mechanisms by means of the proposed method;

- to exemplify the self-aligning mechanisms selected.

1.3 WORK STRUCTURE

The remainder of this work is organized as follow.

Chapter 2 presents the mathematical tools employed in this work. Firstly Screw Theory, where twists and wrenches are reviewed, then Davies' method is introduced. Although static modeling is sufficient to apply the proposed method, both static and kinematic modeling by Davies' method. The method is demonstrated until creating the network unit matrices. Concepts of Matroid theory are demonstrated, a graph and a matrix are used as example.

Chapter 3 presents a review about redundant constraints analysis. Reshetov method is reviewed. Then, it is presented a review about the method proposed by Carboni (2015). In this method Matroid theory is applied to enumerate all self-aligning mechanisms.

Chapter 4 presents the selection method proposed in this work. The method has two inputs, an input is the cobases enumerated by matroid, according to the method proposed by Carboni (2015). Another input is the design requirements established by the designers.

Chapter 5 presents two case studies. The focus application is hospital beds. Two models were elected, and the backrest adjustment and the leg rest adjustment mechanisms were analyzed. Design requirements were presented and applied into the proposed method.

Chapter 6 presents the conclusion. The main steps regarding the bibliography and the proposed method are reviewed, as well as the results of the case studies. Final considerations and further steps of this research are also presented.

Appendix A presents kinematic pair as well as the constraints and degrees of freedom of them. The kinematic pairs are classified according to the number of constraints which is agreement to Reshetov (1982).

Appendix B presents the Matlab programs which model statically the mechanisms used in the case studies. The commands used in the Sagemath to create the matroids are also shown in Appendix B.

2 MATHEMATICAL TOOLS

This chapter presents the review on the mathematical tools employed in this work. Section 2.1 approaches Screw theory and Davies' method. Then, the application of these theories to analize the motion and action of the mechanisms are presented. Davies' method is applied to a steering system aiming to analyze the redundant constraints and mobility of the system. Subsection 2.2 reviews Matroid theory, the concepts are introduced and examples using graphs and matrices are presented.

2.1 SCREW THEORY AND DAVIES' METHOD

Screw theory allows to express displacements, velocities, forces, and torques in three-dimensional space. It is based on the idea that any rigid body motion can be represented as the inseparable union of a rotation about an axis and a translation along the same axis. The axis is coincident with the object or particle undergoing displacement (GALLARDO-ALVARADO, 2016) . The Screw theory is founded upon two celebrated theorems.

The first one relates to the displacements of a rigid body (BALL, 1998). This theorem is attributed to Chasles (1830), but Mozzi (1763) seems to be the first one to give a clear and exact approach on general motion (CECCARELLI, 2000).

The second theorem was discovered by Poinsot. It says that any system of forces which act upon a rigid body can be replaced by a single force and a torque in a plane perpendicular to the force (BALL, 1998). These contributions were discovered before the 19^{th} century, after this, many other researchers advanced the development of the Screw Theory, such as Ball (1900), Waldron (1966) and Hunt (1978).

A screw $\$$ may be understood as two concatenated vectors. The first vector, named primal, is a unit vector along the axis of the screw, while the second vector, named dual, is the moment produced by the first vector about a reference point. Geometrically, a screw is a line l with a scalar pitch h.

For a rigid body, the screw related to the motion in a fixed frame is called *twist* $\$^m$, the superscript m is related to motion. The twist can be expressed as follows:

$$\$^m = \left[\begin{array}{c} \omega \\ \hline V_p \end{array}\right] = \left[\begin{array}{c} \omega_x \\ \omega_y \\ \omega_z \\ \hline V_{px} \\ V_{py} \\ V_{pz} \end{array}\right] \tag{2.1}$$

Where the primal vector, $\omega = [\omega_x\, \omega_y\, \omega_z]^T$, represents the angular velocity in the fixed frame, and the dual vector, $V_p = [V_{px}\, V_{py}\, V_{pz}]^T$, represents the linear velocity of a point of the body at the origin O. The notation adopted in this work follows the Davies notation, so Equation 2.1 can be written as $\$^m = [r\ s\ t \mid u\ v\ w]^T$.

The dual vector V_p results from the sum of two components, the first, τ, is parallel to the screw axis and proportional to the angular velocity. The constant of proportionality is the pitch h of the screw \$, so $\tau = h\omega$. And the second component is normal to the screw axis, $S_o \times \omega$, where S_o is the position vector of any point on the screw axis. So, a normalized twist $\widehat{\$}^m = \$^m/\psi$ can be expressed as a pair of vectors:

$$\widehat{\$}^m = \begin{bmatrix} \omega \\ S_o \times \omega + h\omega \end{bmatrix} \tag{2.2}$$

where ψ is the magnitude of the screw, it is later explained in this section.

For a rigid body the screw related to the action force in a fixed frame is called of *wrench*, $\$^a$, the superscript a is related to action. The wrench can be expressed as follows:

$$\$^a = \left[\begin{array}{c} T_p \\ \hline R \end{array}\right] = \left[\begin{array}{c} T_{px} \\ T_{py} \\ T_{pz} \\ \hline R_x \\ R_y \\ R_z \end{array}\right] \tag{2.3}$$

Where the primal vector $R = [R_x\, R_y\, R_z]^T$ represents a force vector acting on the body in the fixed frame, and the dual vector $T_p = [T_{px}\, T_{py}\, T_{pz}]^T$, represents the moment vector acting on a point of the body at the origin O. Following Davies' notation the Equation 2.3 can be written as $\$^a = [R\ S\ T \mid U\ V\ W]^T$.

In action, such as in motion, the dual vector T_p is the sum of two components, one parallel to the screw axis which is a function of the pitch h, hR, and another normal to the screw axis $S_o \times \omega$. So, a normalized wrench $\widehat{\$^a}$ can be expressed as a pair of vectors:

$$\widehat{\$}^a = \begin{bmatrix} S_o \times R + hR \\ R \end{bmatrix} \tag{2.4}$$

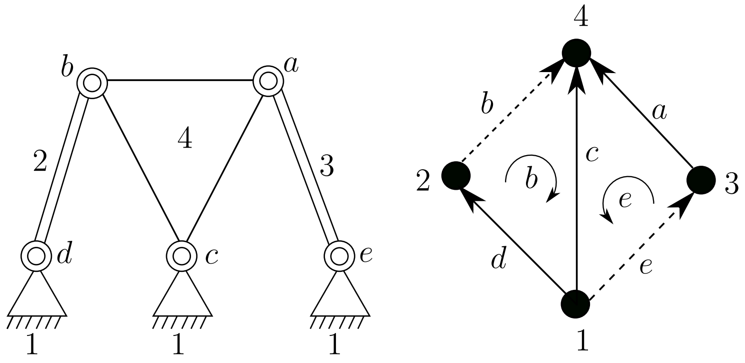
Some considerations must be made to determine the pitch of a screw. If a twist or a wrench is related to a pure angular velocity or a pure force, respectively, the screw have a zero pitch, $h = 0$. But, if it is a pure linear velocity or a pure torque, respectively, the screw have infinite pitch, $h \to \infty$.

For Davies' method, it is convenient to use the screw as a magnitude multiplied by a normalized screw. The magnitude ψ is equal to the norm of the primal vector if the motion is a pure rotation, *i.e.* $\psi = ||\omega||$, or if the action is a pure force, $\psi = ||R||$. The magnitude is equal to the norm of the dual vector if the motion is a pure translation, $\psi = ||V_p||$, or if the action is a pure torque, $\psi = ||T_p||$.

The Screw Theory can describe the displacements and the forces of any rigid body. Davies' Method uses Screw Theory to determine the kinematics and statics of any multibody systems. In addition to Screw Theory, Davies method uses Graph Theory to adapt the Kirchhoff's laws for a network of links and couplings.

The adaptation of Kirchhoff's laws is based on the representation of a coupling network with n links and j joints by a graph, called coupling graph G_C, wherein each body (link) is represented by a node and a number, and each direct coupling (joint) is represented by an edge and a letter (DAVIES, 2006).

Figure 2 – Example of coupling graph.



(a) Multibody system

(b) Coupling graph.

Adapted from Davies (2006)

Figure 3(b) shows an example of a 2-circuit network coupling graph that represents the coupling network of the structure shown in Figure 3(a). To use Davies' method, the graphs must be directed graphs, thus the direction of the edge is indicated by an arrow. The choice of orientation is arbitrary.

The branches a, c and e creates the chosen tree relative to graph G_C shown in Figure 3(b), these edges are represented by continuous lines, while the edges b and e are chords which are represented by the dashed lines. Without the chords, the circuits would be arcs. Each circuit is named with the letter corresponding to the chord which closes it. The positive sense attributed to the circuits are arbitrary.

To Davies' methods, as in Kirchhoff' laws, the circuits and cut-set of the graphs are evaluated in matricial form. The circuit matrix $[B]_{\nu,e}$ of a graph indicates the presence of the edges in the circuits. Considering the graph G_C, Figure 3(b), that has two loops, $\nu = 2$, and five edges, $e = 5$. The circuit matrix $[B_{G_C}]_{2,5}$ represents the circuits of this graph. The elements $b(i,j)$ are arranged according to the presence and orientation of the edges in relation to the loop analysed.

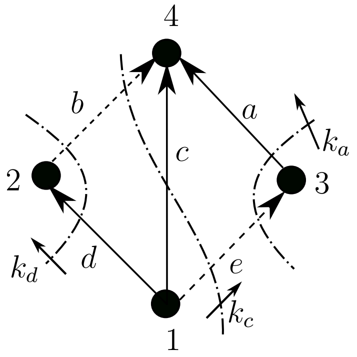
$$b_{(i,j)} = \begin{cases} 1 & \text{if the edge } j \text{ is in the circuit } i \text{ and in the} \\ & \text{same direction} \\ -1 & \text{if the edge } j \text{ is in the circuit } i \text{ and in the} \\ & \text{opposite direction} \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

The circuit matrix is arranged according to the motion graph Davies (2006). The circuit matrix $[B_{G_C}]_{(2,5)}$, relative to the graph of Figure 3(b) is arranged as follows:

$$[B_{G_C}]_{(2,5)} = \begin{array}{c} \\ b \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 1 & -1 & 1 & 0 \\ 1 & 0 & -1 & 0 & 1 \end{bmatrix} \end{array} \tag{2.6}$$

The cut-set matrix $[Q]_{(k,e)}$ demonstrates the presence of the edges in the cut-sets. The number of chords is equal to the number of circuits, while the number of branches is the same number of cut-sets. A cut is a partition of the vertices of a graph, so a cut-set is the set of edges that have one endpoint in the partition. A cut-set contains one branch and any number of chords (DAVIES, 2006).

Figure 3 – Graph G_C with cutsets.



Adapted from Davies (2006)

The cut-set matrix $[Q_{G_C}]_{3,5}$ is arranged according to the graph G_C and the cut-sets shown in Figure 3, values $q_{(i,j)}$ of the matrix follows Equation 2.7.

$$q_{(i,j)} = \begin{cases} 1 & \text{if the edge } j \text{ is in the cut-set } i \text{ and in same} \\ & \text{direction} \\ -1 & \text{if the edge } j \text{ is in the cut-set } i \text{ and in} \\ & \text{opposite direction} \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

So the cut-set matrix $[Q_{G_C}]_{3,5}$ relative to the graph shown in Figure 2.7 is arranged as Equation 2.8.

$$[Q_{G_C}]_{3,5} = \begin{array}{c} \\ k_a \\ k_c \\ k_d \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & -1 & 0 & 1 & 0 \end{bmatrix} \end{array} \tag{2.8}$$

Together with graph theory, Davies' Method uses the twists and wrenches to arrange the unit motion matrix $[\widehat{M}_D]_{(\lambda,F)}$ and the unit action matrix $[\widehat{A}_D]_{(\lambda,C)}$, respectively. The subscript λ is the workspace of the screw system. Given a coupling which is in workspace λ, this workspace is composed by two dual terms, f_i, the degree of freedom of a coupling i, and c_i, the degree of constraint of a coupling i, as shown in Equation 2.9.

$$\lambda = f_i + c_i \tag{2.9}$$

Considering a system, F is the gross degree of freedom of a coupling network, defined by $F = \sum f_i$, and C is the gross degree of constraint of a coupling network, calculated by

$C = \sum c_i + \sum ca_i$, where ca_i is the degree of actuated constraint. Knowing the actuated joint is useful to this work. Instead of a constraint, the actuation is considered as a degree of freedom, so $\sum ca_i$ can be considered equal to zero.

For the unit motion matrix, $[\widehat{M}_D]_{(\lambda,F)}$, each column is related to one twist:

$$[\widehat{M}_D]_{(\lambda,F)} = \left[\begin{array}{ccccccccc} \widehat{\$}^m_{a1} & \widehat{\$}^m_{ai} & \vdots & \widehat{\$}^m_{b1} & \vdots & \ldots & \vdots & \widehat{\$}^m_F \end{array}\right] \tag{2.10}$$

For the unit action matrix, $[\widehat{A}_N]_{(\lambda,C)}$, each column is related to one wrench:

$$[\widehat{A}_D]_{(\lambda,C)} = \left[\begin{array}{ccccccccc} \widehat{\$}^a_{a1} & \widehat{\$}^a_{ai} & \vdots & \widehat{\$}^a_{b1} & \vdots & \ldots & \vdots & \widehat{\$}^a_F \end{array}\right] \tag{2.11}$$

For a system with dimension of workspace equal to λ and ν circuits, the circuit matrix $[B]_{(\nu,e)}$ and the unit motion matrix $[\widehat{M}_D]_{(\lambda,F)}$ allows the arrangement of the network unit motion matrix, $[\widehat{M}_N]_{(\lambda\nu,F)}$. The number of edges of the motion graph is the same number of gross degree of freedom, F, this is a mandatory condition. So, the arrangement of matrix $[\widehat{M}_N]_{(\lambda\nu,F)}$ follows Equation 2.12:

$$[\widehat{M}_N]_{(\lambda\nu,F)} = \begin{bmatrix} [\widehat{M}_D]_{(\lambda,F)}[B_1]_{F,F} \\ [\widehat{M}_D]_{(\lambda,F)}[B_2]_{F,F} \\ \vdots \\ [\widehat{M}_D]_{(\lambda,F)}[B_\nu]_{F,F} \end{bmatrix}_{\lambda\nu,F} \tag{2.12}$$

where $[B_i]_{F,F}$ are diagonal matrices with diagonal elements corresponding to row i of the circuit matrix $[B]_{(\nu,F)}$ (CAZANGI, 2008) .

An important matrix to this work is the network unit action matrix $[\widehat{A}_N]_{(\lambda k,C)}$. This matrix is arranged by the combination of the unit action matrix, $[\widehat{A}_D]_{(\lambda,C)}$, and the cut-set matrix, $[Q]_{(k,e)}$. In this arrangement, the gross number of constraints C, and the edges number of the action graph are necessarily the same. So the arrangement of matrix $[\widehat{A}_N]_{(\lambda k,C)}$ is according to Equation 2.13:

$$[\widehat{A}_N]_{(\lambda k,C)} = \begin{bmatrix} [\widehat{A}_D]_{(\lambda,C)}[Q_1]_{C,C} \\ [\widehat{A}_D]_{(\lambda,C)}[Q_2]_{C,C} \\ \vdots \\ [\widehat{A}_D]_{(\lambda,C)}[Q_k]_{C,C} \end{bmatrix}_{\lambda k,C} \tag{2.13}$$

where $[Q_i]_{(C,C)}$ are diagonal matrices with diagonal elements corresponding to row i of the cut-set matrix $[Q]_{(k,C)}$.

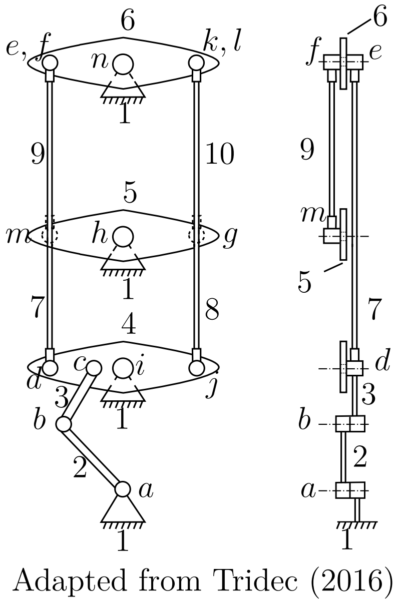
Screw theory and Davies' method are applied in mechanisms theory, mainly in static and kinematic analyses (CAZANGI, 2008). Laus (2011) applied Kirchhoff's equations for the study of mechanisms efficiency. Weihmann (2011) evaluates the wrench capabilities of humanoid robots in static or quasi-static conditions. Carboni (2015) applied the Davies method and matroid theory to determine the redundant constraints of a multibody system and to enumerate all solutions with equivalent constraints. The theoretical tools developed by Carboni (2015) are largely used in this work, and subsection 3.2 presents a review of these tools.

Since the second half of the XIXth century, when Chebychev (1853) proposed a mathematical formalization for the calculation of the mechanism mobility, several formulas and approaches have been proposed (GOGU, 2005b). The most well-known mobility criterion is the Grübler-Kutzbach formulation (GOGU, 2015a):

$$F_N = \lambda(n - j - 1) + \sum_{i=1}^{j} f_i \tag{2.14}$$

Where F_N is the mobility of the system. Equation 2.14 can be applied in the mechanism shown in Figure 4. This mechanism is related to the mechanical steering system TR of TRIDEC company (TRIDEC, 2016).

Figure 4 – Steering System.



Adapted from Tridec (2016)

The mechanism has 10 links and 14 revolute joints working in the planar space, thus $\lambda = 3$. Applying these values in Equation 2.14, $F_N = 3(10 - 14 - 1) + 14 = -1$, which is incorrect. The correct mobility is $F_N = 1$ (MANENTI, 2018). This mechanism has two redundant constraints, it will be explained in the next subsection. As a redundant constraint is defined by Reshetov (1982) as a constraint whose removal does not increase the mobility of the mechanism. In this case, can be observed that the number C_N of redundant constraints can affect the mobility calculation of the steering system (CARBONI, 2015).

Therefore, it is used the Modified Grübler-Kutzbach Criterion (HUANG et al., 2009):

$$F_N = \lambda(n - j - 1) + \sum_{i=1}^{j} f_i + C_N \tag{2.15}$$

for the correct calculation of the steering system shown in Figure 4, the number of redundant constraints must be equal to two, $C_N = 2$. In the Section 3.2 a method to evaluate correctly the number C_N is presented. In Appendix A, it is presented several kinematic pairs which are classified according to the degree of constraint.

2.2 MATROID THEORY

The concept of matroid will be easier to understand using as examples the graph G of Figure 5 and the matrix of Equation 2.16. The graph G has six edges which create the edge set $E = \{1, 2, 3, 4, 5, 6\}$. A subset $X \subseteq E$ of edges is circuit-free if and only if:

1. X does not contain the edge 6;
2. X contains at most two of the edges $1, 2, 3, 4$;
3. X contains at most one of the edges $1, 2$.

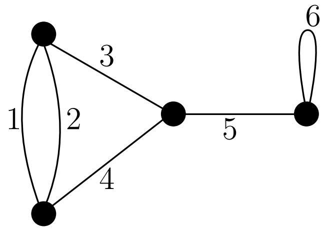


Figure 5 – Graph example

Considering the matrix of Equation 2.16, each column is represented by a letter and composes the column set $T_c = \{a, b, c, d, e, f\}$. A subset $X \subseteq T_c$ is linearly independent if and only if:

1. $f \notin X$;

2. $\{a, b, c, d\} \cap X$ has at most two elements;

3. $\{a, b\} \cap X$ has at most one element.

$$\begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \tag{2.16}$$

Now, we consider the pair (S_M, F_M), where F_M is a collection of subsets of S_M. If F_M satisfies some properties, then the pair (S_M, F_M) is called of *matroid*. The concept of matroid is a common generalization of graphs and matrices, so given a matroid $\mathcal{M}$, we cannot say that $\mathcal{M}$ came from a graph or a matrix. The properties which the subsets of F_M must satisfy are (RECSKI, 2013):

1. $\emptyset \in F$, *i.e.* the empty set must belong to F;

2. If $X \in F$ and $Y \subseteq X$ then $Y \in F$ must also hold;

3. If $X \in F$ and $Y \in F$ and $|X| > |Y|$, then there must exist an element $x \in X - Y$ so that $Y \cup \{x\} \in F$ also holds.

Returning to the column vectors of matrices, for any set S_M of vectors, (S_M, F_M) is a matroid if the linearly independent subsets form F_M. Using the matrix of Equation 2.16, F_M is established by the following subsets:

$$\begin{aligned} F_M = \{&\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, c\}, \\ &\{b, d\}, \{b, e\}, \{c, d\}, \{c, e\}, \{d, e\}, \{a, c, e\}, \{a, d, e\}, \\ &\{b, c, e\}, \{b, d, e\}, \{c, d, e\}\} \end{aligned} \tag{2.17}$$

The property (b) states that all subsets of an independent set are independent sets too. For example, the set $\{a, c, e\} \in F_M$ is an independent set whose subsets $\{a\}$, $\{c\}$, $\{e\}$, $\{a, c\}$ $\{a, e\}$ and $\{c, e\}$ are also independent. So, it is redundant to enumerate all the subsets of F_M. For matroid, it is useful to enumerate just the sets of F_M with maximal cardinality , called the Bases $\mathcal{B}$ of the matroid $\mathcal{M}$. The matroid originated by the matrix of Equation 2.16 has the family of bases $\mathcal{B}$ established by the following subsets:

$$\mathcal{B} = \{\{a, c, e\}, \{a, d, e\}, \{b, c, e\}, \{b, d, e\}, \{c, d, e\}\} \tag{2.18}$$

A matroid $\mathcal{M}$ can be determined by the pair $(S_M, \mathcal{B})$, where S is called the ground set of the matroid and $\mathcal{B}$ is the set of bases of $\mathcal{M}$. Every matroid has a dual denoted by $\mathcal{M}^* = (S_M, \mathcal{B}^*)$. The collection $\mathcal{B}^*$ exchange every set of $\mathcal{B}$ with its complement on the list of the bases, the complement is relative to the ground set S (RECSKI, 2013). The following Equation 2.19 lists the cobases, $\mathcal{B}^*$, respective to the bases $\mathcal{B}$ of Equation 2.18:

$$\mathcal{B}^* = \{\{b, d, f\}, \{b, c, f\}, \{a, d, f\}, \{a, c, f\}, \{a, b, f\}\} \tag{2.19}$$

3 REDUNDANT CONSTRAINT ANALYSIS

In this chapter, a bibliographic review about redundant constraint analysis is presented. Firstly, the concept of redundant constraint is dicussed, also the consequences of having a mechanism with redundant constraint are considered. Section 3.1 details the method created by Reshetov (1982) to evaluate redundant contraints and mobility of mechanisms.

Section 3.2 employs Davies' method to evaluate the presence of redundant constraint. Subsection 3.2.1 discuss about the method presented by Carboni (2015) which applies Matroid theory to enumerate all self-aligning mechanisms derived from a overconstrained mechanism. In this work the overconstrained mechanisms used to enumerate the self-aligning are named as seed mechanisms. Subsection 3.2.2 approaches the Greedy algorithm which is a tool employed to select bases according to set weights.

An object in the free state has six independent degrees of freedom of motion or position, three translational and three rotational motions. To design the connections between parts of a machine the six degrees of freedom must be considered. A good management of the degrees of freedom assist the efficient design of machines (BLANDING, 1999).

The design of the connections means the assembling of parts that work together forming a mechanical system. Disassembled, the parts are just several parts (WHITNEY, 2004).

Assemblies are challenging from both engineering and manufacturing point of view. In the manufacturing it is particularly difficult to attain precise dimensions of parts that are made, so tolerances are allowed beyond permissible values. Thus, the design process needs to select a mechanism that the accuracy requirements should be relatively low (WHITNEY, 2004; RESHETOV, 1982). In light of this, mechanisms staticly determined have self-aligning links and they are free of *redundant constraints*. Redundant constraints are those constraints whose elimination would not increase the mobility of a mechanism (RESHETOV, 1982).

With the point of view of assembling, the elimination of redundant constraints can be a design solution that can reduce the accuracy requirements of the manufacturing process. Moreover, the assembling is simpler compared to a similar mechanism with redundant constraints. Consequently the time and costs of the process can be decreased.

The design of a self-aligning mechanism implies the correct choices of kinematic pairs that constitute a kinematic chain. And the selection of pairs should not modify the mobility predetermined by the chain.

3.1 RESHETOV METHOD

This subsection presents the method proposed by Reshetov (1982) to evaluate the mobility and redundant constraints of a given mechanism. It was explained above that angular and linear deviations would not affect the performance of a mechanism free of redundant contraints. The linear and angular dimensions of the links can vary from nominal ones and a self-aligning mechanism will be assembled without preloads and interference fits.

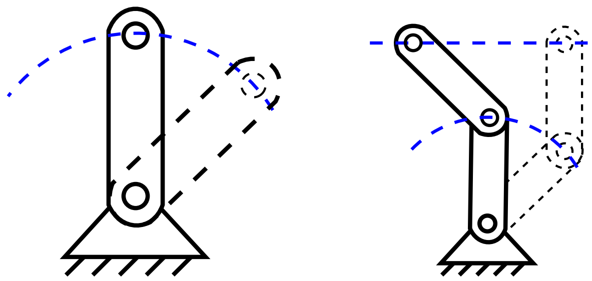
The assembling of a mechanism is the process of connecting the links through joints. To close a circuit the last joint must be assembled, so the both links which will be part of this joint must be aligned. This alignment must be linear and angular along and around the three orthogonal axes. It is useful to explain the reason to use spacial workspace ($\lambda = 6$) in a redundant constraint analysis.

A complete alignment between the links of a joint will be made if a link has six mobilities in relation to the other link. The absence of one of these mobilities results in internal tensions and in a redundant constraint. As this joint was not assembled yet the mobilities of the joint cannot be considered.

Before explaining the Reshetov method, there are more special points that requires special attention. The angular mobility allowed by a revolute joint in Figure 7(a) creates angular displacement forming an arc. However, a linear displacement can be created combining more than one angular displacement, Figure 7(b) . This linear displacement is always perpendicular to the rotation axes. So, linear displacement of links while assembling can be done not only owing linear mobility to the kinematic pairs, but rotating the links around an axis perpendicular to the direction of the linear mobility.

Based on these points Reshetov (1982) developed a method to evaluate the presence of redundant constraints in mechanisms. This method considers the loops of the mechanisms and the freedoms allowed by the joints. In this way, a loop can close without redundant contraints if all six freedoms are present and the translational freedom can be obtained by rotating the links in perpendicular axis. Once the replacements of the missing translation

Figure 6 – Displacements related to revolute joints



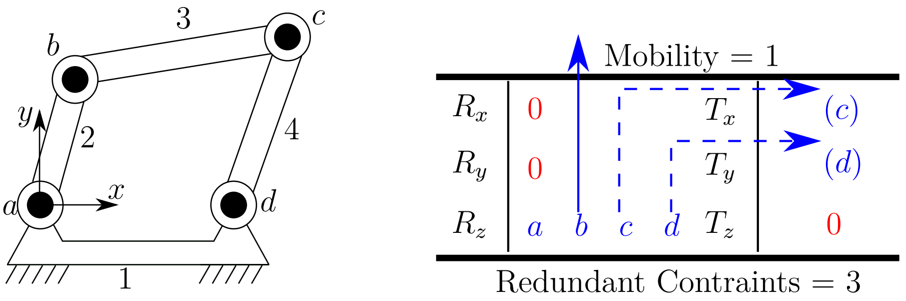
(a) One revolute joint

(b) Two revolute joints

is done, if one freedom still missing, this is a redundant constraint of the system. And, if a extra freedom is present, it indicates this freedom is a mobility of the mechanism.

Figure 7 ilustrates this method analyzing the four-bar mechanism. The four joints are revolute joints with rotational freedom around the z-axis. The rotational freedom related to joints c and d were used to provide translational freedom along x-axis and y-axis, respectively. The translational freedom along z-axis cannot be provided by the other joints because it is parallel to the rotational axis. So, the joint a and b provide rotational freedom around z-axis, one of these degrees of freedom can be considered as an extra mobility. In this case, the four-bar mechanism has mobility equal to one and three redundant constraints.

Figure 7 – Four-bar mechanism and Reshetov table



(a) Four-bar mechanism

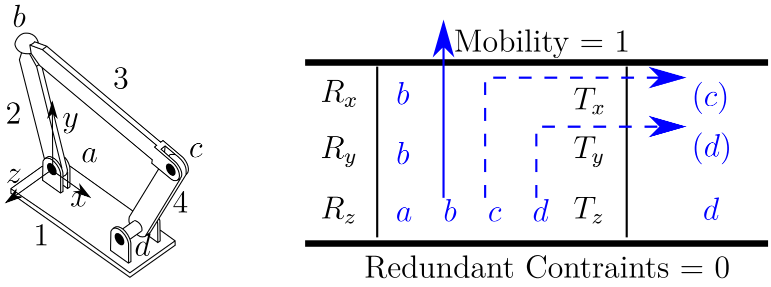
(b) Reshetov table

The values of the Reshetov table (Figure 7) can be applied to the Modified Grübler-Kutzbach Criterion, shown in Equation 2.15:

$$F_N = 6(4 - 4 - 1) + 4 + 3 \Rightarrow F_N = 1 \tag{3.1}$$

The value of $F_N = 1$ is consistent with the value showed in the Reshetov table (Figure 8(b)). Now we can apply the Reshetov method to another mechanism which is kinematically equivalent to the four-bar mechanism. The joints a and c of this mechanism has revolute joints with rotational freedom around the z-axis. The joints b and d were replaced by spherical and cylindrical joints, respectively.

Figure 8 – Self-aligning mechanism equivalent to a four-bar and respective Reshetov Table.



(a) Mechanism

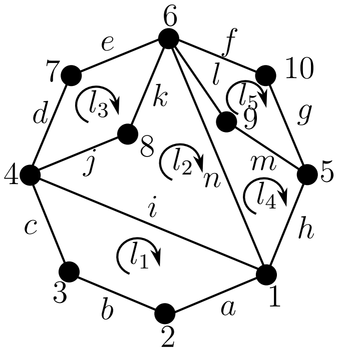
(b) Reshetov Table

The joint b has three rotational degrees of freedom (R_x, R_y, R_z), and the joint d has a translational and a rotational degrees of freedom (R_z, T_z). The freedom of the kinematic pairs were applied to the Reshetov table shown in Figure 9(b). It is possible to note that more degrees of freedom were allowed hence the number of redundant constraints decreased. The number of redundant constraints in this mechanisms is equal to zero, $C_N = 0$. So this mechanism is considered as a self-alingning mechanism, applying it to Equation 2.15:

$$F_N = 6(4 - 4 - 1) + 7 + 0 \Rightarrow F_N = 1 \tag{3.2}$$

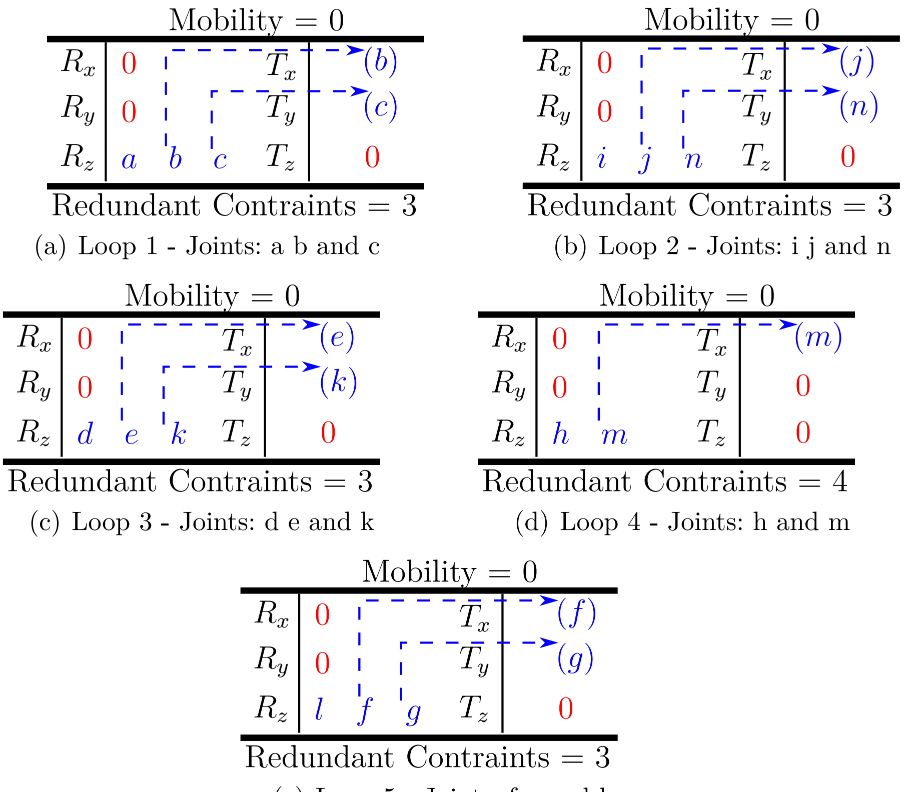
The value of $F_N = 1$ is consistent with the value showed in the Reshetov table (Figure 9(b)). Now, we can apply the Reshetov method to the steering system shown in Figure 4.

Figure 9 – Circuits of the steering system



A circuit can close without redundant constraints if all six freedoms are present (RESHETOV, 1982). The steering system has fourteen revolute joints and five circuits, so each circuit must be analyzed separately. A joint which is present in more than one circuit can be considered in any of this circuits, but cannot be repeated in the analyses of other circuits .

Figure 10 – Reshetov Table related to the steering system



(a) Loop 1 - Joints: a b and c

(b) Loop 2 - Joints: i j and n

(c) Loop 3 - Joints: d e and k

(d) Loop 4 - Joints: h and m

(e) Loop 5 - Joints: f g and l

Analyzing the Reshetov tables related to the steering system, Figure 10, this mechanism has 16 redundant constraints $(3+3+3+4+3=16)$, applying this value to Equation 2.15:

$$F_N = 6(10-14-1)+14+16 \Rightarrow F_N = 0 \tag{3.3}$$

The value of $F_N = 0$ is consistent to the results shown in Reshetov table, Figure 10. But the result calculated is not correct, this mechanism has one degree of freedom, $F_N = 1$ (MANENTI, 2018), while the analysis detected that this mechanism has no mobility. This counter-example can be analyzed by means of screw theory and Davies' method and it is presented in Section 3.2.

3.2 REDUNDANT CONSTRAINT ANALYSIS BY MEANS OF DAVIES' METHOD

In this section the matrices $[\widehat{M}_N]$ and $[\widehat{A}_N]$ are analyzed aiming to determine the number and the modes of redundant constraints present in the mechanism. Matroid theory is then used to enumerate all the self-aligning mechanisms derived from a given mechanism. This approach was developed by Carboni (2015).

The steering system, shown in Figure 4, is the example of this section. The coupling graph, Figure 12(a), represents the interaction among the links of the mechanism. As all the joints are revolute joints, the motion and action screws of each joint are similarly modeled 3.4, the difference are the position of each joint.

The workspace of the steering system is planar, $\lambda = 3$. To respect Equation 2.9, the sum of the number of twists and wrenches to each joint must be equal to three. The screws for any joint are:

$$\$^m_{*t} = \begin{bmatrix} \cancel{0} \\ \cancel{0} \\ 1 \\ y_* \\ -x_* \\ \cancel{0} \end{bmatrix}, \$^a_{*U} = \begin{bmatrix} \cancel{0} \\ \cancel{0} \\ -y_* \\ 1 \\ 0 \\ \cancel{0} \end{bmatrix}, \$^a_{*V} = \begin{bmatrix} \cancel{0} \\ \cancel{0} \\ x_* \\ 0 \\ 1 \\ \cancel{0} \end{bmatrix} \tag{3.4}$$

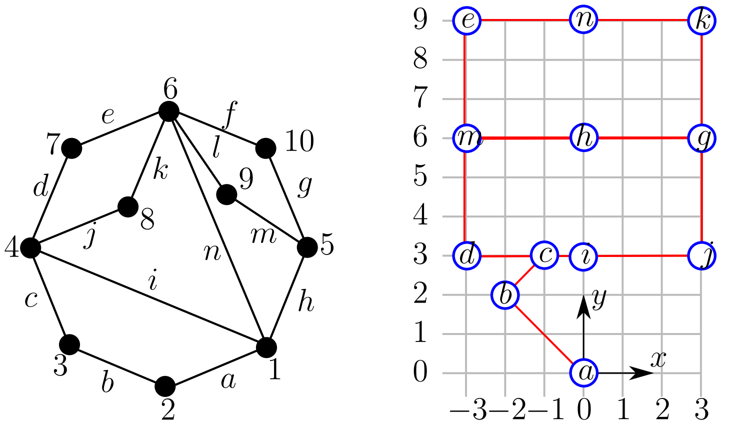
Where the twist $\$^m_{*t}$ is related to the rotational freedom around z-axis of any joint. The wrenches $\$^a_{*U}$ and $\$^a_{*V}$ are related to the force constraints of any joint along the axes x and y. Figure 12(b) illustrates the positioning of the joints for a given pose, the joints f and l have the same position of the joints e and k, respectively.

According to Davies' notation, the terms r, s and w of all twists are equal to zero, this characteristic evidences that the workspace is planar, these terms are then removed. After, modeling the twists of the other joints, it is possible to arrange the unit motion matrix $[\widehat{M}_D]$, Equation 3.5.

$$\left[\widehat{M}_D\right]_{3,14} = \Big[\widehat{\$}^m_{at} \quad \widehat{\$}^m_{bt} \quad \widehat{\$}^m_{ct} \quad \widehat{\$}^m_{dt} \quad \widehat{\$}^m_{et} \quad \widehat{\$}^m_{ft} \quad \widehat{\$}^m_{gt} \dots$$
$$\dots \widehat{\$}^m_{ht} \quad \widehat{\$}^m_{it} \quad \widehat{\$}^m_{jt} \quad \widehat{\$}^m_{kt} \quad \widehat{\$}^m_{lt} \quad \widehat{\$}^m_{mt} \quad \widehat{\$}^m_{nt}\Big] \tag{3.5}$$

The terms R, S and W of all wrenches are also equal to zero, so the unit action matrix $[\widehat{A}_D]$ is given by:

Figure 11 – Coupling graph of the steering system



(a) Couping graph of the steering system

(b) Joint position for a given pose

$$\left[\widehat{A}_D\right]_{3,28} = \Big[\widehat{\$}^a_{aU} \quad \widehat{\$}^a_{aV} \quad \widehat{\$}^a_{bU} \quad \widehat{\$}^a_{bV} \quad \widehat{\$}^a_{cU} \quad \widehat{\$}^a_{cV} \quad \widehat{\$}^a_{dU} \quad \widehat{\$}^a_{dV} \ldots$$
$$\ldots \widehat{\$}^a_{eU} \quad \widehat{\$}^a_{eV} \quad \widehat{\$}^a_{fU} \quad \widehat{\$}^a_{fV} \quad \widehat{\$}^a_{gU} \quad \widehat{\$}^a_{gV} \quad \widehat{\$}^a_{hU} \quad \widehat{\$}^a_{hV} \quad \ldots$$
$$\ldots \widehat{\$}^a_{iU} \quad \widehat{\$}^a_{iV} \quad \widehat{\$}^a_{jU} \quad \widehat{\$}^a_{jV} \quad \widehat{\$}^a_{kU} \quad \widehat{\$}^a_{kV} \quad \widehat{\$}^a_{lU} \quad \widehat{\$}^a_{lV} \ldots$$
$$\ldots \quad \widehat{\$}^a_{mU} \quad \widehat{\$}^a_{mV} \quad \widehat{\$}^a_{nU} \quad \widehat{\$}^a_{nV}\Big] \qquad (3.6)$$

To arrange the matrices $[\widehat{M}_N]$ and $[\widehat{A}_N]$, it is necessary to know the relationship among the mobilities and constraints of the system. Therefore the motion and the action graph are studied. Both graphs are shown in Figure 12. Since all the joints have one degree of freedom t, a rotation around the z-axis, the edges of the coupling graph are not expanded, so the motion graph is equivalent to the coupling graph, Figure 13(a).

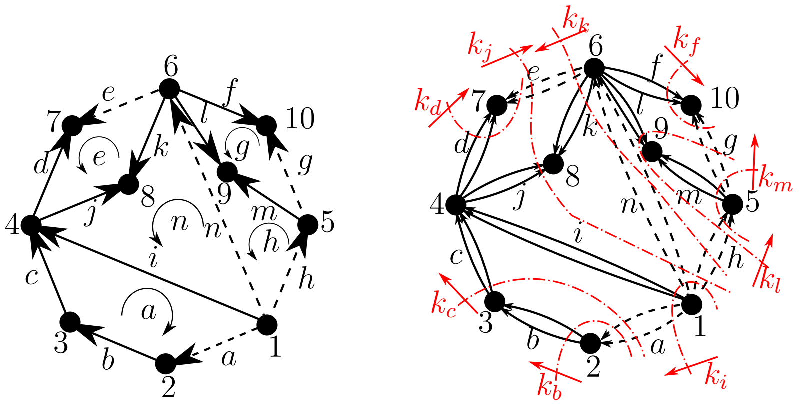
The joints have two constraints, u and v, thus the edges of the coupling graph are replaced by two parallel edges, resulting in the action graph, Figure 13(b). The edges a, g, h and n were arbitrary defined as chords.

According to the Section 2.1, the motion graph is useful to arrange the circuit matrix $[B]_{5,14}$, while the action graph allows arranging the cut-set matrix $[Q]_{9,28}$.

The matrices $[\widehat{M_N}]_{15,14}$ and $[\widehat{A_N}]_{27,28}$ are arranged following Equations 2.12 and 2.13, respectively. Then, both matrices are separately and differently analyzed to evaluate the redundant constraints of the system. The results of both analyzes must indicate the same results.

The presence of each degree of freedom in each independent circuit is analyzed using

Figure 12 – Motion and action graph of the steering system



(a) Motion graph

(b) Action graph

the network unit motion matrix $[M_N]_{\lambda\nu,F}$. The number of missing freedoms is the number of redundant constraints (CARBONI, 2015).

For a given system modeled by the matrix $[M_N]_{\lambda\nu,F}$, each row is related to a specific degree of freedom of a circuit. The linear independence of the rows is analyzed by the rank of the $[M_N]^T_{\lambda\nu,F}$, and the difference between $\lambda\nu$ and $rank([M_N]^T_{\lambda\nu,F})$ indicates the number of missing freedoms (DAVIES, 2006):

$$C_N = \lambda\nu - rank([M_N]^T_{\lambda\nu,F}) \tag{3.7}$$

Applying Equation 3.7 to the steering system of Figure 4 which has five loops ($\nu = 5$) and works in a planar system ($\lambda = 3$) , it gives $C_N = (3)(5) - 13 \Rightarrow C_N = 2$, meaning the system has two redundant constraints. This result is useful to correctly calculate the mobility, according to the Modified Grübler-Kutzbach Criterion, Equation 2.15:

$$F_N = 3(10 - 14 - 1) + 14 + 2 \Rightarrow F_N = 1 \tag{3.8}$$

Knowing the cardinality of C_N is interesting to know which constraints are redundant, or which deegres of freedom are missing. The matrix $[M_N]^T_{\lambda\nu,F}$ is arranged in reduced row echelon form $(rref)$ to verify the presence of redundant constraints in the mechanisms (CARBONI, 2015).

The columns of the new matrix $(rref[M_N]^T_{\lambda\nu,F})$ are then analyzed. If the column is pivoted, it indicates that the freedom is present in the loop, while if the column is not-pivoted, it indicates that the freedom must be added in a coupling of the loop (CARBONI, 2015).

The matrix $(rref[M_N]^T_{15,14})$ of the steering system is shown in Equation 3.9. Each group of three columns (t, u, v) represents a loop of the coupling graph, shown in Figure 13(a). The v columns of the loops h and n, marked by the red boxes, are not pivoted, meaning that the freedoms of translation v along the y-axis are missing in these loops.

$$\text{rref}([\widehat{M_N}]^T_{15,14}) = \begin{array}{c}
\begin{array}{ccccccccccccccc} t & u & v & t & u & v & t & u & v & t & u & v & t & u & v \end{array} \\
\left[\begin{array}{ccc|ccc|ccc|ccc|ccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1{,}5 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0{,}5 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -7{,}5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -0{,}5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\end{array} \tag{3.9}$$

Another way to evaluate the redundant constraints is using the network unit action matrix $[A_N]_{\lambda k,C}$. In matrix $[A_N]_{\lambda k,C}$ each column represents a single constraint belonging to a coupling. The linear dependence among the constraints can be analyzed by studying the properties of matrix $[A_N]_{\lambda k,C}$ (CARBONI, 2015).

If a mechanism is overconstrained, the rank of $[A_N]_{\lambda k,C}$ will be less than C. So, the cardinality of redundant constraints can be defined by the following Equation 3.10 (CARBONI, 2015):

$$C_N = C - rank([A_N]_{\lambda k,C}) \tag{3.10}$$

Applying Equation 3.10 to the steering system, statically modelled by the matrix $[A_N]_{27,28}$, the number of redundant constraints is $C_N = 28 - 26 \Rightarrow C_N = 2$. This value is the same value of C_N found using the matrix $[M_N]_{15,14}$ and Equation 3.7.

$$\mathrm{rref}([A_N]_{27,28}) = \begin{array}{cccccccc} \widehat{\$}^a_{aU} & \widehat{\$}^a_{aV} & \widehat{\$}^a_{bU} & \cdots & \widehat{\$}^a_{mU} & \widehat{\$}^a_{mV} & \widehat{\$}^a_{nU} & \widehat{\$}^a_{nV} \\ \hline 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0.5 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -2 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \end{array} \tag{3.11}$$

It is possible to identify the modes of the redundant constraints arranging the matrix $[\widehat{A_N}]_{\lambda k,C}$ in the reduced row echelon form, as shown in Equation 3.11. The columns which represent redundant constraints are not pivoted.

The constraints of the wrenches $\widehat{\$}^a_{mV}$ and $\widehat{\$}^a_{nV}$ are redundant constraints in the steering system. Both redundant constraints are forces along the y-axis. This result is consistent to the result found in Equation 3.9, where the missing freedoms are translations along the y-axis.

By removing the constraints $\widehat{\$}^a_{mV}$ and $\widehat{\$}^a_{nV}$ of the mechanism, a new self-aligning mechanism is generated. This new mechanism is kinematically equivalent to the original. However, it is not the unique self-aligning mechanism kinematically equivalent to the original. To enumerate all the self-aligning solutions of an overconstrained mechanism, Matroid theory can be employed. The next subsection approaches an enumeration method based on Matroid theory.

3.2.1 Enumeration of non-isomorphic self-aligning mechanisms derived from a seed mechanism

This subsection reviewes the method introduced by Carboni (2015) and applies it to the four-bar mechanism. After introducing the concepts of a matroid, it is possible to use the network unit action matrix $[\widehat{A}_N]$ to define the matroid $\mathcal{M}_{AN}$, automatically to eliminate the redundant constraints of the system. It interesting to use the dual matroids because normally the dual ranks are smaller than matroid ranks, which facilitates the understanding of the process, it is true when the matrix $[\widehat{A}_N]$ is analyzed.

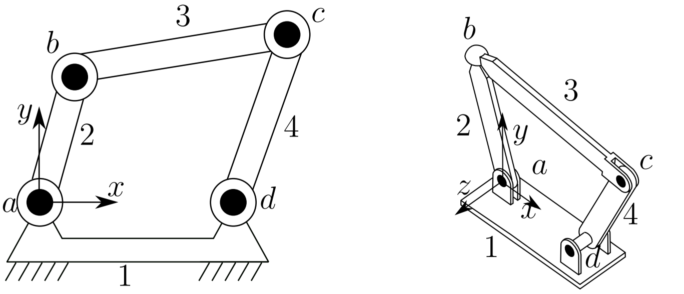
Given a network unit action matrix $[\widehat{A}_N]$, the matrix columns are ordered according to the wrenches. The Equation 3.12 shows the sequence of wrenches for the four-bar mechanism, shown in Figure 14(a). Note the wrenches T relative to the constraints of moment around the z-axis, are not added in the matrix. It occurs because the wrenches T are the freedoms of the joints. So each joint has five constraints hence five wrenches.

$$\begin{aligned}[\widehat{A}_N]_{6*3,20} = \Big[\widehat{\$}^a_{aR} \quad \widehat{\$}^a_{aS} \quad \widehat{\$}^a_{aU} \quad \widehat{\$}^a_{aV} \quad \widehat{\$}^a_{aW} \quad \ldots \\ \ldots \quad \widehat{\$}^a_{bR} \quad \widehat{\$}^a_{bS} \quad \widehat{\$}^a_{bU} \quad \widehat{\$}^a_{bV} \quad \widehat{\$}^a_{bW} \quad \ldots \\ \ldots \quad \widehat{\$}^a_{cR} \quad \widehat{\$}^a_{cS} \quad \widehat{\$}^a_{cU} \quad \widehat{\$}^a_{cV} \quad \widehat{\$}^a_{cW} \quad \ldots \\ \ldots \quad \widehat{\$}^a_{dR} \quad \widehat{\$}^a_{dS} \quad \widehat{\$}^a_{dU} \quad \widehat{\$}^a_{dV} \quad \widehat{\$}^a_{dW}\Big]\end{aligned} \tag{3.12}$$

Now, a dual linear matroid $\mathcal{M}^*_{AN}$ is created on the real field $I\!R$ defined over matrix $[\widehat{A}_N]$. For the four-bar mechanism, $\mathcal{M}^*_{AN}$ has 112 cobases. In other words, the four-bar mechanism generates 112 different self-aligning mechanisms by removing sets of redundant constraints. The rank of matrix $[\widehat{A}_N]$ is 17, and the gross degree of constraint is equal to 20, $C = 20$, so applying this values in Equation 3.10, the number of redundant constraints is equal to three. Then, the cobases are sets of three redundant constraints, which are shown in Equation 3.13.

The self-aligning mechanism shown in Figure 14(b) is kinematically identical to the four-bar mechanism. But in this mechanism a set of three redundant constraints, $\{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{dW}\}$, were replaced by freedoms. This cobasis is shown in red in Equation 3.13. The degrees of freedom, which were replaced, are two rotational freedom around the axes x and y in joint b, and a translational freedom along the z-axis in joint d . Now, the joints (b) and (d) are spherical and cylindrical joints, respectively, and the joints (a) and (c) were not modified and are still as revolute joints.

Figure 13 – Four-bar mechanism and a self-aligning four-bar mechanism.



(a) Four-bar mechanism

(b) A self-aligning four-bar mechanism

As shown in this subsection, given a seed mechanism with the $[\widehat{A}_N]$ matrix arranged, matroid theory is employed to enumerate all the self-aligning derived mechanisms. However, the amount of solutions grows exponentially when the complexity of the seed mechanism increases, and selecting the better self-aligning mechanism is a hard task for the designer. So, Carboni (2015) proposed to use Greedy algorithm to select an optimal self-aligning mechanism among all possible solutions.

$$
\begin{aligned}
\mathcal{B}^* = \{&\{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aS}, \widehat{\$}^a_{aW}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aS}, \widehat{\$}^a_{bR}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aS}, \widehat{\$}^a_{bS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aS}, \widehat{\$}^a_{bW}\},\\
&\{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aS}, \widehat{\$}^a_{cR}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aS}, \widehat{\$}^a_{cS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aS}, \widehat{\$}^a_{cW}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aS}, \widehat{\$}^a_{dR}\},\\
&\{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aS}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aS}, \widehat{\$}^a_{dW}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aW}, \widehat{\$}^a_{bS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aW}, \widehat{\$}^a_{cS}\},\\
&\{\widehat{\$}^a_{aR}, \widehat{\$}^a_{aW}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{cS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{dS}\},\\
&\{\widehat{\$}^a_{aR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{bW}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{cR}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{cS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{cW}\},\\
&\{\widehat{\$}^a_{aR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{dW}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{bW}, \widehat{\$}^a_{cS}\},\\
&\{\widehat{\$}^a_{aR}, \widehat{\$}^a_{bW}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{cS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{cW}\},\\
&\{\widehat{\$}^a_{aR}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dW}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{cW}, \widehat{\$}^a_{dS}\},\\
&\{\widehat{\$}^a_{aR}, \widehat{\$}^a_{dR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{aR}, \widehat{\$}^a_{dS}, \widehat{\$}^a_{dW}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{aW}, \widehat{\$}^a_{bR}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{aW}, \widehat{\$}^a_{cR}\},\\
&\{\widehat{\$}^a_{aS}, \widehat{\$}^a_{aW}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{bW}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{cR}\},\\
&\{\widehat{\$}^a_{aS}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{cS}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{cW}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{dS}\},\\
&\{\widehat{\$}^a_{aS}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{dW}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{cR}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{bW}, \widehat{\$}^a_{cR}\},\\
&\{\widehat{\$}^a_{aS}, \widehat{\$}^a_{bW}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{cS}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{cW}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{dR}\},\\
&\{\widehat{\$}^a_{aS}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{dW}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{cW}, \widehat{\$}^a_{dR}\},\\
&\{\widehat{\$}^a_{aS}, \widehat{\$}^a_{dR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{aS}, \widehat{\$}^a_{dR}, \widehat{\$}^a_{dW}\}, \{\widehat{\$}^a_{aW}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}\}, \{\widehat{\$}^a_{aW}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{cS}\},\\
&\{\widehat{\$}^a_{aW}, \widehat{\$}^a_{bR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{aW}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{cR}\}, \{\widehat{\$}^a_{aW}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{aW}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{cS}\},\\
&\{\widehat{\$}^a_{aW}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{aW}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{aW}, \widehat{\$}^a_{dR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{bW}\},\\
&\{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{cR}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{cS}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{cW}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{dR}\},\\
&\{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{dW}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bW}, \widehat{\$}^a_{cS}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bW}, \widehat{\$}^a_{dS}\},\\
&\{\widehat{\$}^a_{bR}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{cS}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{cW}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dR}\},\\
&\{\widehat{\$}^a_{bR}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dW}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{cW}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{bR}, \widehat{\$}^a_{dR}, \widehat{\$}^a_{dS}\},\\
&\{\widehat{\$}^a_{bR}, \widehat{\$}^a_{dS}, \widehat{\$}^a_{dW}\}, \{\widehat{\$}^a_{bS}, \widehat{\$}^a_{bW}, \widehat{\$}^a_{cR}\}, \{\widehat{\$}^a_{bS}, \widehat{\$}^a_{bW}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{bS}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{cS}\},\\
&\{\widehat{\$}^a_{bS}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{cW}\}, \{\widehat{\$}^a_{bS}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{bS}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{bS}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{dW}\},\\
&\{\widehat{\$}^a_{bS}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{bS}, \widehat{\$}^a_{cW}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{bS}, \widehat{\$}^a_{dR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{bS}, \widehat{\$}^a_{dR}, \widehat{\$}^a_{dW}\},\\
&\{\widehat{\$}^a_{bW}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{cS}\}, \{\widehat{\$}^a_{bW}, \widehat{\$}^a_{cR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{bW}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{bW}, \widehat{\$}^a_{dR}, \widehat{\$}^a_{dS}\},\\
&\{\widehat{\$}^a_{cR}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{cW}\}, \{\widehat{\$}^a_{cR}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dR}\}, \{\widehat{\$}^a_{cR}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{cR}, \widehat{\$}^a_{cS}, \widehat{\$}^a_{dW}\},\\
&\{\widehat{\$}^a_{cR}, \widehat{\$}^a_{cW}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{cR}, \widehat{\$}^a_{dR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{cR}, \widehat{\$}^a_{dS}, \widehat{\$}^a_{dW}\}, \{\widehat{\$}^a_{cS}, \widehat{\$}^a_{cW}, \widehat{\$}^a_{dR}\},\\
&\{\widehat{\$}^a_{cS}, \widehat{\$}^a_{dR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{cS}, \widehat{\$}^a_{dR}, \widehat{\$}^a_{dW}\}, \{\widehat{\$}^a_{cW}, \widehat{\$}^a_{dR}, \widehat{\$}^a_{dS}\}, \{\widehat{\$}^a_{dR}, \widehat{\$}^a_{dS}, \widehat{\$}^a_{dW}\}\}
\end{aligned}
\tag{3.13}
$$

3.2.2 Greedy algorithm

Given a mechanism with C constraints imposed, a weight w_i is attributed to each constraint c_i. The set of weights is chosen according to criteria of the mechanism specifications. Each column of $[\widehat{A}_N]$ corresponds to a specific constraint imposed by the couplings, so a set of weights is atributted to the collumns of $[\widehat{A}_N]$ and the greedy algorithm is used to determine the maximum weight independent set in a matroid. In this manner, the set of variables corresponding to this set is the self-aligning mechanism which best compromises with the mechanism specifications.

Returning to the four-bar seed mechanism, Table 1 shows an example of weight set to reach the self-aligning mechanism derived from the four-bar mechanism.

Table 1 – Weight constraints for the four-bar mechanism.

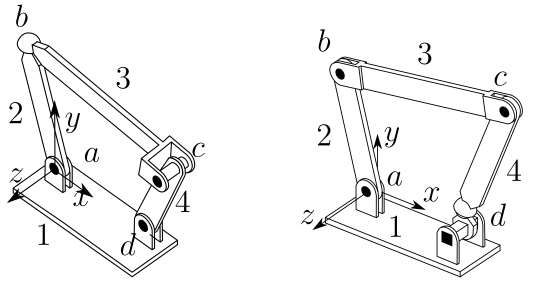
$\mathbf{c_i}$	$\mathbf{w_i}$	$\mathbf{c_i}$	$\mathbf{w_i}$	$\mathbf{c_i}$	$\mathbf{w_i}$	$\mathbf{c_i}$	$\mathbf{w_i}$
$\widehat{\$}^a_{aR}$	5	$\widehat{\$}^a_{bR}$	2	$\widehat{\$}^a_{cR}$	5	$\widehat{\$}^a_{dR}$	5
$\widehat{\$}^a_{aS}$	5	$\widehat{\$}^a_{bS}$	2	$\widehat{\$}^a_{cS}$	5	$\widehat{\$}^a_{dS}$	5
$\widehat{\$}^a_{aU}$	5	$\widehat{\$}^a_{bU}$	3	$\widehat{\$}^a_{cU}$	5	$\widehat{\$}^a_{dU}$	5
$\widehat{\$}^a_{aV}$	5	$\widehat{\$}^a_{bV}$	3	$\widehat{\$}^a_{cV}$	5	$\widehat{\$}^a_{dV}$	5
$\widehat{\$}^a_{aW}$	5	$\widehat{\$}^a_{bW}$	3	$\widehat{\$}^a_{cW}$	2	$\widehat{\$}^a_{dW}$	2

The result is a self-aligning mechanism shown in Figure 14(b). Note the three constraints with lesser weight are the constraints which are replaced by three freedoms. They corresponds to the two moments around the x and y axes in the joint b that were replaced by two rotative freedoms around the same axes. A force along the z-axis that was also replaced by a linear displacement freedom in the joint d. The joints a and c have the same constraints of the seed mechanism.

Other examples applying Matroid theory to enumerate the self-aligning mechanisms and greedy algorithm to select a solution, can be found in Carboni (2015), Barreto (2018), and Carboni (2017).Though the weight setting is according to the design requirements of the mechanism, establishing the values of constraints' weights is a hard task for the designer. Moreover, it requires knowledge in screw theory and Matroid theory.

By analysing the other cobases we can note that the cobasis $\{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{cW}\}$ has the same maximum weight independent set. However the greedy algorithm returned other basis as result and this basis would not be checked by the designer. The mechanism derived from this basis $\{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{cW}\}$ is shown in Figure 15(a). This mechanism has a cylindrical joint in the joint c, instead of joint d as the greedy algorithm returned in the basis $\{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{dW}\}$.

Figure 14 – Others examples of self-aligning mechanisms derived from a four-bar.



(a) $\{\widehat{\$}^a_{bR}, \widehat{\$}^a_{bS}, \widehat{\$}^a_{cW}\}$

(b) $\{\widehat{\$}^a_{dR}, \widehat{\$}^a_{dS}, \widehat{\$}^a_{dW}\}$

The greedy algorithm returns just a maximum independent set by a set of weights, other bases which have the same maximum weight independent set are not shown in the result. So the designer must test different sets of weights until obtaining a feasible self-aligning mechanism.

An example of an unfeasible self-aligning mechanism derived from a four-bar is shown in Figure 15(b). This mechanism is a didactic representation of the cobasis $\{\widehat{\$}^a_{dR}, \widehat{\$}^a_{dS}, \widehat{\$}^a_{dW}\}$, which was enumerated by the matroid $\mathcal{M}_{AN}$. This mechanism is self-aligning but the design and manufacturing are more complicated than the other mechanisms presented in Figures 14(b) and 15(a). This is due to the fact that the joint d has three constraints replaced by three degrees of freedom, resulting a new joint considered as a higher pair.

Higher pairs are those pairs whose contact between the parts are a point, a curve or a line. This kind of pairs decreases the rigidity of a mechanism and they are normally undesirable. Lower pairs are those whose elements touch one another over a substantial region of a surface (HUNT, 1978). This work follows the classification established by Hunt (1978), where the pairs considered as lower pairs are the spherical, planar, cylindrical, revolute, prismatic and screw pairs.

3.3 FINAL CONSIDERATIONS OF THE CHAPTER

In this chapter, an bibliographic review about redundant constraint analysis were presented. Firstly, the reshetov method was reviewed and a counter-example was discussed. Then, a review about redundant constraint analysis by means of Davies' method as reviewed and applied to a steering system. Then, a method to enumerate all self-aligning mechanisms derived from a seed mechanism was reviewed and applied to a four-bar mechanism by means of matroid theory.

The cobases of a four-bar mechanism were enumerated and it was explained that each cobasis is related to a self-aligning mechanism. Finally, the greedy algorithm was presented to select a self-aligning mechanism derived from the four-bar mechanism.

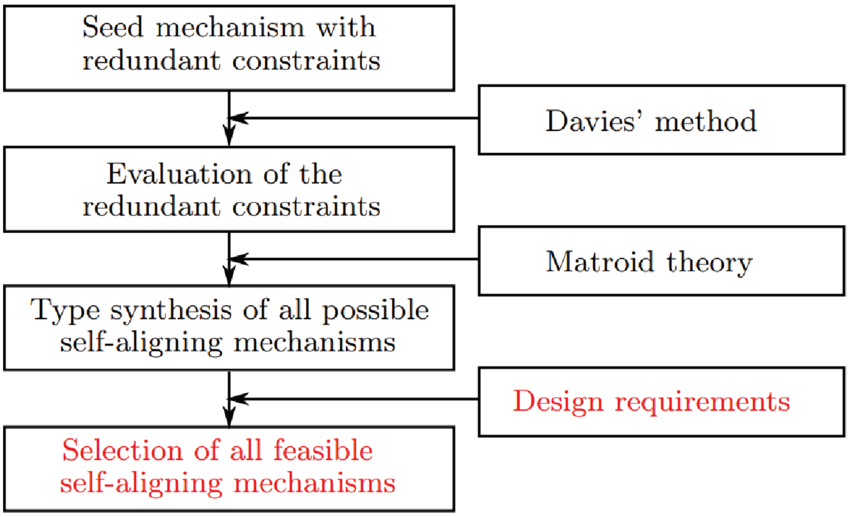
The next chapter presents a new method to select sets of cobases according to design requirements. In the new selection method, the design requirements are not transformed in constraint weights and the result is a set of feasible self-aligning mechanisms.

4 SELECTION METHOD OF SELF-ALIGNING MECHANISMS

This chapter presents a new selection method of self-aligning mechanisms according to design requirements. This method uses the design requirements and the self-aligning mechanisms (cobases) enumerated from the dual matroid $\mathcal{M}^*_{AN}$ as inputs. The dual matroid $\mathcal{M}^*_{AN}$ is created from the network unit action matrix $[A_N]_{\lambda k,C}$. Figure 15 shows the flowchart which represents the type synthesis procedure for self-aligning mechanisms. The first step is to determine an appropriate seed mechanism with redundant constraints. By applying Davies' method those redundant constraints are evaluated. Then, Matroid theory is used to enumerate the cobases which represent all possible self-aligning mechanisms.

The contibuition of this work is the selection method, highlighted in red in the flowchart of Figure 15. It transforms the engineering design requirements in selection criteria which are used to evaluate all cobases enumerated by the method presented by Carboni (2015).

Figure 15 – Flowchart of the type synthesis of self-aligning mechanisms.


Seed mechanism with redundant constraints
Davies' method
Evaluation of the redundant constraints
Matroid theory
Type synthesis of all possible self-aligning mechanisms
Design requirements
Selection of all feasible self-aligning mechanisms

This chapter makes clear the procedure of the proposed method. Firstly in Section 4.1, the cobases are organized for the application of selection criteria. Then, Section 4.2 explains how the design requirements are transformed into selection criteria and these criteria are then applied in the method evaluating the cobases. Finally, in Section 4.3 the set of self-aligning mechanisms which attend the design requirements is presented.

4.1 COBASES BINARY MATRIX

Given a dual matroid $\mathcal{M}^*_{AN}$, the cobases are arranged in sets of columns which are removed from the $[\widehat{A}_N]_{\lambda k,C}$ matrix of the seed mechanism. The remaining columns create linearly independent sets. Thus, each cobasis creates a new matrix $[\widehat{A}_N]'_{\lambda k,C'}$, the ranks of the new matrices are equal to the number C' of columns. C' is named in this works as the *exact gross degree of constraint* and it is defined by Equation 4.1:

$$C' = C - C_N \tag{4.1}$$

C' is the difference between the gross degree of constraint C and the number C_N of redundant constraints, both variables come from the seed mechanism. The dual matroid $\mathcal{M}^*_{AN}$ has μ cobases, therefore μ matrices $[\widehat{A}_N]'_{\lambda k,C'}$ must be evaluated. The main difference between the seed mechanism and the new self-aligning mechanisms is the replacement of some constraints by freedoms, in this way the Cobases Binary Matrix $[N]_{\mu,C}$ is arranged, where the cobasis i of $\mathcal{B}^*$ is arranged in the line i of matrix $[N]_{\mu,C}$.

The matrix $[N]_{\mu,C}$ is arranged in a binary form following the organization of matrix $[\widehat{A}_N]_{\lambda k,C}$, and the elements $n_{i,j}$ are defined by:

$$n_{(i,j)} = \begin{cases} 1 & \text{if the constraint } j \text{ is in the cobasis } i \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

In other words, if the element $n(i,j)$ is equal to one, it means the self-aligning mechanism i has the constraint j removed from the seed mechanism. While if the element $n(i,j)$ is equal to zero, the self-aligning mechanism i maintains the constraint j.

By arranging the matrix $[N]_{\mu,C}$, it is possible to identify which constraints were replaced by freedoms, considering the modes of constraints and joints. The sum of the elements of each line is equal to C_N because the exact number of redundant constraints were removed in each new self-aligning mechanism.

Considering the four-bar mechanism of Figure 14(a) as an example of seed mechanism with $C_N = 3$ redundant constraints. The dual matroid $\mathcal{M}^*_{AN}$ was created from the matrix $[\widehat{A}_N]_{18,20}$. This dual matroid was applied to the Equation 4.2 to arrange the matrix $[N]_{112,20}$. The matrix $[N]_{112,20}$ has twenty columns because the seed mechanism has four joints with five constraints. Equation 4.3 shows the three first lines of $[N]_{112,20}$. Equation 3.12 shows the sequence as the constraints were organized in Equation 4.3. Some elements of matrix

$[N]_{112,20}$ were recovered by suspension points, the values of these elements are equal to zero.

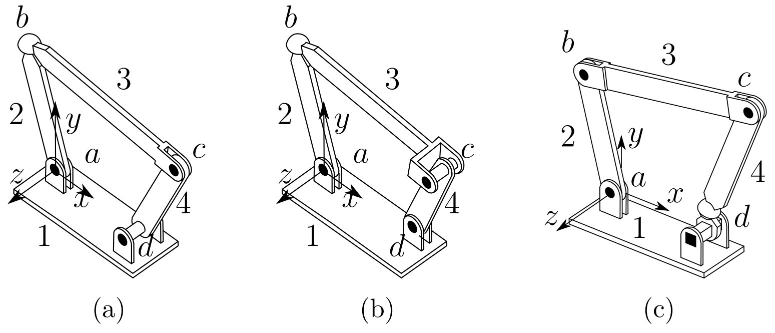
$$[N]_{112,20} = \begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ \vdots \end{array} \begin{array}{cccccccccc} \cdots & \widehat{\$}^a_{bR} & \widehat{\$}^a_{bS} & \cdots & \widehat{\$}^a_{cW} & \widehat{\$}^a_{dR} & \widehat{\$}^a_{dS} & \widehat{\$}^a_{dU} & \widehat{\$}^a_{dV} & \widehat{\$}^a_{dW} \\ \end{array} \begin{bmatrix} \cdots & 1 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 & 1 \\ \cdots & 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 & 0 \\ \cdots & 0 & 0 & \cdots & 0 & 1 & 1 & 0 & 0 & 1 \\ \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{4.3}$$

Analysing the line l_1, the constraints related to the wrenches $\widehat{\$}^a_{bR}$, $\widehat{\$}^a_{bS}$ and $\widehat{\$}^a_{dW}$ were removed from the matrix $[\widehat{A}_N]_{18,20}$, hence the joints b and d were transformed in spherical and cylindrical joints, respectively. The wrenches of the new matrix $[\widehat{A}_N]'_{18,17}$ are shown in Equation 4.4, and Figure 17(a) shows a representation of this self-aligning mechanism

$$\begin{aligned}[\widehat{A}_N]'_{18,17} = \Big[& \widehat{\$}^a_{aR} \quad \widehat{\$}^a_{aS} \quad \widehat{\$}^a_{aU} \quad \widehat{\$}^a_{aV} \quad \widehat{\$}^a_{aW} \quad \widehat{\$}^a_{bU} \quad \widehat{\$}^a_{bV} \quad \widehat{\$}^a_{bW} \quad \cdots \\ & \cdots \quad \widehat{\$}^a_{cR} \quad \widehat{\$}^a_{cS} \quad \widehat{\$}^a_{cU} \quad \widehat{\$}^a_{cV} \quad \widehat{\$}^a_{cW} \quad \widehat{\$}^a_{dR} \quad \widehat{\$}^a_{dS} \quad \widehat{\$}^a_{dU} \quad \widehat{\$}^a_{dV} \Big]\end{aligned} \tag{4.4}$$

In the line l_2, the constraints $\widehat{\$}^a_{bR}$, $\widehat{\$}^a_{bS}$ and $\widehat{\$}^a_{cW}$ were replaced by freedoms, so the joints b and c become spherical and cylindrical joints, respectively, as shown in Figure 17(b). In line l_3 the wrenches $\widehat{\$}^a_{dR}$, $\widehat{\$}^a_{dS}$ and $\widehat{\$}^a_{dW}$ were removed from matrix $[\widehat{A}_N]_{18,20}$, in this way, just the joint d was modified. The new joint d has four deegres of freedom, three rotations around the x, y and z axes and a translation along the z-axis. Figure 17(c) shows the self-aligning mechanism related to the line l_3.

Figure 16 – Self-aligning mechanisms derived from four-bar mechanism.



(a) (b) (c)

The Cobases Binary Matrix $[N]_{\mu,C}$ is useful to organize the cobases in matrix formulation. The next section discuss how the design requirements are converted into selection criteria. Later the selection criteria are employed in the matrix $[N]_{\mu,C}$ to evaluate all the self-aligning mechanisms.

4.2 SELECTION CRITERIA

The conversion of design requirements into selection criteria is the most important part of the present work. The designer must be careful to determine which requirements are relevant for the Type synthesis. The selection criterion must truly represent the design requirement, otherwise improper self-aligning mechanisms will be selected, or feasible self-aligning mechanisms will be discarded.

The presented method allows the use of a wide range of design requirements, therefore specific applications will demand specific design requirements. This section shows some examples of design requirements and the conversion in selection criteria.

Given a design requirement, it is possible to convert it into a α selection criterion. There is a subset $K_\alpha \in \mathcal{B}^*$ of self-aligning mechanisms which satisfy the criterion α. Remembering that $\mathcal{B}^*$ is the cobases set of the dual matroid $\mathcal{M}^*_{AN}$. So any subset K_α can be defined as:

$$\{\forall i = 1, 2, \ldots, \mu \quad | B^*_i \in K_\alpha \Leftrightarrow n(i, j) = \beta\} \tag{4.5}$$

where μ is the total number of independent bases, B^*_i is any cobasis from $\mathcal{B}^*$. K_α is any subset created from any criterion used in the method. $n(i, j)$ is the element of line i and column j from the Cobases Binary Matrix. β is a binary element defined according to the criterion, if $\beta = 1$ that respective redundant constraint was eliminated from the seed mechanism, if $\beta = 0$ that constraint was not altered. Instead of binary functions $f(\beta)$, other functions can be used in the criteria, for example, inequations or boolean algebra.

In this work, the design requirements are classified in two categories:

1. Joint requirements;

2. System requirements.

The joint requirement includes the design requirements which are related to specific joints. And the system requirement includes the requirements which are related to the system, consequently, all the joints must agree with them.

So, given a joint requirement α_1 for the joint a, the constraints related to this joint must be identified in the matrix $[N]_{\mu,C}$. Then the design requirement is converted into a function $n(i, j) = \beta$. A common example of a requirement is that joint a cannot be modified because it is a revolute actuated joint. Actuated joints has normally one degree of freedom, so no freedom can be added. Considering the five constraints related to the joint a are arranged in the first five columns of the matrix $[N]_{\mu,C}$, the selection criterion can defined as follow:

$$\{\forall i = 1, 2, \ldots, \mu \quad |B_i^* \in K_{\alpha 1} \Leftrightarrow n(i,1) = n(i,2) = \\ n(i,3) = n(i,4) = n(i,5) = 0\}. \quad (4.6)$$

Equation 4.6 states all the cobases are evaluated by the lines of matrix $[N]_{\mu,C}$, and a cobasis belongs to the set $K_{\alpha 1}$ if and only if the first five elements of the line are equal to zero. The other constraints are not evaluated by this criterion.

Now a system requirement α_2 is exemplified. The constraints of all joints must be indentified and evaluated in the matrix $[N]_{\mu,C}$. An example of a system requirement is: at most two constraints cannot be removed from the joints of a four-bar mechanism, otherwise, the mechanism rigidity may be affected. In this case, all the columns are evaluated and boolean algebra is employed. The system criterion can be written as:

$$\{\forall i = 1, 2, \ldots, \mu \quad |B_i^* \in K_{\alpha 2} \Leftrightarrow n(i,1) + n(i,2) + \\ n(i,3) + n(i,4) + n(i,5) \leq 2 \quad \text{AND} \\ n(i,6) + n(i,7) + n(i,8) + n(i,9) + n(i,10) \leq 2 \quad \text{AND} \\ n(i,11) + n(i,12) + n(i,13) + n(i,14) + n(i,15) \leq 2 \quad \text{AND} \\ n(i,16) + n(i,17) + n(i,18) + n(i,19) + n(i,20) \leq 2\} \quad (4.7)$$

Equation 4.7 states all the cobases are evaluated by the lines of matrix $[N]_{\mu,C}$, and a cobasis belongs to the set $K_{\alpha 2}$ if and only if the sums of the elements of each joint, respective to the line evaluated, is less or equal to two. All joints were evaluated by this criterion.

Other examples of joint and system requirements are found in Section 5. Generally, more than one selection criterion is applied to the seed mechanism, then more than a set K is created. The next section explains how the sets K are used to create the set of feasible self-aligning mechanisms.

4.3 SET OF FEASIBLE SELF-ALIGNING MECHANISMS

Each criterion applied to the seed mechanism creates a subset K of self-aligning mechanisms. In this case, a group of criteria can be applied to the seed mechanism, creating a set of subsets K. The intersection between all the subsets is another subset K_F, named as *final subset.* K_F corresponds to the set of feasible self-aligning mechanisms:

$$K_F = K_1 \cap K_2 \cap \ldots \cap K_\alpha \tag{4.8}$$

If K_F is an empty set, it means the seed mechanism does not have any self-aligning mechanism that satisfies all the design requirements. In this case, the designer must re-evaluate the design requirements or find the set of self-aligning mechanisms which satisfy the highest quantity of requirements. Another possibility is to evaluate the importance of each requirement, selecting the set of self-aligning mechanisms which satisfies the most relevant design requirements.

The joint and system criteria of Equations 4.6 and 4.7, respectively were applied to the four-bar mechanism, so two subsets were created, $K_{\alpha 1}$ and $K_{\alpha 2}$. The subset $K_{\alpha 1}$ has 108 cobases which comply with the joint requirement. The subset $K_{\alpha 2}$ has 45 cobases which comply with the system requirement. Both subsets were intersected ,$K_F = K_{\alpha 1} \cap K_{\alpha 2}$, the final subset has 42 cobases. These 42 cobases represents 42 self-aligning mechanisms derived from the four-bar which comply with the design requirements from Equations 4.6 and 4.7.

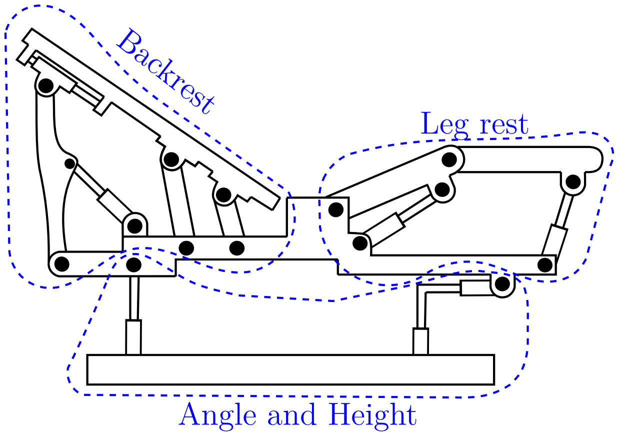
5 CASE STUDIES

The method proposed in the Chapter 4 will be applied in some seed mechanisms to design self-aligning mechanisms. The application focus is directed to hospital beds, where the backrest and leg rest mechanisms of two hospital beds are analyzed and applied to the selection method. The first hospital bed is the Eleganza 3XC (LINET BRASIL, 2019), produced by Linet company, Section 5.1. The second is the hospital bed designed by the researchers of the Laboratory of Applied Robotics (Federal University of Santa Catarina), Section 5.2. The aim of this chapter is to explore the proposed method.

5.1 LINET ELEGANZA 3XC HOSPITAL BED

The mechanisms herein analyzed are employed in the hospital bed Eleganza Smart, developed by Linet company. This particular bed was chosen as a study case due to the availability of this product in the University Hospital of the Federal University Santa Catarina. After analyzing the mechanisms, a structural representation was created, as shown in Figure 17.

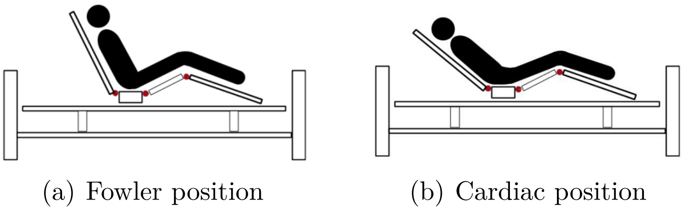
Figure 17 – Structural representation of Eleganza Smart bed.



The mechanism is divided into three independent sections: the backrest adjustment mechanism, the leg rest adjustment mechanism and the height and angle adjustment mechanism. The backrest adjustment mechanism alters the angle between the backrest and the bed structure. The leg rest adjustment mechanism alters the angle of the upper leg rest and adjusting both the angle and position of the lower leg rest. Finally, the angle and height adjustment alters the height of the bed and enables adjustments in angles.

These mechanisms are relevant for the patient as they enable several positions that are important for the rehabilitation process. For example, adjusting the backrest angle as well as elevating the knee enables the Fowler and the cardiac positions (MALETZ, 2017), Figure 18.

Figure 18 – Positions of the backrest adjustment mechanism.



(a) Fowler position

(b) Cardiac position

Adapted from: Maletz (2017)

The backrest adjustment and the leg rest adjustment mechanisms are seed mechanisms analyzed employing Davies' method. The redundant constraints are then evaluated by means of Matroid theory, where all the cobases are listed. Hence all the self-aligning mechanisms derived from the seed mechanisms are enumerated. After, the selection method is applied to define a set of self-aligning mechanisms which satisfies the established design requirements.

5.1.1 Case I : Backrest adjustment mechanism

A structural representation of the backrest adjustment mechanism is shown in Figure 19. The mechanism has ten joints and eight links. The type, the position vector and the wrenches of each joint are shown in Table 2. This system has fifty constraints.

The aim of a redundant constraint analysis is to evaluate the constraints of a given system, the joints positions must be determined without change the mechanism mobility, *i.e.* singularities must be discarded to the analyzed pose.

The next step of the method is to generate the $[\widehat{A}_D]_{6,50}$. Equation 5.1 shows the ordering of the wrenches in this matrix. The modeled wrenches are shown in Appendix B.

Figure 19 – Backrest adjustment mechanism from Eleganza

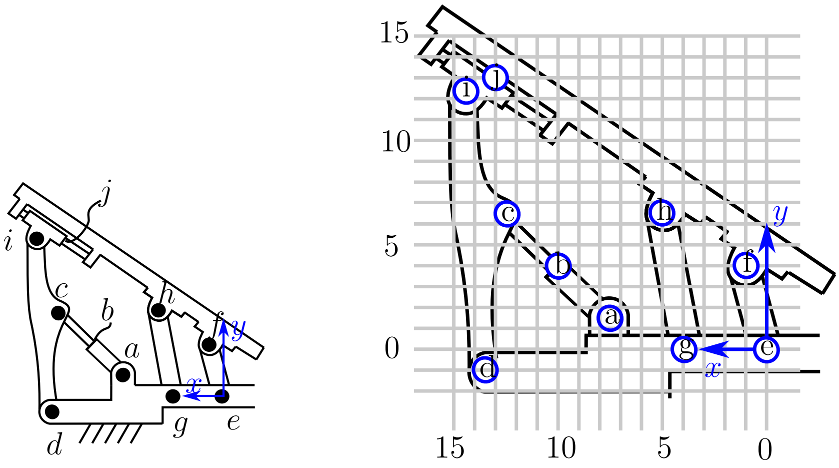


Table 2 – Type of the joints and respective wrenches of the backrest adjustment mechanism

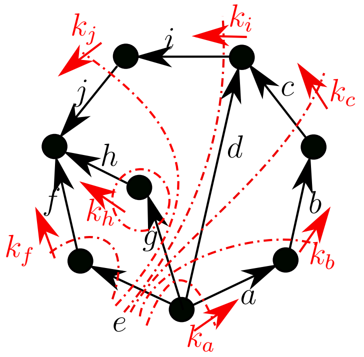
Joint	Type	Position vector $\$_o$	Wrenches $\widehat{\$}^a_{ij}$
a	revolute z	[7.5 1.5 0]	$\widehat{\$}^a_{aR}$ $\widehat{\$}^a_{aS}$ $\widehat{\$}^a_{aU}$ $\widehat{\$}^a_{aV}$ $\widehat{\$}^a_{aW}$
b	prismatic 45^o	[10 4 0]	$\widehat{\$}^a_{bR}$ $\widehat{\$}^a_{bS}$ $\widehat{\$}^a_{bT}$ $\widehat{\$}^a_{bV}$ $\widehat{\$}^a_{bW}$
c	revolute z	[12.5 6.5 0]	$\widehat{\$}^a_{cR}$ $\widehat{\$}^a_{cS}$ $\widehat{\$}^a_{cU}$ $\widehat{\$}^a_{cV}$ $\widehat{\$}^a_{cW}$
d	revolute z	[13.5 − 1 0]	$\widehat{\$}^a_{dR}$ $\widehat{\$}^a_{dS}$ $\widehat{\$}^a_{dU}$ $\widehat{\$}^a_{dV}$ $\widehat{\$}^a_{dW}$
e	revolute z	[0 0 0]	$\widehat{\$}^a_{eR}$ $\widehat{\$}^a_{eS}$ $\widehat{\$}^a_{eU}$ $\widehat{\$}^a_{eV}$ $\widehat{\$}^a_{eW}$
f	revolute z	[1 4 0]	$\widehat{\$}^a_{fR}$ $\widehat{\$}^a_{fS}$ $\widehat{\$}^a_{fU}$ $\widehat{\$}^a_{fV}$ $\widehat{\$}^a_{fW}$
g	revolute z	[4 0 0]	$\widehat{\$}^a_{gR}$ $\widehat{\$}^a_{gS}$ $\widehat{\$}^a_{gU}$ $\widehat{\$}^a_{gV}$ $\widehat{\$}^a_{gW}$
h	revolute z	[5 6.5 0]	$\widehat{\$}^a_{hR}$ $\widehat{\$}^a_{hS}$ $\widehat{\$}^a_{hU}$ $\widehat{\$}^a_{hV}$ $\widehat{\$}^a_{hW}$
i	revolute z	[14.5 12.5 0]	$\widehat{\$}^a_{iR}$ $\widehat{\$}^a_{iS}$ $\widehat{\$}^a_{iU}$ $\widehat{\$}^a_{iV}$ $\widehat{\$}^a_{iW}$
j	prismatic 45^o	[13 3 0]	$\widehat{\$}^a_{jR}$ $\widehat{\$}^a_{jS}$ $\widehat{\$}^a_{jT}$ $\widehat{\$}^a_{jV}$ $\widehat{\$}^a_{jW}$

$$\begin{aligned}[\widehat{A}_D]_{6,50} = \Big[&\widehat{\$}^a_{dR} \quad \widehat{\$}^a_{dS} \quad \widehat{\$}^a_{dU} \quad \widehat{\$}^a_{dV} \quad \widehat{\$}^a_{dW} \quad \widehat{\$}^a_{eR} \quad \widehat{\$}^a_{eS} \quad \widehat{\$}^a_{eU} \quad \widehat{\$}^a_{eV} \dots \\ &\dots \widehat{\$}^a_{eW} \quad \widehat{\$}^a_{gR} \quad \widehat{\$}^a_{gS} \quad \widehat{\$}^a_{gU} \quad \widehat{\$}^a_{gV} \quad \widehat{\$}^a_{gW} \quad \widehat{\$}^a_{aR} \quad \widehat{\$}^a_{aS} \quad \widehat{\$}^a_{aU} \quad \widehat{\$}^a_{aV} \quad \widehat{\$}^a_{aW} \dots \\ &\dots \widehat{\$}^a_{bR} \quad \widehat{\$}^a_{bS} \quad \widehat{\$}^a_{bT} \quad \widehat{\$}^a_{bV} \quad \widehat{\$}^a_{bW} \quad \widehat{\$}^a_{cR} \quad \widehat{\$}^a_{cS} \quad \widehat{\$}^a_{cU} \quad \widehat{\$}^a_{cV} \quad \widehat{\$}^a_{cW} \quad \widehat{\$}^a_{fR} \dots \\ &\dots \widehat{\$}^a_{fS} \quad \widehat{\$}^a_{fU} \quad \widehat{\$}^a_{fV} \quad \widehat{\$}^a_{fW} \quad \widehat{\$}^a_{hR} \quad \widehat{\$}^a_{hS} \quad \widehat{\$}^a_{hU} \quad \widehat{\$}^a_{hV} \quad \widehat{\$}^a_{hW} \quad \widehat{\$}^a_{iR} \quad \widehat{\$}^a_{iS} \dots \\ &\dots \widehat{\$}^a_{iU} \quad \widehat{\$}^a_{iV} \quad \widehat{\$}^a_{iW} \quad \widehat{\$}^a_{jR} \quad \widehat{\$}^a_{jS} \quad \widehat{\$}^a_{jT} \quad \widehat{\$}^a_{jV} \quad \widehat{\$}^a_{jW}\Big]\end{aligned} \tag{5.1}$$

Considering the structural representation, the coupling graph of the mechanism was created and the cut-sets were determined, Figure 20. Contrarily of the action graph, the coupling graph does not have the edges replaced by c_i edges expanded in parallel, where c_i is the number of constraints which are necessary to establish the kinematic pair. So, each edge of the coupling graph must be replaced by five parallel edges. Now, the cut-set matrix $[Q]_{7,50}$ is arranged according to the Equation 2.7. Equation 5.2 shows the cut-set matrix without replacing five parallel edges in each edge of coupling graph.

$$[Q] = \begin{array}{c|cccccccccc|} & d & e & g & a & b & c & f & h & i & j \\ \hline k_a & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ k_b & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ k_c & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ k_f & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ k_h & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ k_i & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ k_j & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \tag{5.2}$$

Figure 20 – Coupling graph with cut-sets representing the backrest mechanism



Combining the matrices $[\widehat{A}_D]_{6,50}$ and $[Q]_{7,50}$ according to the Equation 2.13, the network unit action matrix $[\widehat{A}_N]_{42,50}$ is arranged. The rank of $[\widehat{A}_N]_{42,50}$ is 41. So, the number of redundant constraints is given by Equation 3.10:

$$C_N = C - \text{rank}([\widehat{A}_N]_{6*7,50}) = 50 - 41 = 9 \tag{5.3}$$

The backrest adjustment mechanism from Eleganza Smart bed has nine redundant constraints. The mobility of the mechanism is evaluated by Equation 2.15:

$$F_N = \lambda(n - j - 1) + \sum_{i=1}^{j} f_i + C_N \Rightarrow 6(8 - 10 - 1) + 10 + 9 = 1 \tag{5.4}$$

The backrest adjustment has one degree of freedom, which is actuated by the prismatic joint b. The dual matroid $\mathcal{M}^*_{AN}$ derived from matrix $[\widehat{A}_N]_{42,50}$ can be created. As the seed mechanism has nine redundant constraints, the cobases of the set $\mathcal{B}^*$ have cardinality equal to nine. The dual matroid $\mathcal{M}^*_{AN}$ has 838.941 cobases.

All the cobases were arranged into the cobases binary matrix $[N]_{838451,50}$. To select an acceptable number of self-aligning mechanisms some design requirements must be defined. Considering the Eleganza hospital bed as a commercial product with the manufacturing well established, the changes must be technically and commercially satisfactory.

It was decided that the actuator structure RPR must remain and the remaining joints must be lower pairs. The lower pair class is defined by the revolute, prismatic, universal, spherical and planar pairs.

With the strategy of selection defined, in this work the mechanisms will be selected according to following design requirements:

1. The prismatic actuated joint b and the adjacent joints a and c cannot be modified;

2. The joints d, e, f, g, h, i and j can be modified, but they must be lower pairs.

The reason for the design requirement (i) is to maintain the actuation structure RPR, once the RPR structure is a commercial item. Requirement (ii) was defined because the other joints must be modified to remove the redundant constraints, but the rigidity of the mechanism cannot decrease, so the joints must remain lower pairs.

The design requirements must be transformed into selection criteria. For the design requirement (i), the respective constraints to the joints a, b and c cannot be removed, so the criterion 1 must to select all the cobases where the elements of this joints are not present. Mathematically, $\{\forall i = 1, 2, \ldots, 838451 \quad | B_i^* \in K_1 \Leftrightarrow n(i,k) = 0, k = 16, 17, \ldots, 30\}$.

For the design requirement (ii), five specific sets of constraints can represent lower pairs to be selected. Considering the revolute joint d, the constraints which represent this kinematic pair are $\widehat{\$}^a_{dR}$, $\widehat{\$}^a_{dS}$, $\widehat{\$}^a_{dU}$, $\widehat{\$}^a_{dV}$ and $\widehat{\$}^a_{dW}$, the constraint $\widehat{\$}^a_{dT}$ is not present because it is the degree of freedom of the joint, a rotation around the z-axis. Now, it will be presented how the selection criteria for the design requirement (ii) were created.

The revolute joint d around z-axis can remain as revolute joint so the elements $n(i,1)$, $n(i,2)$, $n(i,3)$, $n(i,4)$, $n(i,5)$ are equal to zero and none constraint will be replaced by freedom, mathematicaly, $\{\forall i = 1, 2, \ldots, 838451 \quad | B_i^* \in K_2 \Leftrightarrow n(i,k) = 0, k = 1, 2, \ldots, 5\}$.

The revolute joint d around z-axis has two possibilities to be transformed into a universal joint: the constraints $\widehat{\$}^a_{dR}$ or $\widehat{\$}^a_{dS}$ must be removed, so the new joint d would be a universal joint with the freedoms around axes x and z or around axes y and z, respectively. In this case the cobases i will be selected if the elements $n(i,2)$,

$n(i,3)$, $n(i,4)$, $n(i,5)$ are equal to zero and $n(i,1)$ is equal to one, or if the elements $n(i,1)$, $n(i,3)$, $n(i,4)$, $n(i,5)$ are equal to zero and $n(i,2)$ is equal to one, mathematically, $\{\forall i = 1,2,\ldots,838451 \quad | B_i^* \in K_2 \Leftrightarrow \{n(i,1) = 1 \quad \text{AND} \quad n(i,k) = 0, k = 2,3\ldots5\} \quad \text{OR} \quad \{n(i,2) = 1 \quad \text{AND} \quad n(i,k) = 0, k = 1,3,4,5\}\}$.

The joint d can also be transformed into a spherical pair, so the constraints $\widehat{\$}^a_{dR}$ and $\widehat{\$}^a_{dS}$ must be removed. Thus, the cobases i will be selected if the elements $n(i,3)$, $n(i,4)$, $n(i,5)$ are equal to zero and $n(i,1)$ and $n(i,2)$ are equal to one, mathematically, $\{\forall i = 1,2,\ldots,838451 \quad | B_i^* \in K_2 \Leftrightarrow n(i,1) = n(i,2) = 1 \quad \text{AND} \quad n(i,k) = 0, k = 3,4,5\}$.

Lastly, the joint d can be transformed into a planar pair, the constraints of translation $\widehat{\$}^a_{dU}$ and $\widehat{\$}^a_{dV}$ along the axes x and y must be removed. Therefore, the cobases i will be selected if the elements $n(i,1)$, $n(i,2)$, $n(i,5)$ are equal to zero and $n(i,3)$, $n(i,4)$ are equal to one, $\{\forall i = 1,2,\ldots,838451 \quad | B_i^* \in K_2 \Leftrightarrow n(i,1) = n(i,2) = n(i,5) = 0 \quad \text{AND} \quad n(i,3) = n(i,4) = 1\}$. Equation 5.5 shows the criteria and Table 3 organizes the sets of constraints to transform the joint d in the lower pairs mentioned above.

$$\begin{gathered}\{\forall i = 1,2,\ldots,838451 \quad | B_i^* \in K_2 \Leftrightarrow \{\{n(i,k) = 0, k = 1,2,\ldots,5\} \\ \text{OR} \quad \{\{n(i,1) = 1 \quad \text{AND} \quad n(i,k) = 0, k = 2,3\ldots5\} \quad \text{OR} \\ \{n(i,2) = 1 \quad \text{AND} \quad n(i,k) = 0, k = 1,3,4,5\}\} \quad \text{OR} \\ \{n(i,1) = n(i,2) = 1 \quad \text{AND} \quad n(i,k) = 0, k = 3,4,5\} \quad \text{OR} \\ \{n(i,1) = n(i,2) = n(i,5) = 0 \quad \text{AND} \quad n(i,3) = n(i,4) = 1\}\}\end{gathered} \tag{5.5}$$

Table 3 – Set of wrenches to transform the joint d in lower pairs

Type of Joint	**Contraints/Wrenches**
Revolute pair (original)	$\widehat{\$}^a_{dR}$, $\widehat{\$}^a_{dS}$, $\widehat{\$}^a_{dU}$, $\widehat{\$}^a_{dV}$, $\widehat{\$}^a_{dW}$
Universal pair (around z and x axes)	$\widehat{\$}^a_{dS}$, $\widehat{\$}^a_{dU}$, $\widehat{\$}^a_{dV}$, $\widehat{\$}^a_{dW}$
Universal pair (around z and y axes)	$\widehat{\$}^a_{dR}$, $\widehat{\$}^a_{dU}$, $\widehat{\$}^a_{dV}$, $\widehat{\$}^a_{dW}$
Spherical pair	$\widehat{\$}^a_{dU}$, $\widehat{\$}^a_{dV}$, $\widehat{\$}^a_{dW}$
Planar pair	$\widehat{\$}^a_{dR}$, $\widehat{\$}^a_{dS}$, $\widehat{\$}^a_{dW}$

These criteria must be replied for the other revolute joints e, f, g, h and i. As j is originally a prismatic joint, the selection criteria are different. A prismatic joint cannot be transformed into a revolute, universal or spherical joint, because these joints do not have any translational freedom. So, a prismatic joint along the x-axis can be transformed into planar pairs along the planes xy or xz.

To transform the prismatic joint j in a planar pair along the plane xy, the constraints $\widehat{\$}^a_{dT}$ and $\widehat{\$}^a_{dV}$ must be removed. Thus, the cobases i will be selected if the elements $n(i,46)$ $n(i,47)$ and $n(i,50)$ are equal to zero and $n(i,48)$, $n(i,49)$ are equal to one, mathematically: $\{\forall i = 1, 2, \ldots, 838451 \quad |B^*_i \in K_2 \Leftrightarrow n(i,46) = n(i,47) = n(i,50) = 0 \quad \text{AND} \quad n(i,48) = n(i,49) = 1\}$.

To transform the prismatic joint j in a plane pair along its axis and z-axis, the constraints $\widehat{\$}^a_{dS}$ and $\widehat{\$}^a_{dW}$ must be removed. Therefore, the cobases i will be selected if the elements $n(i,46)$ $n(i,48)$ and $n(i,49)$ are equal to zero and $n(i,47)$, $n(i,50)$ are equal to one, mathematically: $\{\forall i = 1, 2, \ldots, 838451 \quad |B^*_i \in K_2 \Leftrightarrow n(i,46) = n(i,48) = n(i,49) = 0 \quad \text{AND} \quad n(i,47) = n(i,50) = 1\}$.

As the prismatic pair is a lower pair, the joint j can maintain all the constraints, so the cobases i will be selected if the elements $n(i,46)$, $n(i,47)$, $n(i,48)$, $n(i,49)$ and $n(i,50)$ are equal to zero, mathematically: $\{\forall i = 1, 2, \ldots, 838451 \quad |B^*_i \in K_2 \Leftrightarrow n(i,k) = 0, k = 46, 47, \ldots, 50\}$.

The selection criteria discussed above were applied to the binary cobases matrix $[N]_{838941,50}$. Criterion 1 created a subset K_1 with 4,416 cobases, which corresponds to 0.52% of the entire set of cobases $\mathcal{B}^*$. Criterion 2 created a subset K_2 with 39.957 cobases, that corresponds to 4.7% of the entire set $\mathcal{B}^*$.

The intersection between subsets $K_1 \cap K_2$ created an empty subset, *i.e.* no self-aligning mechanism derived from the seed mechanism complies with the design requirements proposed. This occurs because, by the view of self-aligning, the requirement of maintaining all the constraints respective to RPR structure is too restrictive.

Although frustrating, the empty set shows to the designers that the requirements are not being accomplished. Thus, the design requirements must be reviewed. It was decided the actuated joint b cannot be modified, aiming to reduce the work and the costs to adapt the seed mechanisms, only the joints of the base (a, d, e and g) and the joint i can be modified into lower pairs. Summarizing, the other joints c, f, h and j cannot be modified. So the new design requirements are:

1. The prismatic actuated joint b cannot be modified;

2. The joints a, d, e, g and i can be transformed into lower pairs;

3. The joints c, f, h and j cannot be modified.

The design requirements are then transformed into selection criteria which evaluate the rows of the cobases binary matrix $[N]_{838941,50}$. The condition representing the design requirements are shown in Tables 4 and 5.

Table 4 – Set of constraints and respective binary conditions.

Criterion 1 - K_1			
Contraints		Condition	
$\{\ \$^a_{bR},\quad \$^a_{bS},\quad \$^a_{bT},\quad \$^a_{bV},\quad \$^a_{bW}\ \}$		$n_{(i,j)} = 0$	
Criterion 2 - K_2			
Constraints	Condition	Constraints	Condition
$\{\$^a_{aS}, \$^a_{aU}, \$^a_{aV}, \$^a_{aW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{aR}\ \}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{aR}, \$^a_{aU}, \$^a_{aV}, \$^a_{aW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{aS},\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{aU}, \$^a_{aV}, \$^a_{aW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{aR}, \$^a_{aS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{aR}, \$^a_{aS}, \$^a_{aW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{aU}, \$^a_{aV}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{aR}, \$^a_{aS}, \$^a_{aU}, \$^a_{aV}, \$^a_{aW}\}$		$n_{(i,j)} = 0$	
AND			
$\{\$^a_{dS}, \$^a_{dU}, \$^a_{dV}, \$^a_{dW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{dR},\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{dR}, \$^a_{dU}, \$^a_{dV}, \$^a_{dW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{dS},\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{dU}, \$^a_{dV}, \$^a_{dW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{dR}, \$^a_{dS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{dR}, \$^a_{dS}, \$^a_{dW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{dU}, \$^a_{dV}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{dR}, \$^a_{dS}, \$^a_{dU}, \$^a_{dV}, \$^a_{dW}\}$		$n_{(i,j)} = 0$	
AND			
$\{\$^a_{eS}, \$^a_{eU}, \$^a_{eV}, \$^a_{eW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{eR},\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{eR}, \$^a_{eU}, \$^a_{eV}, \$^a_{eW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{eS},\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{eU}, \$^a_{eV}, \$^a_{eW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{eR}, \$^a_{eS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{eR}, \$^a_{eS}, \$^a_{eW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{eU}, \$^a_{eV}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{eR}, \$^a_{eS}, \$^a_{eU}, \$^a_{eV}, \$^a_{eW}\}$		$n_{(i,j)} = 0$	
AND (continue in Table 5)			

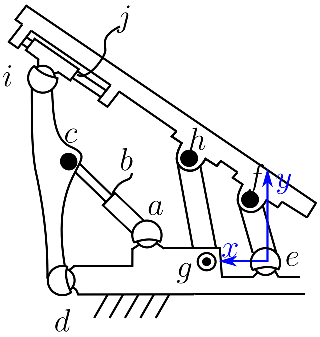
Table 5 – Sets of constraints and respective conditions (continuation).

... AND (continuing Table 4)			
$\{\$^a_{gS}, \$^a_{gU}, \$^a_{gV}, \$^a_{gW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{gR},\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{gR}, \$^a_{gU}, \$^a_{gV}, \$^a_{gW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{gS},\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{gU}, \$^a_{gV}, \$^a_{gW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{gR}, \$^a_{gS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{gR}, \$^a_{gS}, \$^a_{gW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{gU}, \$^a_{gV}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{gR}, \$^a_{gS}, \$^a_{gU}, \$^a_{gV}, \$^a_{gW}\}$		$n_{(i,j)} = 0$	
AND			
$\{\$^a_{iS}, \$^a_{iU}, \$^a_{iV}, \$^a_{iW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{iR},\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{iR}, \$^a_{iU}, \$^a_{iV}, \$^a_{iW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{iS},\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{iU}, \$^a_{iV}, \$^a_{iW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{iR}, \$^a_{iS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{iR}, \$^a_{iS}, \$^a_{iW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{iU}, \$^a_{iV}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{iR}, \$^a_{iS}, \$^a_{iU}, \$^a_{iV}, \$^a_{iW}\}$		$n_{(i,j)} = 0$	
Criterion 3 - K_3			
Contraints		Condition	
$\{\ \$^a_{cR},\quad \$^a_{cS},\quad \$^a_{cT},\quad \$^a_{cV},\quad \$^a_{cW}\}$		$n_{(i,j)} = 0$	
AND			
$\{\ \$^a_{hR},\quad \$^a_{hS},\quad \$^a_{hT},\quad \$^a_{hV},\quad \$^a_{hW}\}$		$n_{(i,j)} = 0$	
AND			
$\{\ \$^a_{fR},\quad \$^a_{fS},\quad \$^a_{fT},\quad \$^a_{fV},\quad \$^a_{fW}\}$		$n_{(i,j)} = 0$	
AND			
$\{\ \$^a_{jR},\quad \$^a_{jS},\quad \$^a_{jT},\quad \$^a_{jV},\quad \$^a_{jW}\}$		$n_{(i,j)} = 0$	
AND			

Each logical criterion created a subset with cobases which comply with the design requirement analyzed. Criterion 1 created a subset K_1 with 323,778 cobases which corresponds to 36.6% of the entire set $\mathcal{B}^*$. The subset K_2 created from criterion 2 has $17,079$ cobases, corresponding to 2.03% of the entire set $\mathcal{B}^*$. Finally, criterion 3 created a subset K_3 with 6,052 cobases which corresponds to 0.72% of the entire set $\mathcal{B}^*$.

By intersection among sets it is possible to create the final subset $K_F = K_1 \cap K_2 \cap K_3$. K_F is composed of 10 cobases, corresponding to 0,000019% of the entire set $\mathcal{B}^*$. Therefore, the backrest adjustment mechanism has 10 self-aligning mechanisms which satisfy the design requirements proposed in the second round. Figure 21 shows one of these mechanisms.

Figure 21 – New concept of a self-aligning backrest mechanism.

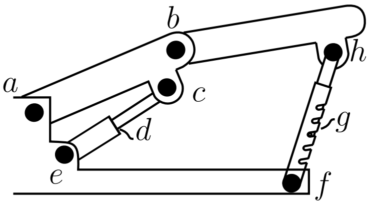


The new mechanism of Figure 21 corresponds to the cobasis $\{\$^a_{aR}, \$^a_{aS}, \$^a_{dR}, \$^a_{dS}, \$^a_{eR}, \$^a_{eS}, \$^a_{gW}, \$^a_{iR}, \$^a_{iS}\}$, so the joints a, d, g and i are now spherical, and the joint g is now a cylindrical, while the others joints were not modified. This mechanism has no redundant constraints, $C_N = 0$.

5.1.2 Case II : Leg rest adjustment mechanism

The topological structure of the leg rest adjustment mechanism will be analyzed, then, a set of self-aligning mechanisms will be selected. The structural representation of the mechanism was arranged. The mechanism has eight joints and seven links, the type, position vectors and wrenches of the joints are shown in Table 6 according to the letters labelled in Figure 22. The set of joints' positions was determined in a pose which the mechanism does not have extra mobility.

Figure 22 – Leg rest adjustment mechanism from Eleganza 3XC.



All the wrenches are ordered into the unit action matrix $[\widehat{A}_D]_{6,40}$, shown in Equation 5.6.

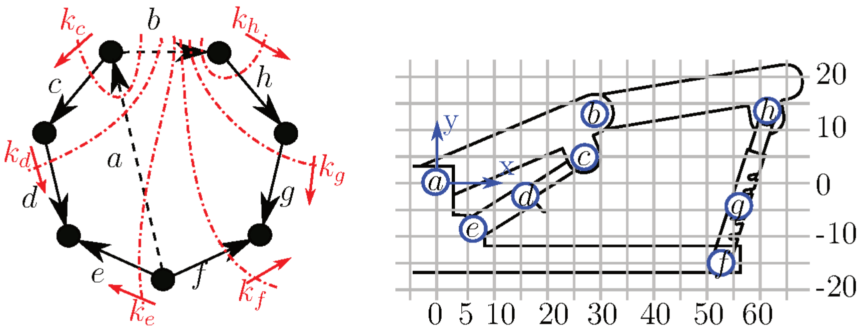
$$\begin{aligned}[\widehat{A}_D]_{6,40} = \Big[&\widehat{\$}^a_{aR} \quad \widehat{\$}^a_{aS} \quad \widehat{\$}^a_{aU} \quad \widehat{\$}^a_{aV} \quad \widehat{\$}^a_{aW} \quad \widehat{\$}^a_{bR} \quad \widehat{\$}^a_{bS} \quad \widehat{\$}^a_{bU} \quad \widehat{\$}^a_{bV} \dots \\ &\dots \widehat{\$}^a_{bW} \quad \widehat{\$}^a_{cR} \quad \widehat{\$}^a_{cS} \quad \widehat{\$}^a_{cU} \quad \widehat{\$}^a_{cV} \quad \widehat{\$}^a_{cW} \quad \widehat{\$}^a_{dR} \quad \widehat{\$}^a_{dS} \quad \widehat{\$}^a_{dT} \quad \widehat{\$}^a_{dV} \quad \widehat{\$}^a_{dW} \dots \\ &\dots \widehat{\$}^a_{eR} \quad \widehat{\$}^a_{eS} \quad \widehat{\$}^a_{eU} \quad \widehat{\$}^a_{eV} \quad \widehat{\$}^a_{eW} \quad \widehat{\$}^a_{fR} \quad \widehat{\$}^a_{fS} \quad \widehat{\$}^a_{fU} \quad \widehat{\$}^a_{fV} \quad \widehat{\$}^a_{fW} \quad \widehat{\$}^a_{gR} \dots \\ &\dots \widehat{\$}^a_{gS} \quad \widehat{\$}^a_{gT} \quad \widehat{\$}^a_{gV} \quad \widehat{\$}^a_{gW} \quad \widehat{\$}^a_{hR} \quad \widehat{\$}^a_{hS} \quad \widehat{\$}^a_{hU} \quad \widehat{\$}^a_{hV} \quad \widehat{\$}^a_{hW}\Big] \end{aligned} \qquad (5.6)$$

Table 6 – Type of the joints and respective wrenches of the leg rest adjustment mechanism.

Joint	**Type**	**Position vector $\$_o$**	**Wrenches $\widehat{\$^a}_{ij}$**
a	revolute z	[0 0 0]	$\widehat{\$}^a_{aR}$ $\widehat{\$}^a_{aS}$ $\widehat{\$}^a_{aU}$ $\widehat{\$}^a_{aV}$ $\widehat{\$}^a_{aW}$
b	revolute z	[27.5 12.5 0]	$\widehat{\$}^a_{bR}$ $\widehat{\$}^a_{bS}$ $\widehat{\$}^a_{bU}$ $\widehat{\$}^a_{bV}$ $\widehat{\$}^a_{bW}$
c	revolute z	[25 5 0]	$\widehat{\$}^a_{cR}$ $\widehat{\$}^a_{cS}$ $\widehat{\$}^a_{cU}$ $\widehat{\$}^a_{cV}$ $\widehat{\$}^a_{cW}$
d	primatic 45^o	[15 – 2.5 0]	$\widehat{\$}^a_{dR}$ $\widehat{\$}^a_{dS}$ $\widehat{\$}^a_{dT}$ $\widehat{\$}^a_{dV}$ $\widehat{\$}^a_{dW}$
e	revolute z	[5 – 10 0]	$\widehat{\$}^a_{eR}$ $\widehat{\$}^a_{eS}$ $\widehat{\$}^a_{eU}$ $\widehat{\$}^a_{eV}$ $\widehat{\$}^a_{eW}$
f	revolute z	[52.5 – 15 0]	$\widehat{\$}^a_{fR}$ $\widehat{\$}^a_{fS}$ $\widehat{\$}^a_{fU}$ $\widehat{\$}^a_{fV}$ $\widehat{\$}^a_{fW}$
g	prismatic 45^o	[55.5 – 5 0]	$\widehat{\$}^a_{gR}$ $\widehat{\$}^a_{gS}$ $\widehat{\$}^a_{gU}$ $\widehat{\$}^a_{gV}$ $\widehat{\$}^a_{gW}$
h	revolute z	[62.5 15 0]	$\widehat{\$}^a_{hR}$ $\widehat{\$}^a_{hS}$ $\widehat{\$}^a_{hU}$ $\widehat{\$}^a_{hV}$ $\widehat{\$}^a_{hW}$

The coupling graph with cut-sets is shown in Figure 23. The edges a and b were chosen as chords, the other six edges are branches which correspond to six cut-sets. The present graph is used to arrange the cut-sets matrix $[Q]_{6,40}$. Note the eight edges of the coupling graph should be replaced by forty edges, arranged five-by-five in parallel to build the action graph.

Figure 23 – Coupling graph with cut-sets and joints' position.



(a) (b)

The network unit action matrix $[\widehat{A}_N]_{36,40}$ is obtained by combining the matrices $[\widehat{A}_D]_{6,40}$ and $[Q]_{6,40}$ according to Equation 2.13. The rank of $[\widehat{A}_N]_{36,40}$ is 34, applying it to Equation 3.10:

$$C_N = C - \text{rank}([\widehat{A}_N]_{6*6,40}) = 40 - 34 = 6 \tag{5.7}$$

The leg rest adjustment mechanism from Eleganza Smart bed has six redundant constraints. The mobility of the mechanism is evaluated by the Equation 2.15:

$$F_N = \lambda(n - j - 1) + \sum_{i=1}^{j} f_i + C_N = 6(7 - 8 - 1) + 8 + 6 = 2 \tag{5.8}$$

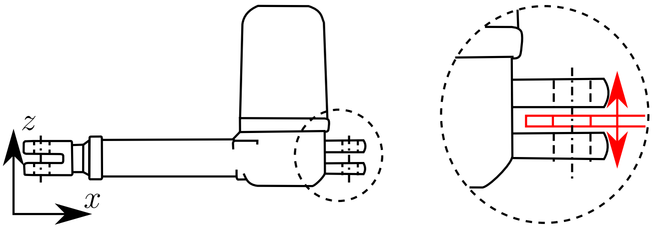
The dual matroid $\mathcal{M}^*_{AN}$ derived from the matrix $[\widehat{A}_N]_{36,40}$ is created. As the seed mechanism has six redundant constraints, the cobases of the set $\mathcal{B}^*$ have cardinality equal to six. The dual matroid has 15.704 cobases.

According to Equation 4.2, all the cobases were arranged into the cobases matroid matrix $[N]_{15704,40}$.

Before listing the design requirements, there are some special features of the leg rest adjustment mechanism that requires special attention. Firstly, the mechanism has an off-the-shelf commercial actuator. This means that the joints RPR of the actuator structure, named as c, d and e, respectively, in Figure 22, have limitations towards self-aligning.

A freedom can be added to RPR structure without modification of the actuator is a translation normal to one of the revolute joints. This freedom may be added by enabling a clearance between the fixation fork of the actuator and the link that will support the actuator. A representation of this freedom is shown in Figure 24.

Figure 24 – Cylindrical pair in the revolute joint of the actuator.

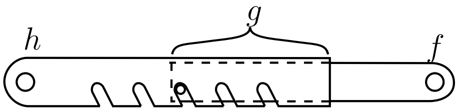


By creating an axial clearance, the revolute joint becomes a cylindrical joint. The additional of this translation freedom may only be done in one of the revolute joints at each time. If both revolute joints allow this additional freedom, the system would increase a degree of freedom and the actuator would become loose in the direction of the translations added. Therefore, the RPR structure can become either CPR or RPC.

Other special feature which must be explained is that RPR structure formed by joints h, g and f is another off-the-shelf commercial component, see Figure 25. This component has a channel that acts as a prismatic joint, corresponding to joint g, which is a ratchet. It is fixed by screws on each side of the component, forming two revolute joints h and j. Changes to the ratchet are unfeasible, so the joint g must remain unchanged. The remaining joints f and h may receive additional freedoms.

After discussing the special features, the design requirements for this case study are the following:

Figure 25 – RPR structure with a ratchet.



1. The joints c, d and e can exclusively creates the structures CPR or RPC;
2. The prismatic actuated joint g cannot be modified;
3. A maximum of two constraints can be removed in each of the joints a, b, h, and f.

The reasons for the design requirements (i) and (ii) were previously presented. Requirement (iii) is defined because the stiffness of the mechanism cannot be decreased. If a joint has less than three constraints the mechanism may become flexible, hence reducing the users' safety.

The design requirements are then transformed into selection criteria presented in Table 7, in which the constraint sets for each criterion are listed with the respective binary conditions, represented by $n(i,j)$. In $n(i,j)$, i is the i^{th} row of the binary matrix, each row is related to a different self-aligning mechanism and j is associated to the j^{th} constraint, whose number follows the arrangement of matrix $[A_D]$ (Equation 5.6).

The criteria listed in Table 7 are used to create logical sentences which are used to select the bases that complies with the design requirements (i), (ii) and (iii). The criteria are applied to the cobases binary matrix $[\mathbf{N}]_{15.704,40}$:

(I) $\{\forall i = 1,2,\ldots,15.704 | B_i^* \in K_1 \Leftrightarrow \{n(i,26) = n(i,27) = n(i,28) = n(i,29) = 0 \quad \text{AND} \quad n(i,30) = 1 \quad \text{AND} \quad n(i,31) = n(i,32) = n(i,33) = n(i,34) = n(i,35) = 0 \quad \text{AND} \quad n(i,36) = n(i,37) = n(i,38) = n(i,39) = n(i,40) = 0\} \quad OR \quad \{n(i,26) = n(i,27) = n(i,28) = n(i,29) = n(i,30) = 0 \quad \text{AND} \quad n(i,31) = n(i,32) = n(i,33) = n(i,34) = n(i,35) = 0 \quad \text{AND} \quad n(i,36) = n(i,37) = n(i,38) = n(i,39) = 0 \quad \text{AND} \quad n(i,40) = 1\}\}$

(II) $\{\forall i = 1,2,\ldots,15.704 | B_i^* \in K_2 \Leftrightarrow n(i,16) = n(i,17) = n(i,18) = n(i,19) = n(i,20) = 0\}$

(III) $\{\forall i = 1,2,\ldots,15.704 | B_i^* \in K_3 \Leftrightarrow n(i,1) + n(i,2) + n(i,3) + n(i,4) + n(i,5) \leq 2 \quad \text{AND} \quad n(i,6)+n(i,7)+n(i,8)+n(i,9)+n(i,10) \leq 2 \quad \text{AND} \quad n(i,11)+n(i,12)+$

Table 7 – Sets of constraints and respective binary conditions.

Criterion 1 - K_1			
Constraints	Condition	Constraints	Condition
$\{\$^a_{cR}, \$^a_{cS}, \$^a_{cU}, \$^a_{cV}, \$^a_{dR}, \$^a_{dS}, \$^a_{dT}, \$^a_{dV}, \$^a_{dW} \$^a_{eR}, \$^a_{eS}, \$^a_{eU}, \$^a_{eV}, \$^a_{eW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{cW}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{cR}, \$^a_{cS}, \$^a_{cU}, \$^a_{cV}, \$^a_{cW}, \$^a_{dR}, \$^a_{dS}, \$^a_{dT},, \$^a_{dV}, \$^a_{dW}, \$^a_{eR}, \$^a_{eS}, \$^a_{eU}, \$^a_{eV}\}$	$n_{(i,j)} = 0$	$\{\$^a_{eW}\}$	$n_{(i,j)} = 1$
Criterion 2 - K_2			
Contraints		Condition	
$\{\ \$^a_{gR}, \$^a_{gS}, \$^a_{gT}, \$^a_{gV}, \$^a_{gW}\}$		$n_{(i,j)} = 0$	
Criterion 3 - K_3			
$\{\$^a_{aR}, \$^a_{aS}, \$^a_{aU}, \$^a_{aV}, \$^a_{aW}\}$		$\sum n_{(i,j)} \leq 2$	
AND			
$\{\$^a_{bR}, \$^a_{bS}, \$^a_{bU}, \$^a_{bV}, \$^a_{bW}\}$		$\sum n_{(i,j)} \leq 2$	
AND			
$\{\$^a_{hR}, \$^a_{hS}, \$^a_{hU}, \$^a_{hV}, \$^a_{hW}\}$		$\sum n_{(i,j)} \leq 2$	
AND			
$\{\$^a_{fR}, \$^a_{fS}, \$^a_{fU}, \$^a_{fV}, \$^a_{fW}\}$		$\sum n_{(i,j)} \leq 2$	

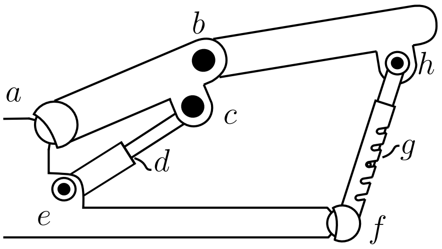
$$n(i,13) + n(i,14) + n(i,15) \leq 2 \quad \text{AND} \quad n(i,21) + n(i,22) + n(i,23) + n(i,24) + n(i,25) \leq 2\}$$

The criterion K_1 generates a subset with 232 bases, corresponding to $1,47\%$ of the entire set $\mathcal{B}^*$. The criterion K_2 is related to the joint g and creates a subset with 7.587 bases, that corresponds to $48,31\%$ of the family of cobases $\mathcal{B}^*$. The last criterion K_3 generates a subset with 14.831 bases, representing $94,44\%$ of the entire set $\mathcal{B}^*$.

It is possible to define the final subset by the intersection among the criteria, $K_F = K_1 \cap K_2 \cap K_3$ that represents the cobases which satisfy all the criteria. K_F is composed of 78 cobases, corresponding to $0,49\%$ of the total number of cobases. Thus, the leg rest seed mechanism has 78 self-aligning mechanisms that satisfy the design requirements proposed. Figure 26 exemplifies a solution to the overconstrained seed mechanism.

In this new mechanism the joints a and f are spherical. For these joints, clearances and flexible bushings can be evaluated to create spherical equivalent joints. The joints e and h are now cylindrical, when previously they were both revolute joints. The mechanism from Figure 26 has $C_N = 0$, which means that it does not have redundant constraints.

Figure 26 – New concept of a self-aligning leg rest mechanism.



5.2 LAR HOSPITAL BED

This section presents the analysis of the hospital bed developed by researchers at the Laboratory of Applied Robotics (LAR) of the UFSC. Design requirements are planned to convert the backrest and the leg rest mechanisms into self-aligning mechanisms.

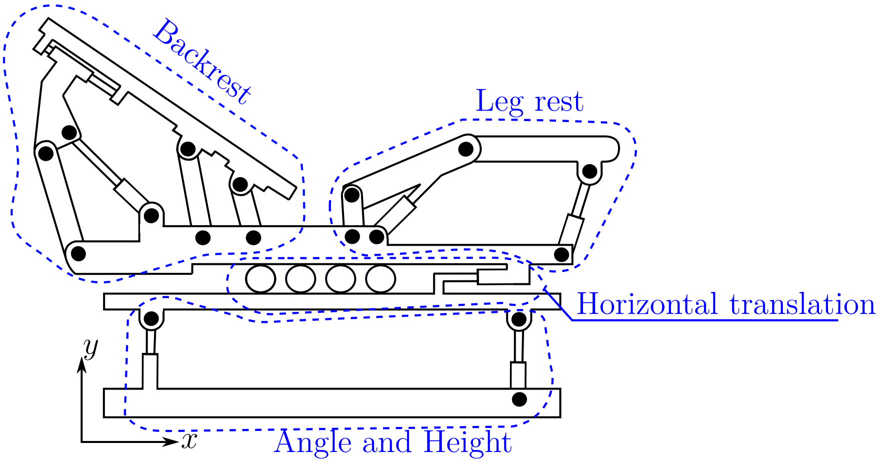
The kinematic chain of the mechanisms were established according to a methodology of synthesis based on graph enumeration. Then, the fixed link, the two output links and the actuated joint were determined according to methods developed by Murai (2019). As the workspace of the mechanisms is theoretically planar, the synthesis process was developed in planar space, consequently, the mechanisms have redundant constraints. A structural representation of the mechanisms is shown in Figure 27.

The bed mechanism is divided into five independent sections. The backrest adjustment mechanism, the leg rest adjustment mechanism, the horizontal translation mechanism, and the height and angle adjustment mechanism are shown in Figure 27. The horizontal translation mechanism facilitates the patient to get out of bed, normally the procedure is performed with help of nurses who make exaggerated efforts. Using the horizontal translation mechanism with backrest mechanism the efforts are decreased, improving the health of nurses. The rotational mechanism, responsible for lateral rotating of the bed cannot be represented in the xy-plane.

5.2.1 Case III: Backrest adjustment mechanism

In this section, the backrest adjustment mechanism is considered as seed mechanism. It is analyzed employing Davies' method, then the redundant constraints are evaluated. By means of Matroid theory all the cobases are listed, hence all the self-aligning mechanism derived from the seed mechanism are enumerated. The selection method is then applied to select a set of self-aligning mechanisms which satisfies the established design requirements.

Figure 27 – Structural representation of LAR hospital bed.



A structural representation of the backrest adjustment mechanism is shown in Figure 28. The mechanism has ten joints and eight links. The type, the position vector and the wrenches of each joint are shown in Table 8.

Figure 28 – Backrest adjustment mechanism from LAR hospital bed.

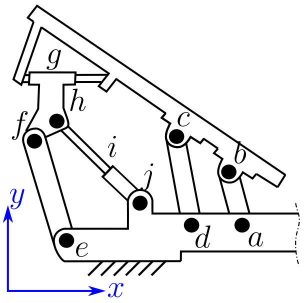


Table 8 – Type of the joints and respective wrenches.

Joint	**Type**	**Position vector $\$_o$**	**Wrenches $\widehat{\$^a}_{ij}$**
a	revolute z	$[14 \quad 1 \quad 0]$	$\widehat{\$}^a_{aR}$ $\widehat{\$}^a_{aS}$ $\widehat{\$}^a_{aU}$ $\widehat{\$}^a_{aV}$ $\widehat{\$}^a_{aW}$
b	revolute z	$[13 \quad 5 \quad 0]$	$\widehat{\$}^a_{bR}$ $\widehat{\$}^a_{bS}$ $\widehat{\$}^a_{bU}$ $\widehat{\$}^a_{bV}$ $\widehat{\$}^a_{bW}$
c	revolute z	$[9 \quad 8 \quad 0]$	$\widehat{\$}^a_{cR}$ $\widehat{\$}^a_{cS}$ $\widehat{\$}^a_{cU}$ $\widehat{\$}^a_{cV}$ $\widehat{\$}^a_{cW}$
d	revolute z	$[10 \quad 1 \quad 0]$	$\widehat{\$}^a_{dR}$ $\widehat{\$}^a_{dS}$ $\widehat{\$}^a_{dU}$ $\widehat{\$}^a_{dV}$ $\widehat{\$}^a_{dW}$
e	revolute z	$[0 \quad 0 \quad 0]$	$\widehat{\$}^a_{eR}$ $\widehat{\$}^a_{eS}$ $\widehat{\$}^a_{eU}$ $\widehat{\$}^a_{eV}$ $\widehat{\$}^a_{eW}$
f	revolute z	$[-2.5 \quad 8 \quad 0]$	$\widehat{\$}^a_{fR}$ $\widehat{\$}^a_{fS}$ $\widehat{\$}^a_{fU}$ $\widehat{\$}^a_{fV}$ $\widehat{\$}^a_{fW}$
g	prismatic x	$[-1 \quad 12.5 \quad 0]$	$\widehat{\$}^a_{gR}$ $\widehat{\$}^a_{gS}$ $\widehat{\$}^a_{gT}$ $\widehat{\$}^a_{gV}$ $\widehat{\$}^a_{gW}$
h	revolute z	$[-1 \quad 9 \quad 0]$	$\widehat{\$}^a_{hR}$ $\widehat{\$}^a_{hS}$ $\widehat{\$}^a_{hU}$ $\widehat{\$}^a_{hV}$ $\widehat{\$}^a_{hW}$
i	prismatic 45^o	$[3 \quad 6 \quad 0]$	$\widehat{\$}^a_{iR}$ $\widehat{\$}^a_{iS}$ $\widehat{\$}^a_{iT}$ $\widehat{\$}^a_{iV}$ $\widehat{\$}^a_{iW}$
j	revolute z	$[5 \quad 3 \quad 0]$	$\widehat{\$}^a_{jR}$ $\widehat{\$}^a_{jS}$ $\widehat{\$}^a_{jT}$ $\widehat{\$}^a_{jV}$ $\widehat{\$}^a_{jW}$

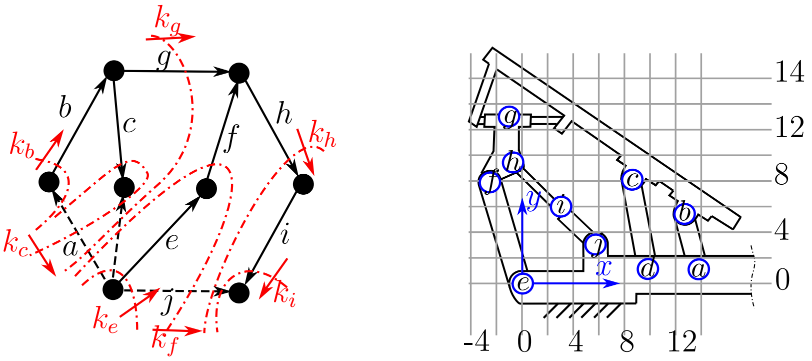
The wrenches are created according to the position point shown in Figure 30(b) and they are ordered into the unit action matrix $[\widehat{A}_D]_{6,50}$ following the structure of Equation

5.9:

$$
\begin{aligned}
[\widehat{A}_D]_{6,50} = \Big[& \widehat{\$}^a_{aR} \quad \widehat{\$}^a_{aS} \quad \widehat{\$}^a_{aU} \quad \widehat{\$}^a_{aV} \quad \widehat{\$}^a_{aW} \quad \widehat{\$}^a_{bR} \quad \widehat{\$}^a_{bS} \quad \widehat{\$}^a_{bU} \quad \widehat{\$}^a_{bV} \dots \\
& \dots \widehat{\$}^a_{bW} \quad \widehat{\$}^a_{cR} \quad \widehat{\$}^a_{cS} \quad \widehat{\$}^a_{cU} \quad \widehat{\$}^a_{cV} \quad \widehat{\$}^a_{cW} \quad \widehat{\$}^a_{dR} \quad \widehat{\$}^a_{dS} \quad \widehat{\$}^a_{dU} \quad \widehat{\$}^a_{dV} \quad \widehat{\$}^a_{dW} \dots \\
& \dots \widehat{\$}^a_{eR} \quad \widehat{\$}^a_{eS} \quad \widehat{\$}^a_{eT} \quad \widehat{\$}^a_{eV} \quad \widehat{\$}^a_{eW} \quad \widehat{\$}^a_{fR} \quad \widehat{\$}^a_{fS} \quad \widehat{\$}^a_{fU} \quad \widehat{\$}^a_{fV} \quad \widehat{\$}^a_{fW} \quad \widehat{\$}^a_{gR} \dots \\
& \dots \widehat{\$}^a_{gS} \quad \widehat{\$}^a_{gT} \quad \widehat{\$}^a_{gV} \quad \widehat{\$}^a_{gW} \quad \widehat{\$}^a_{hR} \quad \widehat{\$}^a_{hS} \quad \widehat{\$}^a_{hU} \quad \widehat{\$}^a_{hV} \quad \widehat{\$}^a_{hW} \quad \widehat{\$}^a_{iR} \quad \widehat{\$}^a_{iS} \dots \\
& \dots \widehat{\$}^a_{iT} \quad \widehat{\$}^a_{iV} \quad \widehat{\$}^a_{iW} \quad \widehat{\$}^a_{jR} \quad \widehat{\$}^a_{jS} \quad \widehat{\$}^a_{jU} \quad \widehat{\$}^a_{jV} \quad \widehat{\$}^a_{jW} \Big] \qquad (5.9)
\end{aligned}
$$

The coupling graph with cut-sets is shown in Figure 30(a). The edges a, d and j were chosen as chords, the other seven edges are branches which correspond to the seven cut-sets. The graph is employed to arrange the cut-sets matrix $[Q]_{6,40}$. To create the action graph the coupling graph edges must be replaced by fifty edges arranged by five-by-five edges in parallel.

Figure 29 – Coupling graph with cut-sets and position points.



(a)

(b)

The network unit action matrix $[\widehat{A}_N]_{42,50}$ is given by combining the matrices $[\widehat{A}_D]_{6,50}$ and $[Q]_{7,50}$ according to Equation 2.13. The rank of $[\widehat{A}_N]_{42,50}$ is 41. Now, it is possible to evaluate the number of redundant constraints of the mechanism:

$$C_N = C - rank([\widehat{A}_N]_{6*7,50}) = 50 - 41 = 9 \qquad (5.10)$$

Therefore, the backrest adjustment mechanism from the LAR bed has nine redundant constraints. The mobility of the mechanism is calculated by Equation 2.15:

$$F_N = \lambda(n - j - 1) + \sum_{i=1}^{j} f_i + C_N = 6(8 - 10 - 1) + 10 + 9 = 1 \qquad (5.11)$$

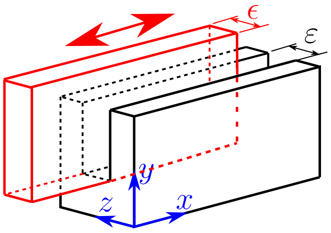
The dual matroid $\mathcal{M}^*{}_{AN}$ derived from the matrix $[\widehat{A}_N]_{42,50}$ is created. As the seed mechanism has nine redundant constraints, the cobases of the set $\mathcal{B}^*$ has cardinality equal to nine. The dual matroid has 773,212 cobases.

The cobases binary matrix $[N]_{773212,50}$ is arranged by Equation 4.2. Some design requirements were defined to select a set of self-aligning mechanisms. The scope of the design requirements is to remove the redundant constraints employing specific clearances to some joints.

In this work, the clearances are understood as mobilities which are imposed to the joints and are not the main mobility of the joint. The usage of clearances can be a low cost possibility to remove redundant constraints, because the manufacturing tolerances can be increased, reducing the manufacturing complexity and the manufacturing cost.

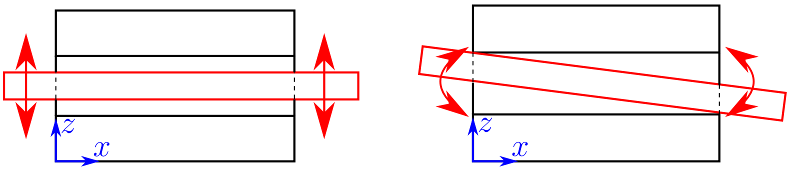
Some clearances conditions are imposed in this case study. Firstly, the joint g, defined as a prismatic joint along the x-axis is constructed by a sheet which slides inside of a linear cavity, as shown in Figure 30

Figure 30 – Joint g: Prismatic along x-axis.



Theoretically, this pair has one mobility, but considering that the thickness ϵ of the metallic sheet is smaller than the cavity thickness ε, two more mobilities are imposed by clearances in this joint, a translational and an angular mobilities, as shown in Figure 31.

Figure 31 – Mobilities imposed by clearances in the joint g.



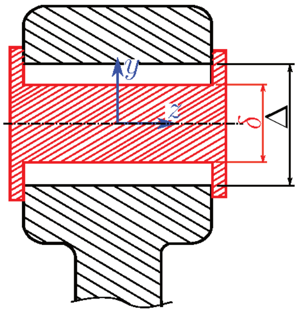
(a) Translational mobility

(b) Angular mobility

This prismatic joint can now be considered as a planar pair by the imposing of the thickness clearance. This planar pair has three mobilities, a rotational around y-axis and two translational along the axes x and z.

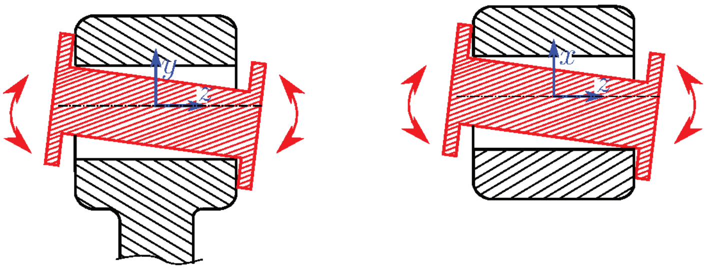
Clearances can be also imposed in revolute joints, as radial or axial clearances. The radial clearance is imposed when the hole diameter Δ is bigger than the pin diameter δ, according to Figure 32.

Figure 32 – Radial clearance in a revolute joint.



A revolute pair without clearances has one angular mobility, but considering a radial clearance imposed to the joint can be considered as a spherical pair, with two more angular mobilities (RESHETOV, 1982). These mobilities derived from radial clearance are shown in Figure 33.

Figure 33 – Mobilities derived from radial clearance.



(a) Angular mobility around x-axis

(b) Angular mobility around y-axis

The axial clearance is imposed when the length of the pin is longer than the length of the hole, as shown in Figure 34. The difference between dimensions increases the joint mobility, this extra mobility is a translational freedom along the rotation axis, then this joint can be considered as a cylindrical pair.

After discussing the clearances considerations, the design requirements of this study case are listed.

1. The joints h, i and j may have the structure CPR or RPC;

2. The prismatic joint g, due to clearances, can be considered as a planar pair;

3. The other revolute joints a, b, c, d, e and f may be considered as spherical or cylindrical joints, due to clearances.

The reason for the design requirement (i) is the axial clearance allowed to one of the two revolute joints of the RPR structure. This consideration was presented previously in Section 5.1.2

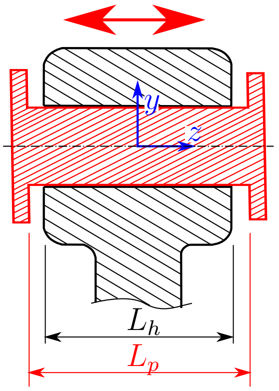
The design requirement (ii) is imposed because a thickness clearance can be added to the prismatic joint g, Figures 30 and 31. The design requirement (iii) is imposed on the remaining revolute joints, radial or axial clearance can be added in these pairs. If the radial clearance is used then the revolute joint is considered as a spherical joint, Figures 32 and 33; if the axial clearance is used then the revolute joint is considered as a cylindrical pair, Figure 34; while if no clearance is applied the joint remain as a revolute joint.

The design requirements are then transformed into the selection criteria presented in Table 9 and 10. The constraint sets for each criterion are listed with the respective conditions, the constraints are represented by the matrix elements $n(i,j)$. Remembering that i is the i^{th} row of $[N]_{773212,50}$, and each row represents a different self-aligning mechanism and j is associated to the j^{th} constraint, whose number follows the arrangement of matrix $[\widehat{A}_D]_{6,50}$ (Equation 5.9).

The criteria listed in Tables 9 and 10 are used to create logical sentences to select the bases that comply with the design requirements (i), (ii) and (iii). These criteria are then applied on the cobases binary matrix $[N]_{773212,50}$.

The criterion K_1 focuses on removing the constraints $\$^a_{hW}$ or $\$^a_{jW}$ from the seed mechanism, creating a structure CPR or RPC. This criterion created a set with 14.472 cobases, corresponding to $1,87\%$ of the entire dual set $\mathcal{B}^*$. The criterion K_2 considered

Figure 34 – Axial clearance in a revolute joint.



the prismatic joint g as a planar joint, due to clearances, so the constraints $\$^a_{gS}$ and $\$^a_{gW}$ are removed from the seed mechanism. The criterion K_2 selected 25.198 cobases, that corresponds to $3,26\%$ of the entire set $\mathcal{B}^*$.

Finally, the criterion K_3 can add two kinds of clearances to the revolute joints a, b, c, d, e and f, radial or axial, considering then the joints as spherical or cylindrical joints, respectively. This criterion selected 13.024 cobases, corresponding to $1,68\%$ of the entire set $\mathcal{B}^*$.

By the intersection among the three criteria, it is possible to define the final subset $K_F = K_1 \cap K_2 \cap K_3$ that represents the cobases which satisfy all the criteria. K_F is composed of eight cobases, corresponding to 0,0000103% of the entire set $\mathcal{B}^*$. Thus, the backrest seed mechanism of LAR hospital bed has eight self-aligning mechanisms that satisfy the design requirements proposed. Figure 35 exemplifies a solution to the overconstrained seed mechanism.

Table 9 – Sets of constraints and respective binary conditions.

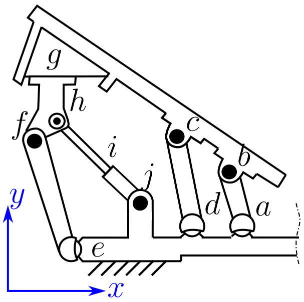
Criterion 1 - K_1			
Constraints	Condition	Constraints	Condition
$\{\$^a_{hR}, \$^a_{hS}, \$^a_{hU}, \$^a_{hV}, \$^a_{iR}, \$^a_{iS}, \$^a_{iT}, \$^a_{iV}, \$^a_{iW}, \$^a_{jR}, \$^a_{jS}, \$^a_{jU}\ \$^a_{jV}, \$^a_{jW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{hW}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{hR}, \$^a_{hS}, \$^a_{hU}, \$^a_{hV}, \$^a_{hW}, \$^a_{iR}, \$^a_{iS}, \$^a_{iT}, \$^a_{iV}, \$^a_{iW}, \$^a_{jR}, \$^a_{jS}, \$^a_{jU}, \$^a_{jV}\}$	$n_{(i,j)} = 0$	$\{\$^a_{jW}\}$	$n_{(i,j)} = 1$
Criterion 2 - K_2			
Contraints		Condition	
$\{\ \$^a_{gR}, \$^a_{gT}, \$^a_{gV}\}$		$n_{(i,j)} = 0$	
AND			
$\{\ \$^a_{gS},\quad \$^a_{gW}\}$		$n_{(i,j)} = 1$	
Criterion 3 - K_3			
Constraints	Condition	Constraints	Condition
$\{\$^a_{aR}, \$^a_{aS}, \$^a_{aU}, \$^a_{aV}\}$	$n_{(i,j)} = 0$	$\{\$^a_{aW}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{aU}, \$^a_{aV} \$^a_{aW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{aR}, \$^a_{aS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{aR}, \$^a_{aS}, \$^a_{aU}, \$^a_{aV}, \$^a_{aW}\}$		$n_{(i,j)} = 0$	

AND... (continue in Table 10)

Table 10 – Sets of constraints and respective binary conditions.

... AND (continuing Table 9)			
$\{\$^a_{bR},\ \$^a_{bS},\ \$^a_{bU},\ \$^a_{bV}\}$	$n_{(i,j)} = 0$	$\{\$^a_{bW}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{bU},\ \$^a_{bV}\ \$^a_{bW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{bR},\ \$^a_{bS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{bR},\ \$^a_{bS},\ \$^a_{bU},\ \$^a_{bV},\ \$^a_{bW}\}$	$n_{(i,j)} = 0$		
AND			
$\{\$^a_{cR},\ \$^a_{cS},\ \$^a_{cU},\ \$^a_{cV}\}$	$n_{(i,j)} = 0$	$\{\$^a_{cW}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{cU},\ \$^a_{cV}\ \$^a_{cW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{cR},\ \$^a_{cS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{cR},\ \$^a_{cS},\ \$^a_{cU},\ \$^a_{cV},\ \$^a_{cW}\}$	$n_{(i,j)} = 0$		
AND			
$\{\$^a_{dR},\ \$^a_{dS},\ \$^a_{cU},\ \$^a_{dV}\}$	$n_{(i,j)} = 0$	$\{\$^a_{dW}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{dU},\ \$^a_{dV}\ \$^a_{dW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{dR},\ \$^a_{dS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{dR},\ \$^a_{dS},\ \$^a_{dU},\ \$^a_{dV},\ \$^a_{dW}\}$	$n_{(i,j)} = 0$		
AND			
$\{\$^a_{eR},\ \$^a_{eS},\ \$^a_{eU},\ \$^a_{eV}\}$	$n_{(i,j)} = 0$	$\{\$^a_{eW}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{eU},\ \$^a_{eV}\ \$^a_{eW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{eR},\ \$^a_{eS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{eR},\ \$^a_{eS},\ \$^a_{eU},\ \$^a_{eV},\ \$^a_{eW}\}$	$n_{(i,j)} = 0$		
AND			
$\{\$^a_{fR},\ \$^a_{fS},\ \$^a_{fU},\ \$^a_{fV}\}$	$n_{(i,j)} = 0$	$\{\$^a_{fW}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{fU},\ \$^a_{fV}\ \$^a_{fW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{fR},\ \$^a_{fS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{fR},\ \$^a_{fS},\ \$^a_{fU},\ \$^a_{fV},\ \$^a_{fW}\}$	$n_{(i,j)} = 0$		

Figure 35 – New concept of a self-aligning backrest mechanism.



The new self-aligning mechanism has the joints a, d and e as spherical, the joint h as cylindrical and the joint g is a planar joint. The changing of the seed mechanism to the selected self-aligning mechanism can be made by imposing specific clearances to the system joints.

5.2.2 Case IV: Leg rest adjustment mechanism

The leg rest adjustment mechanism of the LAR Hospital Bed is used as seed mechanism in this subsection. As in the backrest adjustment mechanism, the self-aligning design requirements are aiming to remove the redundant constraint by the application of specific clearances in the joints.

A structural representation of the leg rest adjustment mechanism is shown in Figure 36. The mechanism has eight joints and seven links. The type, the position vector and the wrenches of each joint are shown in Table 11.

Figure 36 – Leg rest adjustment mechanism from LAR.

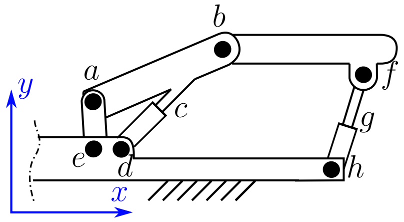


Table 11 – Type of the joints and respective wrenches from LAR hospital bed.

Joint	**Type**	**Position vector $\$_o$**	**Wrenches $\widehat{\$}^a_{ij}$**
a	revolute z	$[0 \quad 3 \quad 0]$	$\widehat{\$}^a_{aR}$ $\widehat{\$}^a_{aS}$ $\widehat{\$}^a_{aU}$ $\widehat{\$}^a_{aV}$ $\widehat{\$}^a_{aW}$
b	revolute z	$[10 \quad 7 \quad 0]$	$\widehat{\$}^a_{bR}$ $\widehat{\$}^a_{bS}$ $\widehat{\$}^a_{bU}$ $\widehat{\$}^a_{bV}$ $\widehat{\$}^a_{bW}$
c	prismatic 45^o	$[5 \quad 3 \quad 0]$	$\widehat{\$}^a_{cR}$ $\widehat{\$}^a_{cS}$ $\widehat{\$}^a_{cU}$ $\widehat{\$}^a_{cV}$ $\widehat{\$}^a_{cW}$
d	revolute z	$[2 \quad 0 \quad 0]$	$\widehat{\$}^a_{dR}$ $\widehat{\$}^a_{dS}$ $\widehat{\$}^a_{dT}$ $\widehat{\$}^a_{dV}$ $\widehat{\$}^a_{dW}$
e	revolute z	$[0 \quad 0 \quad 0]$	$\widehat{\$}^a_{eR}$ $\widehat{\$}^a_{eS}$ $\widehat{\$}^a_{eU}$ $\widehat{\$}^a_{eV}$ $\widehat{\$}^a_{eW}$
f	revolute z	$[21 \quad 5 \quad 0]$	$\widehat{\$}^a_{fR}$ $\widehat{\$}^a_{fS}$ $\widehat{\$}^a_{fU}$ $\widehat{\$}^a_{fV}$ $\widehat{\$}^a_{fW}$
g	prismatic 45^o	$[20 \quad 2 \quad 0]$	$\widehat{\$}^a_{gR}$ $\widehat{\$}^a_{gS}$ $\widehat{\$}^a_{gU}$ $\widehat{\$}^a_{gV}$ $\widehat{\$}^a_{gW}$
h	revolute z	$[18 \quad -2 \quad 0]$	$\widehat{\$}^a_{hR}$ $\widehat{\$}^a_{hS}$ $\widehat{\$}^a_{hU}$ $\widehat{\$}^a_{hV}$ $\widehat{\$}^a_{hW}$

The wrenches are established according to the position points shown in Figure 38(b) and Table 11, they are organized into the matrix $[\widehat{A}_D]_{6,40}$ as follows:

$$\begin{aligned}[\widehat{A}_D]_{6,40} = \Big[&\widehat{\$}^a_{aR} \quad \widehat{\$}^a_{aS} \quad \widehat{\$}^a_{aU} \quad \widehat{\$}^a_{aV} \quad \widehat{\$}^a_{aW} \quad \widehat{\$}^a_{bR} \quad \widehat{\$}^a_{bS} \quad \widehat{\$}^a_{bU} \quad \widehat{\$}^a_{bV} \ldots \\ &\ldots \widehat{\$}^a_{bW} \quad \widehat{\$}^a_{cR} \quad \widehat{\$}^a_{cS} \quad \widehat{\$}^a_{cT} \quad \widehat{\$}^a_{cV} \quad \widehat{\$}^a_{cW} \quad \widehat{\$}^a_{dR} \quad \widehat{\$}^a_{dS} \quad \widehat{\$}^a_{dU} \quad \widehat{\$}^a_{dV} \quad \widehat{\$}^a_{dW} \ldots \\ &\ldots \widehat{\$}^a_{eR} \quad \widehat{\$}^a_{eS} \quad \widehat{\$}^a_{eU} \quad \widehat{\$}^a_{eV} \quad \widehat{\$}^a_{eW} \quad \widehat{\$}^a_{fR} \quad \widehat{\$}^a_{fS} \quad \widehat{\$}^a_{fU} \quad \widehat{\$}^a_{fV} \quad \widehat{\$}^a_{fW} \quad \widehat{\$}^a_{gR} \ldots \\ &\ldots \widehat{\$}^a_{gS} \quad \widehat{\$}^a_{gT} \quad \widehat{\$}^a_{gV} \quad \widehat{\$}^a_{gW} \quad \widehat{\$}^a_{hR} \quad \widehat{\$}^a_{hS} \quad \widehat{\$}^a_{hU} \quad \widehat{\$}^a_{hV} \quad \widehat{\$}^a_{hW}\Big] \end{aligned} \tag{5.12}$$

The coupling graph with cut-sets is shown in Figure 37. Edges a and b were chosen as chords, the other six edges are branches which correspond to six cut-sets. This graph is used to arrange the cut-sets matrix $[Q]_{6,40}$. To create the action graph, the edges of the coupling graph must be replaced by forty edges, arranged five-by-five in parallel.

The matrices $[\widehat{A}_D]_{6,40}$ and $[Q]_{6,40}$ are combined according to Equation 2.13 to arrange the network unit action matrix $[\widehat{A}_N]_{36,40}$. The rank of $[\widehat{A}_N]_{36,40}$ is 34. According to Equation 3.7, it is possible to evaluate the number of redundant constraints of the seed mechanism:

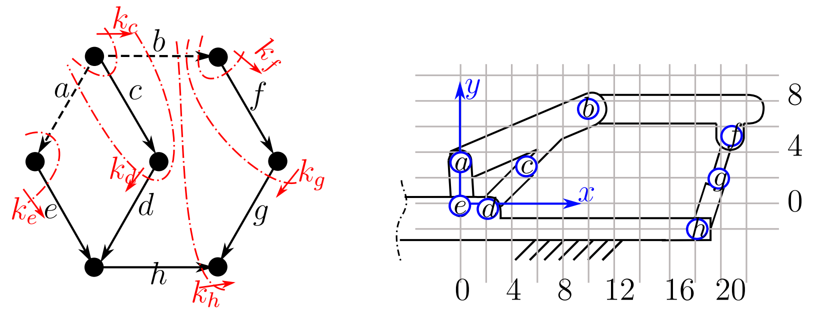
$$C_N = C - rank([\widehat{A}_N]_{6*6,40}) = 40 - 34 = 6 \tag{5.13}$$

According to Equation 5.13 the leg rest adjustment mechanism has six redundant constraints. Now, the mechanism mobility is evaluated by Equation 2.15:

$$F_N = \lambda(n - j - 1) + \sum_{i=1}^{j} f_i + C_N = 6(7 - 8 - 1) + 8 + 6 = 2 \tag{5.14}$$

The dual matroid $\mathcal{M}^*{}_{AN}$ is created from the matrix $[\widehat{A}_N]_{36,40}$. As the seed mechanism has six redundant constraints, the cobases of $\mathcal{B}^*$ have cardinality equal to six. The dual matroid $\mathcal{M}^*_{AN}$ has 21.988 cobases.

Figure 37 – Coupling graph with cut-sets and position points.



(a)

(b)

The cobases binary matrix $[N]_{21988,40}$ is then arranged by Equation 4.2. The design requirements herein applied follows the same scope of case 3 (section 5.2.1), to allow specific clearances to some joints. Therefore, three design requirements are established:

1. The actuated joint c cannot be modified ;
2. The joints f, g and h may have the structure CPR or RPC;
3. The other revolute joints a b, d and e may be considered as spherical or cylindrical joints, due to clearances.

The design requirement (i) was defined because the prismatic actuator is a commercial actuator, and in this study case, the inclusion of more mobilities in the actuator is not considered. The reason for design requirement (ii) is the axial clearance allowed to one of two revolute joints of the RPR structure. The design requirement (iii) is imposed on the remaining revolute joints, as radial or axial clearance can be added in these pairs. The considerations for design requirements (ii) and (iii) were presented in Section 5.1.2.

These design requirements are then transformed into selection criteria which evaluate the rows of the cobases binary matrix $[N]_{21988,40}$. The columns matrix $[N]_{21988,40}$ are ordered according to the matrix $[\widehat{A}_D]_{36,40}$ (Equation 5.12). The conditions for the design requirements (i) and (ii) are shown in Table 12, while the conditions for the design requirement (iii) is shown in Table 13.

Table 12 – Set of constraints and respective binary conditions.

<table>
<tr><th colspan="4"><b>Criterion 1 -</b> K_1</th></tr>
<tr><td colspan="2">Contraints</td><td colspan="2">Condition</td></tr>
<tr><td colspan="2">$\{\ \$^a_{cR},\quad \$^a_{cS},\quad \$^a_{cT},\quad \$^a_{cV},\quad \$^a_{cW}\}$</td><td colspan="2">$n_{(i,j)} = 0$</td></tr>
<tr><td colspan="4"><b>Criterion 2 -</b> K_2</td></tr>
<tr><td>Constraints</td><td>Condition</td><td>Constraints</td><td>Condition</td></tr>
<tr><td>$\{\$^a_{fR}, \$^a_{fS}, \$^a_{fU}, \$^a_{fV}, \$^a_{gR}, \$^a_{gS}, \$^a_{gT}, \$^a_{gV}, \$^a_{gW}\$^a_{hR}, \$^a_{hS}, \$^a_{hU}, \$^a_{hV}, \$^a_{hW}\}$</td><td>$n_{(i,j)} = 0$</td><td>$\{\$^a_{fW}\}$</td><td>$n_{(i,j)} = 1$</td></tr>
<tr><td colspan="4">OR</td></tr>
<tr><td>$\{\$^a_{fR}, \$^a_{fS}, \$^a_{fU}, \$^a_{fV}, \$^a_{fW}, \$^a_{gR}, \$^a_{gS}, \$^a_{gT}, \$^a_{gV}, \$^a_{gW}, \$^a_{hR}, \$^a_{hS}, \$^a_{hU}, \$^a_{hV}\}$</td><td>$n_{(i,j)} = 0$</td><td>$\{\$^a_{hW}\}$</td><td>$n_{(i,j)} = 1$</td></tr>
</table>

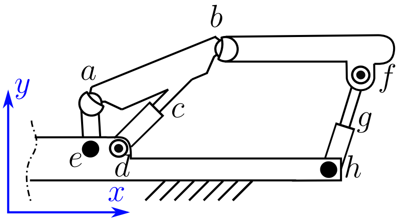
Criterion 1 created a set K_1 with 6,752 cobases, corresponding to 30.7% of the entire set $\mathcal{B}^*$. Criterion 2 created a set K_2 with 768 cobases, corresponding to 3.49% of the entire set $\mathcal{B}^*$. Criterion 3 created a set with 1,090 cobases, corresponding to 4.95% of the entire set $\mathcal{B}^*$.

By the intersection among the criteria, it is possible to define the final subset $K_F = K_1 \cap K_2 \cap K_3$ that represents the cobases which satisfy all the criteria. K_F is composed of four cobases, corresponding to 0.018% on the entire set $\mathcal{B}^*$, *i.e.* the leg rest adjustment mechanism of LAR hospital bed has four self-aligning mechanisms that satisfy the design requirements proposed. Figure 38 exemplifies a solution for the overconstrained seed mechanism.

Table 13 – Set of constraints and respective binary conditions.

Criterion 3 - K_3			
Constraints $\{\$^a_{aR}, \quad \$^a_{aS}, \quad \$^a_{aU}, \quad \$^a_{aV}\}$	Condition $n_{(i,j)} = 0$	Constraints $\{\$^a_{aW}\}$	Condition $n_{(i,j)} = 1$
OR			
$\{\$^a_{aU}, \quad \$^a_{aV} \quad \$^a_{aW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{aR}, \quad \$^a_{aS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{aR}, \quad \$^a_{aS}, \quad \$^a_{aU}, \quad \$^a_{aV}, \quad \$^a_{aW}\}$		$n_{(i,j)} = 0$	
AND			
$\{\$^a_{bR}, \quad \$^a_{bS}, \quad \$^a_{bU}, \quad \$^a_{bV}\}$	$n_{(i,j)} = 0$	$\{\$^a_{bW}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{bU}, \quad \$^a_{bV} \quad \$^a_{bW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{bR}, \quad \$^a_{bS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{bR}, \quad \$^a_{bS}, \quad \$^a_{bU}, \quad \$^a_{bV}, \quad \$^a_{bW}\}$		$n_{(i,j)} = 0$	
AND			
$\{\$^a_{dR}, \quad \$^a_{dS}, \quad \$^a_{cU}, \quad \$^a_{dV}\}$	$n_{(i,j)} = 0$	$\{\$^a_{dW}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{dU}, \quad \$^a_{dV} \quad \$^a_{dW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{dR}, \quad \$^a_{dS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{dR}, \quad \$^a_{dS}, \quad \$^a_{dU}, \quad \$^a_{dV}, \quad \$^a_{dW}\}$		$n_{(i,j)} = 0$	
AND			
$\{\$^a_{eR}, \quad \$^a_{eS}, \quad \$^a_{eU}, \quad \$^a_{eV}\}$	$n_{(i,j)} = 0$	$\{\$^a_{eW}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{eU}, \quad \$^a_{eV} \quad \$^a_{eW}\}$	$n_{(i,j)} = 0$	$\{\$^a_{eR}, \quad \$^a_{eS}\}$	$n_{(i,j)} = 1$
OR			
$\{\$^a_{eR}, \quad \$^a_{eS}, \quad \$^a_{eU}, \quad \$^a_{eV}, \quad \$^a_{eW}\}$		$n_{(i,j)} = 0$	

Figure 38 – New concept of a self-aligning leg rest mechanism.



The new self-aligning mechanism has the joints a and b as spherical, the joints d and f as cylindrical. The other joints keep the constraints of the seed mechanism. This mechanism is complying with the design requirements established.

6 CONCLUSION

This work dealt with mechanisms, in particular with self-aligning mechanisms. Self-aligning mechanisms do not have redundant constraints, implying special characteristics, such as facilitating the designing, manufacturing and assembly processes.

In order to analyze the structure of the mechanisms and the presence of redundant constraints, Davies' method was reviewed. Davies' method is important to the proposed method because the network unit action matrix $[\widehat{A}_N]$ is used to examine the linear dependence among the matrix columns. If the rank of $[\widehat{A}_N]$ is smaller than the number of columns, the mechanisms have redundant constraints, remembering the number of columns is the number of constraints.

Concepts of Matroid theory were introduced using a graph as an example, then they were applied to matrices aiming to create sets of columns which are linearly independent among them. Considering an overconstrained mechanism modeled statically by Davies' method, matroid creates sets of columns which are linearly independent. These sets are related to derived mechanisms that do not have redundant constraints, *i.e* self-aligning mechanisms. To exemplify it, a four-bar mechanism was modeled by Davies' method and to create all possibilities of self-aligning mechanisms, Matroid theory was then applied to the $[\widehat{A}_N]$ matrix.

Greedy algorithm was discussed and applied to select self-aligning mechanisms derived from the four-bar mechanism. The example was useful to show how assigning weights to the constraints is a hard task for designers.

The method for selecting a set of self-aligning mechanisms proposed in this work was introduced in Chapter 4. The designer needs a seed overconstrained mechanism modeled statically by Davies' method, then Matroid theory is applied to create the cobases of the dual matroid $\mathcal{M}^*_{AN}$. The proposed method can be applied from the cobases enumerated by $\mathcal{M}_{AN}$.

The cobases are organized into the cobases binary matrix $[N]$, then the design requirements are converted into selection criteria. Each selection criterion i creates a subset K_i which contains the cobases representing the mechanisms that comply with criterion i. The intersection among the subsets of all criteria will generate a final subset, K_F. All the self-aligning mechanisms which comply with all design requirements are included in K_F.

Moreover, some case studies were presented in order to exemplify the process of selection. The first case study was the backrest adjustment mechanism of Linet Eleganza 3XC Hospital Bed. In this case, the first set of design requirements created a empty final subset K_F, so the design requirements were changed creating a final subset K_F with ten self-aligning mechanisms which comply with the design requirements proposed. Three other study cases were presented with different design requirements, hence different selection criteria were defined. In all cases, the final subsets contain a low number of self-aligning mechanisms. The low number of self-aligning mechanisms present in K_F facilitates the designer choice.

It is important to state the design requirements were defined according to mechanism theory. So, the selected self-aligning mechanisms present in K_F have similar characteristics. For the next steps, it would be desirable to include other engineering details such as stiffness analysis which should be performed into the mechanisms present in subsets K_F and the results should be compared in order to elect the self-aligning mechanism with the biggest stiffness among the selected by the proposed selection method.

In some studies case, clearances were added in some joints, so pose error analysis can be performed to the selected self-aligning mechanisms with lower error.

REFERENCES

BALL, R. **A Treatise on the Theory of Screws**, 1900. Cambridge University Press, Cambridge, UK, 1900.

BALL, R. S. **A Treatise on the Theory of Screws**. Cambridge university press, 1998.

BARRETO, R. L.; MARTINS, D.; SIMONI, R.; DAI, J. S. et al. Self-aligning analysis of the metamorphic palm of the kcl/tju metamorphic hand. In: **IEEE. 2018 International Conference on Reconfigurable Mechanisms and Robots (ReMAR)**. 2018. p. 1–8.

BLANDING, D. **Exact Constraint: Machine Design Using Kinematic Principles**. 1999. ASME Press, 1999.

CARBONI, A. P. **Análise de mecanismos com restrições redundantes através da aplicação da teoria de matroides**. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-graduação em Engenharia Mecânica Florianópolis, SC, 2015 .

CARBONI, A. P.; SIMAS, H.; MARTINS, D. Analysis of self-aligning mechanisms by means of matroid theory. In: **SPRINGER. International Symposium on Multibody Systems and Mechatronics**. 2017. p. 61–73.

CAZANGI, H. R. **Aplicação do método de davies para análise cinemática e estática de mecanismos com múltiplos graus de liberdade**. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-graduação em Engenharia Mecânica Florianópolis, SC, 2008.

CECCARELLI, M. Screw axis defined by giulio mozzi in 1763 and early studies on helicoidal motion. **Mechanism and Machine Theory**, Elsevier, v. 35, n. 6, p. 761–770, 2000.

CHASLES, M. Note sur les propriétés générales du système de deux corps semblables entr'eux et placés d'une manière quelconque dans l'espace; et sur le déplacement fini ou infiniment petit d'un corps solide libre. **Bulletin des Sciences Mathématiques**, Férussac, v. 14, p. 321–26, 1830.

CHEBYCHEV, P. L. Théorie des mécanismes connus sous le nom de parallélogrammes. **Imprimerie de l'Académie impériale des sciences**, 1853.

DAVIES, T. 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, 2006.

GALLARDO-ALVARADO, J. **Kinematic analysis of parallel manipulators by algebraic screw theory**. Springer, 2016.

GOGU, G. Chebychev–grübler–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, 2005.

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

HARTENBERG, R.; DENAVIT, J. **Kinematic synthesis of linkages**. New York: McGraw-Hill, 1964.

HUANG, Z.; LIU, J.; ZENG, D. A general methodology for mobility analysis of mechanisms based on constraint screw theory. **Science in China Series E: Technological Sciences**, Springer, v. 52, n. 5, p.1337–1347, 2009.

HUNT, K. H. **Kinematic geometry of mechanisms**. Oxford University Press, USA, 1978.

LAUS, L. P. **Determinação da eficiência de máquinas com base em teoria de helicoides e grafos: aplicação em trens de engrenagens e robôs paralelos**. Tese (Doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-graduação em Engenharia Mecânica Florianópolis, SC, 2011.

LINET BRASIL. Eleganza 3XC LINET. 2019.<http://www.linetbrasil.com/pt-BR/hospitalar/camas/camas-de-cuidado-intensivo/eleganza-3xc>.

MALETZ, E. R. **Critérios funcionais e estruturais para a síntese de mecanismos para movimentar pacientes acamados**. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-graduação em Engenharia Mecânica Florianópolis, SC, 2017.

MANENTI, V. C. **Estudo de caso de mecanismos de direção para semirreboques com eixos distanciados**. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-graduação em Engenharia Mecânica Florianópolis, SC, 2018.

MOZZI, G. **Discorso matematico sopra il rotamento momentaneo dei corpi**. Donate Campo, 1763.

MURAI, E. H. **Projeto de mecanismos de costura com acesso unilateral usando síntese do número e do tipo**. Dissertação (Mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-graduação em Engenharia Mecânica Florianópolis, SC, 2013.

RECSKI, A. **Matroid theory and its applications in electric network theory and in statics**. Springer Science & Business Media, 2013.

RESHETOV, L. N. **Self-aligning mechanisms**. Imported Pubn, 1982.

TRIDEC. TRIDEC Sales information - Mechanical Steering System TR [0401]. 2016. <https://www.jostinformationcentre.com/static/upload/pdf/kataloge/TRIDEC_SI_TR_EN_20160331.pdf>.

TSAI, L.-W. **Mechanism design: enumeration of kinematic structures according to function**. CRC press, 2000.

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

WEIHMANN, L. **Modelagem e otimização de forças e torques aplicados por robôs com redundância cinemática e de atuação em contato com o meio.** Tese (Doutorado) - Universidade Federal de Santa Catarina , Centro Tecnológico. Programa de Pós-graduação em Engenharia Mecânica, Florianópolis, SC, 2011.

WHITNEY, D. E. **Mechanical assemblies: their design, manufacture, and role in product development**. Oxford university press New York, 2004.

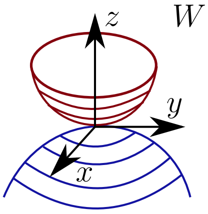
YAN, H.-S. **Creative design of mechanical devices**. Springer Science & Business Media, 1998.

APPENDIX A – CLASSES DETERMINATION OF KINEMATIC PAIRS

For the correct mobility analysis of a mechanism is necessary that the couplings are correctly determined. An object in frees-state has six mobilities, when a coupling is made between the object and a reference object in such a way that the number of degrees of freedom of the object is reduced, it means that the object has been constrained (BLANDING, 1999).

The characteristics of the coupling determines which freedoms are substituted by constraints, so a coupling can be determined by the freedoms allowed or by the constraints imposed by the coupling. Reshetov (1982) presented a classification of most used couplings. The classification has five classes denoted with roman numerals I, II, III, IV and V, the couplings of each class have the same number of constraints.

Figure 39 – Point pair - Class I.



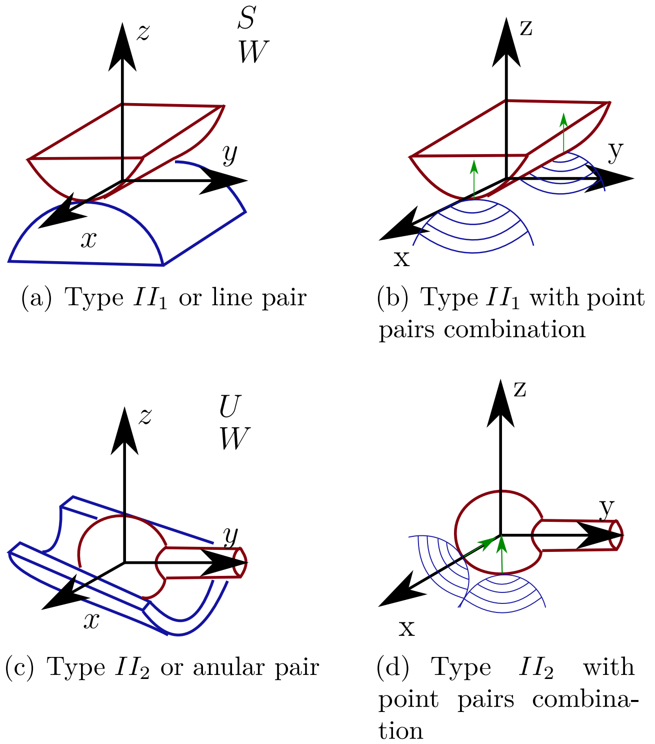
The simplest coupling is the point pair shown in Figure 39, as this coupling has one constraint it is classified as Class I, this coupling opposes relative displacement and transmits a force directed along the normal line of working surface. For example, the coupling shown in Figure 39 imposes the contraint W which is related to the linear displacement along the axis z.

As the point pair has one constraint consequentely it has five mobilities. The combination of point pairs can be converted into all known couplings, in general, to obtain a coupling of any class the component couplings must be connected in parallel, it will become clear by the introducing of the others classes.

To obtain a coupling of Class II, which has two constraints imposed, two pairs of Class I must be combined. Two kinds of second-class couplings are possible of be created by the combination of two point pairs and are shown in Figure 40.

The coupling shown in Figure 41(a) has two cilindrical bodies as example, the contact between the bodies is a line, so this coupling is called of line pair. The line pair can be

Figure 40 – Couplings of Class II.



(a) Type II_1 or line pair

(b) Type II_1 with point pairs combination

(c) Type II_2 or anular pair

(d) Type II_2 with point pairs combination

made by a specif combination of two point pairs, the Figure 41(b) shows that the point pairs must be arranged in a line and the normal forces must be parallel, in this case the normal forces are parallel to the z-axis.

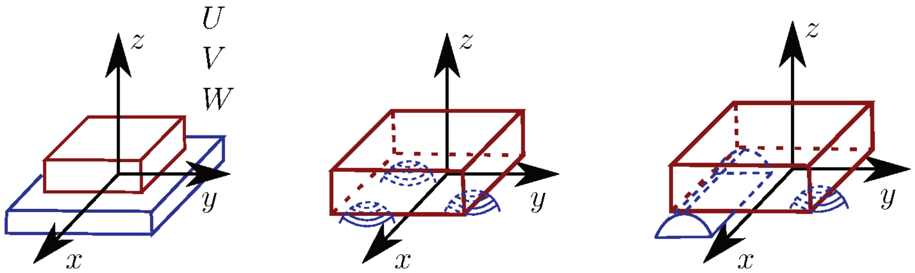
The line pair imposes two constraints, S and W, the first is the rotation around y-axis and the second is the translation along the z-axis, it can be considered as a moment around y-axis and a force along the z-axis, respectively.

The coupling shown in Figure 41(c) can be considered as a sphere in a pipe and can be made by the combination of two point pairs, shown in Figure 41(d). The directions of the point pairs are in the same plane and are perpendicular between them.

The two constraints imposed by the coupling type II_2 are U and W, they are translations along the axes x and z, respectively, and can be considered as two forces along the axes x and z, respectively.

An example of coupling of Class III is the planar pair shown in Figure 42(a), it can be considered as a book in a table, where the surface of contact between the bodies is a plane. As all the couplings of Class III, the planar pair has three constraints imposed, in this case, the constraints are two rotational around the axes x and y, U and V respectively, and a tranlational along the z-axis, W. The constraints can be considered as two moments around the axes x and y and a force along the z-axis.

Figure 41 – Planar pair and the combinations of other pairs.

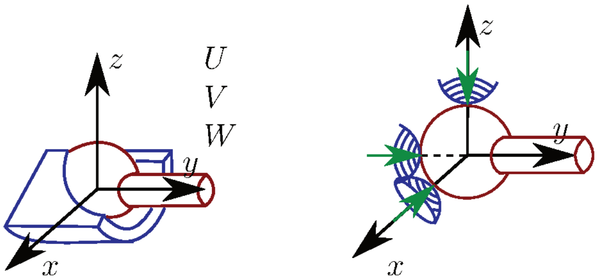


(a) Type III_1 or planar pair (b) Type III_1 with point pairs combination (c) Type III_1 with line and point pair cambined

The planar pair can be made by the combination of three point pairs, Figure 42(b), or by the combination of a line pair and a point pair, Figure 42(c).

Other example of Class III coupling is the spherical pair, Figure 42, which can be made by the combination of three point pairs orthogonal among them, Figure 43(b).

Figure 42 – Spherical pair and the combination of point pairs.



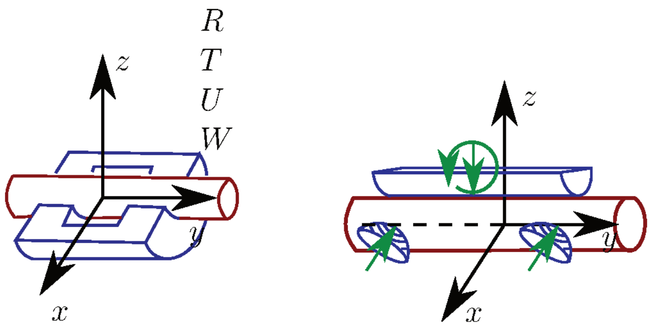
(a) Type III_2 or spherical pair (b) Type III_2 with point pairs combined

The difference between the anular pair, Figure 41(c) and the spherical pair, Figure 42, is the constraint related to the linear displacement along the y-axis. The spherical pair has three translational constraints imposed, U, V and W, they are parallel to the axes x, y and z, respectively.

An example of Class IV coupling is the cylindrical pair, shown in Figure 44(a), this pair has four constraints, two rotational, R and T, around the axes x and z, respectively, and two translational, U and W, along the same axes, respectively. These constraints can be taken as two moments around and two forces along the axes x and z.

A possible combination of lower classes is shown in Figure 44(b), it has a line pair aligned to the y-axis and directed to the z-axis, and two point pairs aligned along the

Figure 43 – Cylindrical pair and the combination of other pairs.

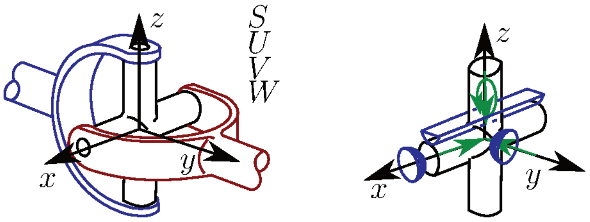


(a) Type IV_1 or cylindrical pair

(b) Type IV_1 with point and line pairs combined

y-axis and directed to the x-axis. By the combination of two point pairs, the cylindrical pair could be the combination of two line pairs.

Figure 44 – Universal pair and the combination of point pairs.



(a) Type IV_2 or universal pair

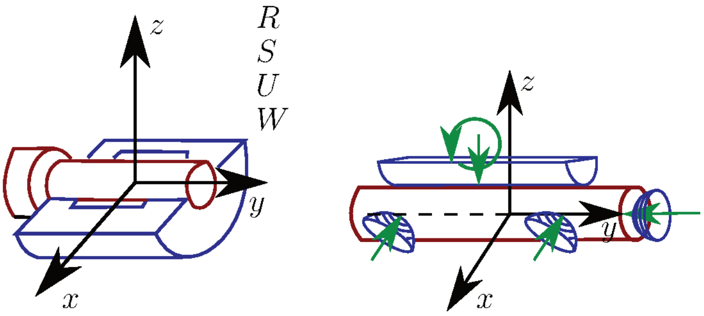
(b) Type IV_2 with point and line pairs combined

Another example of Class IV coupling is the universal pair, shown in Figure 45(a), this pair has four constraints, a rotational, S, around the y-axis and three translational, U, V and W, along the axes x, y and z, respectively. The Figure 45(b) ilustrates a possible combination of lower classes pairs, in this case two point pairs and one line pair, which corresponding to the universal pair.

The constraints of the universal coupling can be understood as a moment around the y-axis and three forces directed along the axes x, y and z.

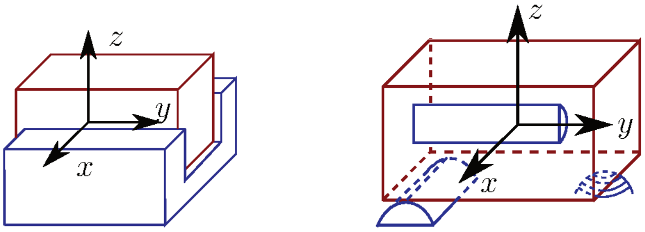
The revolute pair is an example of Class V coupling, Figure 46(a), it has five constraints, three are translational, U, V and W along the axes x, y and z, and two are rotational, R and T around the axes x and y for the given example. These constraints can be interpreted as three forces and two moments. In this case the freedom is the rotation around the y-axis. The Figure 46(b) illustrates a combination of point and line pairs which represents the revolute pair.

Figure 45 – Revolute pair and the combination of other pairs.



(a) Type V_1 or revolute pair (b) Type V_1 with point and line pairs combined

Figure 46 – Prismatic pair and the combination of other pairs.



(a) Type V_2 or prismatic pair (b) Type V_2 with point and line pairs combined

Other type of Class V coupling is the prismatic pair, shown in Figure 47(a), among the five constraints three are rotational, R, S and T, and two are translational, U and W, along the axes x and z, remaining a freedom of translation along the axis y.

The constraints of the prismatic pair can be converted into three moments and two forces. A combination of lower classes pairs that results in the prismatic pair is shown in Figure 47(b)

The couplings presented in this section are usefull to the development of this work, others types can be found in the literature, machines and mechanisms.

APPENDIX A – MATLAB AND SAGEMATH PROGRAMS

The case studies approached in Chapter 5 were implemented in Matlab and this section shows the algorithms used in each case study.

A.1 CASE STUDY I - ELEGANZA'S BACKREST ADJUSTMENT MECHANISM

Mechanism modeled by Davies' method in the pose presented in Table 2.

```
% Constraint analysis of the backrest mechanism of a linet's
robotic hospital bed

clc
clear all

%Points:
a=[7.5 1.5 0];
b=[10 4 0];
c=[12.5 6.5 0];
d=[13.5 -1 0];
e=[0 0 0];
f=[1 4 0];
g=[4 0 0];
h=[5 6.5 0];
i=[14.5 12.5 0];
j=[13 3 0];

% Primal vectors:

R = [1 0 0];
S = [0 1 0];
T = [0 0 1];
U = [1 0 0];
V = [0 1 0];
```

```
W = [0 0 1];

%Wrenches:

%Joint a (rotative in z):
Ra = [R 0 0 0];
Sa = [S 0 0 0];
Ua = [cross(a,U) U];
Va = [cross(a,V) V];
Wa = [cross(a,W) W];
%Joint b(prismatic in y):
Rb = [R 0 0 0];
Sb = [S 0 0 0];
Tb = [T 0 0 0];
Ub = [cross(b,U) U];
Vb = [cross(b,V) V];
Wb = [cross(b,W) W];
%Joint c(rotative in z):
Rc = [R 0 0 0];
Sc = [S 0 0 0];
Uc = [cross(c,U) U];
Vc = [cross(c,V) V];
Wc = [cross(c,W) W];
%Joint d(rotative in z):
Rd = [R 0 0 0];
Sd = [S 0 0 0];
Ud = [cross(d,U) U];
Vd = [cross(d,V) V];
Wd = [cross(d,W) W];
%Joint e(rotative in z):
Re = [R 0 0 0];
Se = [S 0 0 0];
Ue = [cross(e,U) U];
```

```
Ve = [cross(e,V) V];
We = [cross(e,W) W];
%Joint f(rotative in z):
Rf = [R 0 0 0];
Sf = [S 0 0 0];
Uf = [cross(f,U) U];
Vf = [cross(f,V) V];
Wf = [cross(f,W) W];
%Joint g(rotative in z):
Rg = [R 0 0 0];
Sg = [S 0 0 0];
Ug = [cross(g,U) U];
Vg = [cross(g,V) V];
Wg = [cross(g,W) W];
%Joint h(rotative in z):
Rh = [R 0 0 0];
Sh = [S 0 0 0];
Uh = [cross(h,U) U];
Vh = [cross(h,V) V];
Wh = [cross(h,W) W];
%Joint h(rotative in z):
Ri = [R 0 0 0];
Si = [S 0 0 0];
Ui = [cross(i,U) U];
Vi = [cross(i,V) V];
Wi = [cross(i,W) W];
%Joint j(prismtica com 45):
Rj = [R 0 0 0];
Sj = [S 0 0 0];
Tj = [T 0 0 0];
UVj = [cross(j,U) 1/sqrt(2) 1/sqrt(2) 0];
Vj = [cross(j,V) 0 1/sqrt(2) 0];
Wj = [cross(j,W) W];
```

```
%Unit action matrix:

Ad = [Rd' Sd' Ud' Vd' Wd' Re' Se' Ue' Ve' We' Rg' Sg'
 Ug' Vg' Wg' Ra' Sa' Ua' Va' Wa' Rb' Sb' Tb' Ub' Wb'
 Rc' Sc' Uc' Vc' Wc' Rf' Sf' Uf' Vf' Wf' Rh' Sh'
Uh' Vh' Wh' Ri' Si' Ui' Vi' Wi' Rj' Sj' Tj' UVj' Wj'];
% Cut-set matrix:

%    d e g a b c f h i j
Q = [1 1 1 1 0 0 0 0 0 0;
1 1 1 0 1 0 0 0 0 0;
1 1 1 0 0 1 0 0 0 0;
0 -1 0 0 0 0 1 0 0 0;
0 0 -1 0 0 0 0 1 0 0;
0 1 1 0 0 0 0 0 1 0;
0 1 1 0 0 0 0 0 0 1];

Qa = [Q(:,1) Q(:,1) Q(:,1) Q(:,1) Q(:,1) Q(:,2)
Q(:,2) Q(:,2) Q(:,2) Q(:,2) Q(:,3) Q(:,3) Q(:,3)
Q(:,3) Q(:,3) Q(:,4) Q(:,4) Q(:,4) Q(:,4) Q(:,4)
Q(:,5) Q(:,5) Q(:,5) Q(:,5) Q(:,5) Q(:,6) Q(:,6)
Q(:,6) Q(:,6) Q(:,6) Q(:,7) Q(:,7) Q(:,7) Q(:,7)
Q(:,7) Q(:,8) Q(:,8) Q(:,8) Q(:,8) Q(:,8) Q(:,9)
Q(:,9) Q(:,9) Q(:,9) Q(:,9) Q(:,10) Q(:,10) Q(:,10)
Q(:,10) Q(:,10)];

lambda = size(Ad,1);
q = size(Qa,1); %Number of cut-sets
C = size(Ad,2); %Gross deegre of constraint

An = zeros(q*lambda, C);
```

```
for j = 1:1:q

for i = 1:1:C

for k = 1:1:lambda

An((j-1)*lambda+k,i) = Qa(j,i)*Ad(k,i);

end

end

end


Cna = size(Ad,2) - rank(An)


rrefAn = rref(An);


%% Adicionando Atuao em b


Ub = [cross(b,U) U]; %Atuao


%New network unit action matrix:


Ad1 = [Rd' Sd' Ud' Vd' Wd' Re' Se' Ue' Ve' We' Rg' Sg'

Ug' Vg' Wg' Ra' Sa' Ua' Va' Wa' Rb' Sb' Tb' Ub' Vb' Wb'

Rc' Sc' Uc' Vc' Wc' Rf' Sf' Uf' Vf' Wf' Rh' Sh' Uh' Vh'

Wh' Ri' Si' Ui' Vi' Wi' Rj' Sj' Tj' UVj' Wj'];


%New cut-set matrix:


Qa1 = [Q(:,1) Q(:,1) Q(:,1) Q(:,1) Q(:,1) Q(:,2) Q(:,2)

 Q(:,2) Q(:,2) Q(:,2) Q(:,3) Q(:,3) Q(:,3) Q(:,3) Q(:,3)

 Q(:,4) Q(:,4) Q(:,4) Q(:,4) Q(:,4) Q(:,5) Q(:,5) Q(:,5)

 Q(:,5) Q(:,5) Q(:,5) Q(:,6) Q(:,6) Q(:,6) Q(:,6) Q(:,6)

 Q(:,7) Q(:,7) Q(:,7) Q(:,7) Q(:,7) Q(:,8) Q(:,8) Q(:,8)

 Q(:,8) Q(:,8) Q(:,9) Q(:,9) Q(:,9) Q(:,9) Q(:,9) Q(:,10)

 Q(:,10) Q(:,10) Q(:,10) Q(:,10)];
```

```
%lambda = size(Ad,1);
q1 = size(Qa1,1); %Number of cut-sets
C1 = size(Ad1,2); %Gross deegre of constraint

An1 = zeros(q1*lambda, C1);

for j = 1:1:q1
for i = 1:1:C1
for k = 1:1:lambda
An1((j-1)*lambda+k,i) = Qa1(j,i)*Ad1(k,i);
end
end
end

Cna1 = size(Ad1,2) - rank(An1)

rrefAn1 = rref(An1);

%% Par j Prismtico em y
j = [3 4 0];
Uj = [cross(j,U) U];

%New network unit action matrix:
Ad2 = [Rd' Sd' Ud' Vd' Wd' Re' Se' Ue' Ve' We' Rg' Sg'
Ug' Vg' Wg' Ra' Sa' Ua' Va' Wa' Rb' Sb' Tb' Ub' Vb' Wb'
Rc' Sc' Uc' Vc' Wc' Rf' Sf' Uf' Vf' Wf' Rh' Sh' Uh' Vh'
Wh' Ri' Si' Ui' Vi' Wi' Rj' Sj' Tj' Uj' Wj'];

%New cut-set matrix:

Qa2 = [Q(:,1) Q(:,1) Q(:,1) Q(:,1) Q(:,1) Q(:,2) Q(:,2)
Q(:,2) Q(:,2) Q(:,2) Q(:,3) Q(:,3) Q(:,3) Q(:,3) Q(:,3)
Q(:,4) Q(:,4) Q(:,4) Q(:,4) Q(:,4) Q(:,5) Q(:,5) Q(:,5)
Q(:,5) Q(:,5) Q(:,5) Q(:,6) Q(:,6) Q(:,6) Q(:,6) Q(:,6)
Q(:,7) Q(:,7) Q(:,7) Q(:,7) Q(:,7) Q(:,8) Q(:,8) Q(:,8)
Q(:,8) Q(:,8) Q(:,9) Q(:,9) Q(:,9) Q(:,9) Q(:,9) Q(:,10)
Q(:,10) Q(:,10) Q(:,10) Q(:,10)];

%lambda = size(Ad,1);
q2 = size(Qa2,1); %Number of cut-sets
C2 = size(Ad2,2); %Gross deegre of constraint

An2 = zeros(q2*lambda, C2);


for j = 1:1:q2
for i = 1:1:C2
for k = 1:1:lambda
An2((j-1)*lambda+k,i) = Qa2(j,i)*Ad2(k,i);
end
end
end

Cna2 = size(Ad2,2) - rank(An2)

rrefAn2 = rref(An2);
```

Commands applied in the Sagemath to create the matroid $\mathcal{M}_{AN}$ relative to the backrest adjustment mechanism from LINET:

```
COLOCAR
```

Method of selection applied to the backrest adjustment mechanism:

```
%Seleção de mecanismos para o mecanismo de ajuste das costas da Cmaa LINET


clear all
clc
```

```
load('backrest_linet_dualbases.mat')

    for i=1:1:size(Z,1)
       J_a(i,6) = 5 - sum(J_a(i,:));
       J_b(i,6) = 5 - sum(J_b(i,:));
       J_c(i,6) = 5 - sum(J_c(i,:));
       J_d(i,6) = 5 - sum(J_d(i,:));
       J_e(i,6) = 5 - sum(J_e(i,:));
       J_f(i,6) = 5 - sum(J_f(i,:));
       J_g(i,6) = 5 - sum(J_g(i,:));
       J_h(i,6) = 5 - sum(J_h(i,:));
       J_i(i,6) = 5 - sum(J_i(i,:));
       J_j(i,6) = 5 - sum(J_j(i,:));
    end

%Criterio 1 - juntas (b) não deve ser alterada
for i=1:1:size(Z,1)
    if  J_b(i,6) == 5 % && sum(J_c(i,:)) == 0  && sum(J_a(i,:)) == 0
        Sl(i,1) = 1;
    else
        Sl(i,1) = 0;
    end
end

C1 = sum(Sl(:,1))

%Criterio 2 - juntas (d) (e) (f) (g) (h) (i) e (j)  devem continuar como pares inferiore

for i = 1:1:size(Z,1)
    if J_d(i,6) == 5
        X2(i,1) = 1;
    elseif J_d(i,6) == 4 && J_d(i,1) == 1 %Universal xz
```

```
        X2(i,1) = 1;
    elseif J_d(i,6) == 4 && J_d(i,2) == 1 %Universal yz
        X2(i,1) = 1;
    elseif J_d(i,6) == 3 && J_d(i,2) == 1 && J_d(i,1) == 1 %Esfrico
        X2(i,1) = 1;
    elseif J_d(i,6) == 3 && J_d(i,3) == 1 && J_d(i,4) == 1 %Planar
        X2(i,1) = 1;
    else
        X2(i,1) = 0;
    end
end

for i = 1:1:size(Z,1)
    if J_e(i,6) == 5
        X2(i,2) = 1;
    elseif J_e(i,6) == 4 && J_e(i,1) == 1 %Universal xz
        X2(i,2) = 1;
    elseif J_e(i,6) == 4 && J_e(i,2) == 1 %Universal yz
        X2(i,2) = 1;
    elseif J_e(i,6) == 3 && J_e(i,2) == 1 && J_e(i,1) == 1 %Esfrico
        X2(i,2) = 1;
    elseif J_e(i,6) == 3 && J_e(i,3) == 1 && J_e(i,4) == 1 %Planar
        X2(i,2) = 1;
    else
        X2(i,2) = 0;
    end
end


for i = 1:1:size(Z,1)
    if J_f(i,6) == 5
        X2(i,3) = 1;
    elseif J_f(i,6) == 4 && J_f(i,1) == 1 %Universal xz
```

```
        X2(i,3) = 1;
    elseif J_f(i,6) == 4 && J_f(i,2) == 1 %Universal yz
        X2(i,3) = 1;
    elseif J_f(i,6) == 3 && J_f(i,2) == 1 && J_f(i,1) %Esfrico
        X2(i,3) = 1;
    elseif J_f(i,6) == 3 && J_f(i,3) == 1 && J_f(i,4) == 1 %Planar
        X2(i,3) = 1;
    else
        X2(i,3) = 0;
    end
end


for i = 1:1:size(Z,1)
    if J_g(i,6) == 5
        X2(i,4) = 1;
    elseif J_g(i,6) == 4 && J_g(i,1) == 1 %Universal xz
        X2(i,4) = 1;
    elseif J_g(i,6) == 4 && J_g(i,2) == 1 %Universal yz
        X2(i,4) = 1;
    elseif J_g(i,6) == 3 && J_g(i,2) == 1 && J_g(i,1) == 1 %Esfrico
        X2(i,4) = 1;
    elseif J_g(i,6) == 3 && J_g(i,3) == 1 && J_g(i,4) == 1 %Planar
        X2(i,4) = 1;
    else
        X2(i,4) = 0;
    end
end

for i = 1:1:size(Z,1)
    if J_h(i,6) == 5
        X2(i,5) = 1;
    elseif J_h(i,6) == 4 && J_h(i,1) == 1 %Universal xz
```

```
        X2(i,5) = 1;
    elseif J_h(i,6) == 4 && J_h(i,2) == 1 %Universal yz
        X2(i,5) = 1;
    elseif J_h(i,6) == 3 && J_h(i,2) == 1 && J_h(i,1) == 1 %Esfrico
        X2(i,5) = 1;
    elseif J_h(i,6) == 3 && J_h(i,3) == 1 && J_h(i,4) == 1 %Planar
        X2(i,5) = 1;
    else
        X2(i,5) = 0;
    end
end

for i = 1:1:size(Z,1)
    if J_i(i,6) == 5
        X2(i,6) = 1;
    elseif J_i(i,6) == 4 && J_i(i,1) == 1 %Universal xz
        X2(i,6) = 1;
    elseif J_i(i,6) == 4 && J_i(i,2) == 1 %Universal yz
        X2(i,6) = 1;
    elseif J_i(i,6) == 3 && J_i(i,2) == 1 && J_i(i,1) == 1 %Esfrico
        X2(i,6) = 1;
    elseif J_i(i,6) == 3 && J_i(i,3) == 1 && J_i(i,4) == 1 %Planar
        X2(i,6) = 1;
    else
        X2(i,6) = 0;
    end
end

for i = 1:1:size(Z,1)
    if J_j(i,6) == 5
        X2(i,7) = 1;
    elseif J_j(i,6) == 3 && J_j(i,3) == 1 && J_j(i,4) == 1 %Planar xy
        X2(i,7) = 1;
    elseif J_j(i,6) == 3 && J_j(i,3) == 1 && J_j(i,5) == 1 %Planar xy
        X2(i,7) = 1;
    else
        X2(i,7) = 0;
    end
end

for i = 1:1:size(Z,1)
    if J_a(i,6) == 5
        X2(i,8) = 1;
    elseif J_a(i,6) == 4 && J_a(i,1) == 1 %Universal xz
        X2(i,8) = 1;
    elseif J_a(i,6) == 4 && J_a(i,2) == 1 %Universal yz
        X2(i,8) = 1;
    elseif J_a(i,6) == 3 && J_a(i,2) == 1 && J_a(i,1) == 1 %Esfrico
        X2(i,8) = 1;
    elseif J_a(i,6) == 3 && J_a(i,3) == 1 && J_a(i,4) == 1 %Planar
        X2(i,8) = 1;
    else
        X2(i,8) = 0;
    end
end

for i = 1:1:size(Z,1)
    if J_c(i,6) == 5
        X2(i,9) = 1;
    elseif J_c(i,6) == 4 && J_c(i,1) == 1 %Universal xz
        X2(i,9) = 1;
    elseif J_c(i,6) == 4 && J_c(i,2) == 1 %Universal yz
        X2(i,9) = 1;
    elseif J_c(i,6) == 3 && J_c(i,2) == 1 && J_c(i,1) == 1 %Esfrico
        X2(i,9) = 1;
    elseif J_c(i,6) == 3 && J_c(i,3) == 1 && J_c(i,4) == 1 %Planar
```

```
        X2(i,9) = 1;
    else
        X2(i,9) = 0;
    end
end

 XL(:,1) = sum(X2,2);
for i = 1:1:size(Z,1)
    if XL(i,1) == 9
        Sl(i,2) = 1;
    else
        Sl(i,2) = 0;
    end
end

C2 = sum(Sl(:,2))



%Criterio 3 - As juntas (a),(d),(e),(g) e(i) devem ser modificadas(BASE)

for i=1:1:size(Z,1)
    if  J_h(i,6) == 5  && J_f(i,6) == 5  && J_j(i,6) == 5 && J_c(i,6) == 5
        Sl(i,3) = 1;
    else
        Sl(i,3) = 0;
    end
end

C3 = sum(Sl(:,3))

        %Intersecao entre os conjuntos
    SL(:,1) = sum(Sl,2);
        for i=1:1:size(Z,1)
        if SL(i,1) == 3
            SL(i,2) = 1;
            fprintf('%d\n', i)
        else
            SL(i,2) = 0;
        end
    end


 X= sum(SL(:,2))
```

A.2 CASE STUDY II - ELEGANZA'S LEG REST ADJUSTMENT MECHANISM

Mechanism modeled by Davies' method in the pose presented in Table 6.

```
%Constraint analysis for leg rest mechanism:

clc
clear all

%Points:
k = [0 0 0];
l = [27.5 12, 0];
m = [25 5 0];
n = [15 -2.5 0];
o = [5 -10 0];
p = [52.5 -15 0];
q = [55.5 -5 0];
r = [62.5 15 0];

% Primal vectors:

R = [1 0 0];
```

```
S = [0 1 0];
T = [0 0 1];
U = [1 0 0];
V = [0 1 0];
W = [0 0 1];


% Wrenches:

    %Joint k (Rotative in z)
Rk = [R 0 0 0];
Sk = [S 0 0 0];
Uk = [cross(k,U) U];
Vk = [cross(k,V) V];
Wk = [cross(k,W) W];

    %Joint l (Rotative in z)
Rl = [R 0 0 0];
Sl = [S 0 0 0];
Ul = [cross(l,U) U];
Vl = [cross(l,V) V];
Wl = [cross(l,W) W];

    %Joint m (Rotative in z)
Rm = [R 0 0 0];
Sm = [S 0 0 0];
Um = [cross(m,U) U];
Vm = [cross(m,V) V];
Wm = [cross(m,W) W];

    %Joint n (Prismatic in y)
Rn = [R 0 0 0];
Sn = [S 0 0 0];
```

```
Tn = [T 0 0 0];
Un = [cross(n,U) U];
%Vn = [cross(n,V) V];
Wn = [cross(n,W) W];

    %Joint o (Rotative in z)
Ro = [R 0 0 0];
So = [S 0 0 0];
Uo = [cross(o,U) U];
Vo = [cross(o,V) V];
Wo = [cross(o,W) W];

    %Joint o (Rotative in z)
Rp = [R 0 0 0];
Sp = [S 0 0 0];
Up = [cross(p,U) U];
Vp = [cross(p,V) V];
Wp = [cross(p,W) W];

    %Joint q (Prismatic in x)
Rq = [R 0 0 0];
Sq = [S 0 0 0];
Tq = [T 0 0 0];
%Uq = [cross(p,U) U];
Vq = [cross(p,V) V];
Wq = [cross(p,W) W];

    %Joint r (Rotative in z)
Rr = [R 0 0 0];
Sr = [S 0 0 0];
Ur = [cross(r,U) U];
Vr = [cross(r,V) V];
Wr = [cross(r,W) W];
```

```
%Unit network matrix:
Ad = [Rk' Sk' Uk' Vk' Wk' Rl' Sl' Ul' Vl' Wl' Rm' Sm' Um' Vm' Wm' Rn' Sn' Tn' Un' W

%Cut-set matrix:
Q = [0 -1 1 0 0 0 0 0;
     0 -1 0 1 0 0 0 0;
     0  1 0 0 1 0 0 0;
     -1 1 0 0 0 1 0 0;
     -1 1 0 0 0 0 1 0;
     1 -1 0 0 0 0 0 1];

 Qa = [Q(:,1) Q(:,1) Q(:,1) Q(:,1) Q(:,1) Q(:,2) Q(:,2) Q(:,2) Q(:,2) Q(:,2) Q(:,3)

      lambda = size(Ad,1);
 q = size(Qa,1); %Number of cut-sets
 C = size(Ad,2); %Gross deegre of constraint

An = zeros(q*lambda, C);

for j = 1:1:q
    for i = 1:1:C
        for k = 1:1:lambda
   An((j-1)*lambda+k,i) = Qa(j,i)*Ad(k,i);
        end
    end
end

Cna = size(Ad,2) - rank(An)

rrefAn = rref(An);
```

Commands applied in the Sagemath to create the matroid $\mathcal{M}_{AN}$ relative to the leg rest adjustment mechanism from LINET:

```
An = Matrix(QQ, [[1, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
  -3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, -1],
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 1, 0, 0, -7, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, -1],
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 1, 0, 0, 0, -4, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 1, 0, 0, -4, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 1, 0, 0, 0, -3, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 1, 0, 0, 0, 0, 1, 0, 0, 0, -1.5, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 3],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 3],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]])

M = Matroid(An)

Md= M.dual()

sorted(sorted(X) for X in Md.bases())
```

Method of selection applied to the leg rest adjustment mechanism:

```
%Nova selecao de bases para mecanismos das pernas - Iftomm 2018

clear all
clc

load('legrest_linet_dualbases.mat')

Z = zeros(size(D,1), (max(max(D)+1)));
    for i=1:1:size(D,1)
        for j=1:1:size(D,2)
            Z(i,(D(i,j)+1)) = 1;
        end
    end
```

```
%Joint by joint:

   %Joint k (rotative in z):
   for i=1:1:size(D,1)
       for j=1:1:5
           J_k(i,j) = Z(i,j);
       end
   end

   %Joint l (rotative in z):
   for i=1:1:size(D,1)
       for j=6:1:10
           J_l(i,j-5) = Z(i,j);
       end
   end

   %Joint m (rotative in z):
   for i=1:1:size(D,1)
       for j=11:1:15
           J_m(i,j-10) = Z(i,j);
       end
   end

   %Joint n (prismatic in y):
   for i=1:1:size(D,1)
       for j=16:1:20
           J_n(i,j-15) = Z(i,j);
       end
   end

   %Joint o (rotative in z):
   for i=1:1:size(D,1)
       for j=21:1:25
```

```
        J_o(i,j-20) = Z(i,j);
    end
end

%Joint p (rotative in z):
for i=1:1:size(D,1)
    for j=26:1:30
        J_p(i,j-25) = Z(i,j);
    end
end

%Joint q (prismatic in x):
for i=1:1:size(D,1)
    for j=31:1:35
        J_q(i,j-30) = Z(i,j);
    end
end

%Joint r (rotative in z):
for i=1:1:size(D,1)
    for j=36:1:40
        J_r(i,j-35) = Z(i,j);
    end
end

%Types of joints:
for  i=1:1:size(Z,1)
    J_k(i,6) = 5 - sum(J_k(i,:));
    J_l(i,6) = 5 - sum(J_l(i,:));
    J_m(i,6) = 5 - sum(J_m(i,:));
    J_n(i,6) = 5 - sum(J_n(i,:));
    J_o(i,6) = 5 - sum(J_o(i,:));
    J_p(i,6) = 5 - sum(J_p(i,:));
    J_q(i,6) = 5 - sum(J_q(i,:));
    J_r(i,6) = 5 - sum(J_r(i,:));
  end

 %Criterio 1 - Atuador CPR comercial,
    %juntas (q) e (r) nao devem ser alteradas, e as juntas (p) deve
    %ser cilíndricas
  for i=1:1:size(Z,1)
    if  J_p(i,5) == 1 && J_p(i,6) == 4 && J_q(i,6) == 5 && J_r(i,6) == 5 %| J_r(i,:)
      X1(i,1) = 1;
    else
      X1(i,1) = 0;
    end
  end
    for i=1:1:size(Z,1)
    if  J_r(i,5) == 1 && J_r(i,6) == 4 && J_p(i,6) == 5 && J_q(i,6) == 5 %| J_r(i,:)
      X1(i,2) = 1;
    else
      X1(i,2) = 0;
    end
  end


%  for i=1:1:size(Z,1)
%    X1(i,3) = sum(X1(i,:));
%  end
%
  for i=1:1:size(Z,1)
    if X1(i,1) == 1 | X1(i,2) == 1
      Sl(i,1) = 1;
    else
      Sl(i,1) = 0;
    end
```

```
end

C1 = sum(Sl(:,1))


            %Criterio 2 - Junta prismatica possui feita de item comercial,
    %junta (n) nao deve ser alterada:
for i=1:1:size(Z,1)
    if J_n(i,6) == 5
        Sl(i,2) = 1;
    else
        Sl(i,2) = 0;
    end
end
    %Conjunto de bases que atendem ao criterio 2:

    C2 = sum(Sl(:,2))

  %Criterio 3 - As demais juntas devem ter pelo menos tres restricoes
    for i=1:1:size(Z,1)
        if J_k(i,6) >= 3 && J_l(i,6)>=3 && J_m(i,6) >= 3 && J_o(i,6) >= 3
            Sl(i,3) = 1;
        else
            Sl(i,3) = 0;
        end
    end

    C3 = sum(Sl(:,3))

%Interseccao entre os dois conjuntos

SL(1,:) = sum(Sl,2);
    for i=1:1:size(Z,1)
```

```
        if SL(1,i) == 3
            SL(2,i) = 1;
            fprintf('%d\n', i)
        else
            SL(2,i) = 0;
        end
    end


 X= sum(SL(2,:))
```

A.3 CASE STUDY III - LAR'S BACKREST ADJUSTMENT MECHANISM

Mechanism modeled by Davies' method in the pose presented in Table 8.

```
% Constraint analysis of the backrest mechanism of a linet's robotic hospital
% bed

clc
clear all



%Points:
a=[14 1 0];
b=[13 5 0];
c=[9 8 0];
d=[10 1 0];
e=[0 0 0];
f=[-2.5 8 0];
g=[-1 12.5 0];
h=[-1 9 0];
i=[3 6 0];
j=[5 3 0];
```

```
% Primal vectors:

R = [1 0 0];
S = [0 1 0];
T = [0 0 1];
U = [1 0 0];
V = [0 1 0];
W = [0 0 1];

%Wrenches:

%Joint a (rotative in z):
Ra = [R 0 0 0];
Sa = [S 0 0 0];
Ua = [cross(a,U) U];
Va = [cross(a,V) V];
Wa = [cross(a,W) W];
%Joint b(prismatic in x):
Rb = [R 0 0 0];
Sb = [S 0 0 0];
Ub = [cross(b,U) U];
Vb = [cross(b,V) V];
Wb = [cross(b,W) W];
%Joint c(rotative in z):
Rc = [R 0 0 0];
Sc = [S 0 0 0];
Uc = [cross(c,U) U];
Vc = [cross(c,V) V];
Wc = [cross(c,W) W];
%Joint d(rotative in z):
Rd = [R 0 0 0];
Sd = [S 0 0 0];
Ud = [cross(d,U) U];
```

```
Vd = [cross(d,V) V];
Wd = [cross(d,W) W];
%Joint e(rotative in z):
Re = [R 0 0 0];
Se = [S 0 0 0];
Ue = [cross(e,U) U];
Ve = [cross(e,V) V];
We = [cross(e,W) W];
%Joint f(rotative in z):
Rf = [R 0 0 0];
Sf = [S 0 0 0];
Uf = [cross(f,U) U];
Vf = [cross(f,V) V];
Wf = [cross(f,W) W];
%Joint g(prismatic in x):
Rg = [R 0 0 0];
Sg = [S 0 0 0];
Tg = [T 0 0 0];
Vg = [cross(g,V) V];
Wg = [cross(g,W) W];
%Joint h(rotative in z):
Rh = [R 0 0 0];
Sh = [S 0 0 0];
Uh = [cross(h,U) U];
Vh = [cross(h,V) V];
Wh = [cross(h,W) W];
%Joint i(prismatic in x):
Ri = [R 0 0 0];
Si = [S 0 0 0];
Ti = [T 0 0 0];
Vi = [cross(i,V) V];
Wi = [cross(i,W) W];
%Joint j(revolute in z):
```

```
Rj = [R 0 0 0];
Sj = [S 0 0 0];
%Tj = [T 0 0 0];
Uj = [cross(j,U) U];
Vj = [cross(j,V) V];
Wj = [cross(j,W) W];

Ad = [Ra' Sa' Ua' Va' Wa' Rb' Sb' Ub' Vb' Wb' Rc' Sc'
Uc' Vc' Wc' Rd' Sd' Ud' Vd' Wd' Re' Se' Ue' Ve' We' Rf'
Sf' Uf' Vf' Wf' Rg' Sg' Tg' Vg' Wg' Rh' Sh' Uh' Vh' Wh'
Ri' Si' Ti' Vi' Wi' Rj' Sj' Uj' Vj' Wj'];
% a  b  c  d  e  f  g  h  i  j
Q = [-1  1  0  0  0  0  0  0  0  0;
1  0  1  1  0  0  0  0  0  0;
-1  0  0  -1 0  0  1  0  0  0;
1  0  0  1  1  0  0  0  0  1;
-1  0  0  -1 0  1  0  0  0 -1;
0  0  0  0  0  0  0  1  0  1;
0  0  0  0  0  0  0  0  1  1];

Qa = [Q(:,1) Q(:,1) Q(:,1) Q(:,1) Q(:,1) Q(:,2) Q(:,2)
Q(:,2) Q(:,2) Q(:,2) Q(:,3) Q(:,3) Q(:,3) Q(:,3) Q(:,3)
Q(:,4) Q(:,4) Q(:,4) Q(:,4) Q(:,4) Q(:,5) Q(:,5) Q(:,5)
Q(:,5) Q(:,5) Q(:,6) Q(:,6) Q(:,6) Q(:,6) Q(:,6) Q(:,7)
Q(:,7) Q(:,7) Q(:,7) Q(:,7) Q(:,8) Q(:,8) Q(:,8) Q(:,8)
Q(:,8) Q(:,9) Q(:,9) Q(:,9) Q(:,9) Q(:,9) Q(:,10) Q(:,10)
Q(:,10) Q(:,10) Q(:,10)];
lambda = size(Ad,1);
q = size(Qa,1); %Number of cut-sets
C = size(Ad,2); %Gross deegre of constraint

An = zeros(q*lambda, C);
```

```
for j = 1:1:q

for i = 1:1:C

for k = 1:1:lambda

An((j-1)*lambda+k,i) = Qa(j,i)*Ad(k,i);

end

end

end


Cna = size(Ad,2) - rank(An)


rrefAn = rref(An);
```

Commands applied in the Sagemath to create the matroid $\mathcal{M}_{AN}$ relative to the backrest adjustment mechanism from LAR:

```
A = Matrix(QQ, [[-1, 0, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0,  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, -1, 0, 0, 7, 0, 1, 0, 0, -7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, -7, 0, 0, 0, -2, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, -1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, -1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, -1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 4, 1, 0, 0, 0, 1,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, -7, 0, 0, 0, 0, 0, 0, 1, 0, 0, -7, 0, 1, 0, 0,
 -5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, -4, 7, 0, 0, 0, -1, 5, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0,
 -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, -1, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 5,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, -1, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, -7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, -5, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 1, 0, 0, 0, 4],
[0, 1, 0, 0, -7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, -5,
 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 1, 0, 0, -3],
[0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 5, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, -4, 3, 0],
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0,
 -1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, -1, 0, 0, 0, -4],
[0, -1, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 5,
 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, -1, 0, 0, 3],
[0, 0, 0, -7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, -5, 0,
 0, 0, 0, 0, 0, 0, 0, -3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 4, -3, 0],
[0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, -1, 0, 0],
[0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, -1, 0],
[0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, -1],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 2.5,
  0, 0, 0, 0, 0, 1, 0, 0, 0, 4],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, -2,
  0, 0, 0, 0, 0, 0, 1, 0, 0, -3],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -2.5, 2,
  0, 0, 0, 0, 0, 0, 0, 0, -4, 3, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  1, 0, 0, 0, 3.5, 1, 0, 0, 0, 4],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 1, 0, 0, -6, 0, 1, 0, 0, -3],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 1, 6, 0, 0, 0, -4, 3, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 0, 0, 1, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 1, 0, 0, 0, 0, 1]])

M = Matroid(A)
Ma=M.dual()
sorted(sorted(T) for T in Ma.bases())
```

Method of selection applied to the backrest adjustment mechanism:

```
% Seleção de mecanismos para mecanismos
 das costas da cama da UFSC

clear all
clc

load(’backrest_ufsc_duais.mat’)

Z = zeros(size(Duais,1), (max(max(Duais)+1)));
for i=1:1:size(Duais,1)
for j=1:1:size(Duais,2)
Z(i,(Duais(i,j)+1)) = 1;
end
end

disp(’Matriz binaria’);

%Critério 1: Juntas h, i e j devem ter a
 configuracao CPR ou RPC
```

```
%Joint h (rotative in z):
for i=1:1:size(Z,1)
for j=36:1:40
J_h(i,j-35) = Z(i,j);
end
end

disp('Junta h');

%Joint i (prismatic):
for i=1:1:size(Z,1)
for j=41:1:45
J_i(i,j-40) = Z(i,j);
end
end
disp('Junta i');
%Joint j (rotative in z):
for i=1:1:size(Z,1)
for j=46:1:50
J_j(i,j-45) = Z(i,j);
end
end
disp('Junta j');
%Types of joints:
for  i=1:1:size(Z,1)
J_h(i,6) = 5 - sum(J_h(i,:));
J_i(i,6) = 5 - sum(J_i(i,:));
J_j(i,6) = 5 - sum(J_j(i,:));
end
disp('types of joint');
%Criterio 1 - Atuador CPR ou RPC comercial,
%juntas (h) nao devem ser alteradas, e as juntas
```

```
 (i) e (j) devem ser cilíndricas, uma por vez

for i=1:1:size(Z,1)
if  J_h(i,5) == 1 && J_h(i,6) == 4 && J_i(i,6) == 5
&& J_j(i,6) == 5 %| J_r(i,:) == [0 0 0 0 1 4]
X1(i,1) = 1;
else
X1(i,1) = 0;
end
end
disp('CPR')

for i=1:1:size(Z,1)
if  J_j(i,5) == 1 && J_j(i,6) == 4 && J_i(i,6) == 5
&& J_h(i,6) == 5 %| J_r(i,:) == [0 0 0 0 1 4]
X1(i,2) = 1;
else
X1(i,2) = 0;
end
end
disp('RPC')

for i=1:1:size(Z,1)
if X1(i,1) == 1 | X1(i,2) == 1
Sl(i,1) = 1;
else
Sl(i,1) = 0;
end
end

C1 = sum(Sl(:,1))

%Criterio 2 - Junta g, originalmente prismática
```

```
 pela adicao de folga eh considerada como planar

for i=1:1:size(Z,1)
for j=31:1:35
J_g(i,j-30) = Z(i,j);
end
end

disp(’Junta g’)
for i=1:1:size(Z,1)
if J_g(i,:) == [0 1 0 0 1]
Sl(i,2) = 1;
else
Sl(i,2) = 0;
end
end

C2 = sum(Sl(:,2))



%Joint a (rotative in z):
for i=1:1:size(Z,1)
for j=1:1:5
J_a(i,j) = Z(i,j);
end
end
disp(’Junta a’)

for i=1:1:size(Z,1)
if J_a(i,:) == [0 0 0 0 0]
X3(i,1) = 1;
elseif J_a(i,:) == [1 1 0 0 0]
X3(i,1) = 1;
```

```
elseif J_a(i,:) == [0 0 0 0 1]
X3(i,1) = 1;
else
X3(i,1) = 0;
end
end

disp('criterio a')

%Joint b (rotative in z):
for i=1:1:size(Z,1)
for j=6:1:10
J_b(i,j-5) = Z(i,j);
end
end

disp('Junta b')

for i=1:1:size(Z,1)
if J_b(i,:) == [0 0 0 0 0]
X3(i,2) = 1;
elseif J_b(i,:) == [1 1 0 0 0]
X3(i,2) = 1;
elseif J_b(i,:) == [0 0 0 0 1]
X3(i,2) = 1;
else
X3(i,2) = 0;
end
end

disp('criterio b')

%Joint c (rotative in z):
```

```
for i=1:1:size(Z,1)

for j=11:1:15

J_c(i,j-10) = Z(i,j);

end

end


disp('Joint c')


for i=1:1:size(Z,1)

if J_c(i,:) == [0 0 0 0 0]

X3(i,3) = 1;

elseif J_c(i,:) == [1 1 0 0 0]

X3(i,3) = 1;

elseif J_c(i,:) == [0 0 0 0 1]

X3(i,3) = 1;

else

X3(i,3) = 0;

end

end


disp('criterio c')


%Joint d (prismatic in y):

for i=1:1:size(Z,1)

for j=16:1:20

J_d(i,j-15) = Z(i,j);

end

end


disp('Junta d')


for i=1:1:size(Z,1)

if J_d(i,:) == [0 0 0 0 0]
```

```
X3(i,4) = 1;
elseif J_d(i,:) == [1 1 0 0 0]
X3(i,4) = 1;
elseif J_d(i,:) == [0 0 0 0 1]
X3(i,4) = 1;
else
X3(i,4) = 0;
end
end

disp('criterio d')

%Joint e (rotative in z):
for i=1:1:size(Z,1)
for j=21:1:25
J_e(i,j-20) = Z(i,j);
end
end

disp('Junta e')
for i=1:1:size(Z,1)
if J_e(i,:) == [0 0 0 0 0]
X3(i,5) = 1;
elseif J_e(i,:) == [1 1 0 0 0]
X3(i,5) = 1;
elseif J_e(i,:) == [0 0 0 0 1]
X3(i,5) = 1;
else
X3(i,5) = 0;
end
end

disp('criterio e')
```

```
%Joint f (rotative in z):
for i=1:1:size(Z,1)
for j=26:1:30
J_f(i,j-25) = Z(i,j);
end
end

disp ('Junta f')

for i=1:1:size(Z,1)
if J_f(i,:) == [0 0 0 0 0]% | J_a(i,:) == [1 1 0 0 0]
| J_a(i,:) == [0 0 0 0 1]
X3(i,6) = 1;
elseif J_f(i,:) == [1 1 0 0 0]
X3(i,6) = 1;
elseif J_f(i,:) == [0 0 0 0 1]
X3(i,6) = 1;
else
X3(i,6) = 0;
end
end

disp('criterio f')

XL(1,:) = sum(X3,2);
for i=1:1:size(Z,1)
if XL(1,i) == 6
Sl(i,3) = 1;
else
Sl(i,3) = 0;
end
end
```

```
%Intersecao entre os conjuntos
SL(1,:) = sum(Sl,2);
for i=1:1:size(Z,1)
if SL(1,i) == 3
SL(2,i) = 1;
fprintf('%d\n', i)
else
SL(2,i) = 0;
end
end


X= sum(SL(2,:))
```

A.4 CASE STUDY IV - LAR'S LEG REST ADJUSTMENT MECHANISM

Mechanism modeled by Davies' method in the pose presented in Table 11.

```
% Constraint analysis of the backrest mechanism of a linet's robotic hospital
% bed

clc
clear all



%Points:
a=[0 3 0];
b=[10 7 0];
c=[5 3 0];
d=[2 0 0];
e=[0 0 0];
f=[21 5 0];
g=[20 2 0];
```

```
h=[18 -2 0];

% Primal vectors:

R = [1 0 0];
S = [0 1 0];
T = [0 0 1];
U = [1 0 0];
V = [0 1 0];
W = [0 0 1];

%Wrenches:

%Joint a (rotative in z):
Ra = [R 0 0 0];
Sa = [S 0 0 0];
Ua = [cross(a,U) U];
Va = [cross(a,V) V];
Wa = [cross(a,W) W];



%Joint a (rotative in z):
Rb = [R 0 0 0];
Sb = [S 0 0 0];
Ub = [cross(b,U) U];
Vb = [cross(b,V) V];
Wb = [cross(b,W) W];

%Joint c (prismatic in x):
Rc = [R 0 0 0];
Sc = [S 0 0 0];
Tc = [T 0 0 0];
Vc = [cross(b,V) V];
```

```
Wc = [cross(b,W) W];


%Joint d (rotative in z):
Rd = [R 0 0 0];
Sd = [S 0 0 0];
Ud = [cross(d,U) U];
Vd = [cross(d,V) V];
Wd = [cross(d,W) W];

%Joint e (rotative in z):
Re = [R 0 0 0];
Se = [S 0 0 0];
Ue = [cross(e,U) U];
Ve = [cross(e,V) V];
We = [cross(e,W) W];

%Joint f (rotative in z):
Rf = [R 0 0 0];
Sf = [S 0 0 0];
Uf = [cross(f,U) U];
Vf = [cross(f,V) V];
Wf = [cross(f,W) W];

%Joint g (rotative in z):
Rg = [R 0 0 0];
Sg = [S 0 0 0];
Tg = [T 0 0 0];
Vg = [cross(g,V) V];
Wg = [cross(g,W) W];


%Joint h (rotative in z):
```

```
Rh = [R 0 0 0];
Sh = [S 0 0 0];
Uh = [cross(h,U) U];
Vh = [cross(h,V) V];
Wh = [cross(h,W) W];

%UNIT ACTION MATRIX:
Ad = [Ra' Sa' Ua' Va' Wa' Rb' Sb' Ub' Vb' Wb' Rc' Sc' Tc'
Vc' Wc' Rd' Sd' Ud' Vd' Wd' Re' Se' Ue' Ve' We' Rf' Sf'
Uf' Vf' Wf' Rg' Sg' Tg' Vg' Wg' Rh' Sh' Uh' Vh' Wh'];

%CUTSET MATRIX:

%    a  b  c  d  e  f  g  h
Q = [1  1  1  0  0  0  0  0;
1  1  0  1  0  0  0  0;
-1  0  0  0  1  0  0  0;
0 -1  0  0  0  1  0  0;
0 -1  0  0  0  0  1  0;
0  1  0  0  0  0  0  1];

Qa = [Q(:,1) Q(:,1) Q(:,1) Q(:,1) Q(:,1) Q(:,2) Q(:,2)
Q(:,2) Q(:,2) Q(:,2) Q(:,3) Q(:,3) Q(:,3) Q(:,3) Q(:,3)
Q(:,4) Q(:,4) Q(:,4) Q(:,4) Q(:,4) Q(:,5) Q(:,5) Q(:,5)
Q(:,5) Q(:,5) Q(:,6) Q(:,6) Q(:,6) Q(:,6) Q(:,6) Q(:,7)
Q(:,7) Q(:,7) Q(:,7) Q(:,7) Q(:,8) Q(:,8) Q(:,8) Q(:,8)
Q(:,8)];
lambda = size(Ad,1);
q = size(Qa,1); %Number of cut-sets
C = size(Ad,2); %Gross deegre of constraint

An = zeros(q*lambda, C);
```

```
for j = 1:1:q

for i = 1:1:C

for k = 1:1:lambda

An((j-1)*lambda+k,i) = Qa(j,i)*Ad(k,i);

end

end

end


Cna = size(Ad,2) - rank(An)


rrefAn = rref(An);
```

Commands applied in the Sagemath to create the matroid $\mathcal{M}_{AN}$ relative to the leg rest adjustment mechanism from LAR:

```
A = Matrix(QQ, [[1, 0, 0, 0, 3, 1, 0, 0, 0, 7, 1, 0, 0, 0, 7,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 1, 0, 0, -10, 0, 1, 0, 0, -10, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0],
[0, 0, -3, 0, 0, 0, 0, -7, 10, 0, 0, 0, 1, 10, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0],
[0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[1, 0, 0, 0, 3, 1, 0, 0, 0, 7, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 1, 0, 0, -10, 0, 0, 0, 0, 0, 0, 1, 0, 0,
 -2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, -3, 0, 0, 0, 0, -7, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[-1, 0, 0, 0, -3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, -1, 0, 0, 0, -7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 1, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, -1, 0, 0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 1, 0, 0, -21, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 7, -10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, -5, 21, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, -1, 0, 0, 0, -7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, -1, 0, 0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, -20, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 7, -10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 20, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 1, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, -2],
[0, 0, 0, 0, 0, 0, 1, 0, 0, -10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, -18],
[0, 0, 0, 0, 0, 0, 0, -7, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 18, 0],
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]])

M = Matroid(A)
Ma=M.dual()
sorted(sorted(T) for T in Ma.bases())
```

Method of selection applied to the leg rest adjustment mechanism:

```
% Selection of self-aligning mechanisms which are
 kinematically identical to the leg rest adjustment
  mechanism designed by LAR Laboratory
```

```
clc
clear all

load('dual_bases.mat')

Z = zeros(size(D,1), (max(max(D)+1)));
for i=1:1:size(D,1)
for j=1:1:size(D,2)
Z(i,(D(i,j)+1)) = 1;
end
end

disp('Matriz binaria');

% Criterio 1 - A junta (c) deve se manter prismática:

%Joint c (prismatic):
for i=1:1:size(Z,1)
for j=11:1:15
J_c(i,j-10) = Z(i,j);
end
end

for i=1:1:size(Z,1)
J_c(i,6) = 5 - sum(J_c(i,:));
end

for i = 1:1:size(Z,1)
if J_c(i,6) == 5
S1(i,1) = 1;
else
S1(i,1) = 0;
```

```
end
end

disp('Junta c');


C1 = sum(Sl(:,1))


%Critrio 2 - Juntas (f),(g) e (h) possui estrutura
 RPR, devem ser CPR ou RPC

%Joint f (rotative in z):
for i=1:1:size(Z,1)
for j=26:1:30
J_f(i,j-25) = Z(i,j);
end
end

disp('Junta f');

%Joint g (prismatic):
for i=1:1:size(Z,1)
for j=31:1:35
J_g(i,j-30) = Z(i,j);
end
end

disp('Junta g');

%Joint h (rotative in z):
for i=1:1:size(Z,1)
for j=36:1:40
```

```
J_h(i,j-35) = Z(i,j);
end
end

disp('Junta h');

%Types of joints:
for  i=1:1:size(Z,1)
J_f(i,6) = 5 - sum(J_f(i,:));
J_g(i,6) = 5 - sum(J_g(i,:));
J_h(i,6) = 5 - sum(J_h(i,:));
end
disp('types of joint');

for i=1:1:size(Z,1)
if  J_f(i,5) == 1 && J_f(i,6) == 4 && J_g(i,6) == 5
&& J_h(i,6) == 5 %| J_r(i,:) == [0 0 0 0 1 4]
X2(i,1) = 1;
else
X2(i,1) = 0;
end
end
disp('CPR')
for i=1:1:size(Z,1)
if  J_h(i,5) == 1 && J_h(i,6) == 4 && J_g(i,6) == 5
&& J_f(i,6) == 5 %| J_r(i,:) == [0 0 0 0 1 4]
X2(i,2) = 1;
else
X2(i,2) = 0;
end
end
disp('RPC')
for i=1:1:size(Z,1)
```

```
if X2(i,1) == 1 | X2(i,2) == 1
Sl(i,2) = 1;
else
Sl(i,2) = 0;
end
end

C2 = sum(Sl(:,2))

%Criterio 3
% As juntas (a) (b) (d) e (e) originalmente rotativas,
podem ser transformadas em juntas esfericas ou cilindricas
pela adicao de folga:

%Joint a (rotative in z):
for i=1:1:size(Z,1)
for j=1:1:5
J_a(i,j) = Z(i,j);
end
end

%Joint b (rotative in z):
for i=1:1:size(Z,1)
for j = 6:1:10
J_b(i,j-5) = Z(i,j);
end
end

%Joint d (rotative in z):
for i=1:1:size(Z,1)
for j=16:1:20
J_d(i,j-15) = Z(i,j);
end
```

```
end

%Joint e (rotative in z):
for i=1:1:size(Z,1)
for j=21:1:25
J_e(i,j-20) = Z(i,j);
end
end

%Types of joints:
for  i=1:1:size(Z,1)
J_a(i,6) = 5 - sum(J_a(i,:));
J_b(i,6) = 5 - sum(J_b(i,:));
J_d(i,6) = 5 - sum(J_d(i,:));
J_e(i,6) = 5 - sum(J_e(i,:));
end
disp('types of joint');

for i=1:1:size(Z,1)
if J_a(i,6) == 5
X3(i,1) = 1;
elseif J_a(i,6) == 3 && J_a(i,1) == 1 && J_a(i,2) == 1
 %J_a(i,:) == [1 1 0 0 0]
X3(i,1) = 1;
elseif J_a(i,6) == 5 && J_a(i,5) == 1
%J_a(i,:) == [0 0 0 0 1]
X3(i,1) = 1;
else
X3(i,1) = 0;
end
end

for i=1:1:size(Z,1)
```

```
if J_b(i,6) == 5
X3(i,2) = 1;
elseif J_b(i,6) == 3 && J_b(i,1) == 1 && J_b(i,2) == 1
%J_b(i,:) == [1 1 0 0 0]
X3(i,2) = 1;
elseif J_b(i,6) == 5 && J_b(i,5) == 1
%J_b(i,:) == [0 0 0 0 1]
X3(i,2) = 1;
else
X3(i,2) = 0;
end
end


for i=1:1:size(Z,1)
if J_d(i,6) == 5
X3(i,3) = 1;
elseif J_d(i,6) == 4 && J_d(i,5) == 1
X3(i,3) = 1;
elseif J_d(i,6) == 3 && J_d(i,1) == 1 && J_d(i,2)== 1
X3(i,3) = 1;
else
X3(i,3) = 0;
end
end

for i=1:1:size(Z,1)
if J_e(i,6) == 5
X3(i,4) = 1;
elseif J_e(i,6) == 3 && J_e(i,1) == 1 && J_e(i,2) == 1
%J_e(i,:) == [1 1 0 0 0]
X3(i,4) = 1;
elseif J_e(i,6) == 5 && J_e(i,5) == 1
```

```
%J_e(i,:) == [0 0 0 0 1]
X3(i,4) = 1;
else
X3(i,4) = 0;
end
end

XL(1,:) = sum(X3,2);
for i=1:1:size(Z,1)
if XL(1,i) == 4
Sl(i,3) = 1;
else
Sl(i,3) = 0;
end
end
C3= sum(Sl(:,3))

%    Interseccao entre os conjuntos
SL(1,:) = sum(Sl,2);
for i=1:1:size(Z,1)
if SL(1,i) == 3
SL(2,i) = 1;
fprintf('%d\n', i)
else
SL(2,i) = 0;
end
end

X= sum(SL(2,:))
```