UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA MECÂNICA

COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

Autora: Mariana Moretti

Estudo Dinâmico e Simulação de uma Plataforma de Stewart com Ênfase na Implementação do Sistema de Controle

Orientador: Prof. Dr. João Maurício Rosário

Campinas, 2010.

63/2010

Mariana Moretti

Estudo Dinâmico e Simulação de uma Plataforma de Stewart com Ênfase na Implementação do Sistema de Controle

Dissertação apresentada ao Curso de Mestrado da Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas, como requisito para a obtenção do título de Mestre em Engenharia Mecânica.

Área de Concentração: Mecânica dos Sólidos e Projeto Mecânico

Orientador: Prof. Dr. João Maurício Rosário

Campinas, 1 de junho de 2010

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

	Moretti, Mariana
M817e	Estudo dinâmico e simulação de uma plataforma de Stewart com ênfase na implementação do sistema de controle / Mariana Moretti. --Campinas, SP: [s.n.], 2010. Orientador: João Maurício Rosário. Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica. 1. Robos - Sistemas de controle. 2. Mecatronica. 3. Robótica. I. Rosário, João Maurício. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

Título em Inglês: Dynamic study and simulation of the Stewart platform with emphasis on control system
Palavras-chave em Inglês: Robots - Control systems, Mechatronics, Robotics
Área de concentração: Mecânica dos Sólidos e Projeto Mecânico
Titulação: Mestre em Engenharia Mecânica
Banca examinadora: Humberto Ferasoli Filho, Ely Carneiro de Paiva
Data da defesa: 01/06/2010
Programa de Pós Graduação: Engenharia Mecânica

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO

DISSERTAÇÃO DE MESTRADO

Estudo Dinâmico e Simulação de uma Plataforma de Stewart com Ênfase na Implementação do Sistema de Controle

Autor: Mariana Moretti
Orientador: Prof. Dr. João Maurício Rosário

A Banca Examinadora composta pelos membros abaixo aprovou esta Tese:

Prof. Dr. João Maurício Rosário, Presidente
UNICAMP

Prof. Dr. Humberto Ferasoli Filho
UNESP - Bauru

Prof. Dr. Ely Carneiro de Paiva
UNICAMP

Campinas, 1 de junho de 2010

Dedicatória:

Dedico este trabalho aos meus pais Marinho e Li, ao meu irmão Eduardo e ao Rafael, meu marido.

Agradecimentos

Ao meu orientador, Prof. Dr. João Maurício Rosário, pela amizade, confiança, dedicação, e por acreditar no desenvolvimento deste trabalho.

Ao Prof. Dr. Didier Dumur pela orientação e direcionamento durante os primeiros meses de desenvolvimento deste trabalho. Agradeço novamente a ele e ao Prof. Dr. Patrick Boucher pela receptividade a mim proporcionada no *Département Automatique* da Supélec – Gif-sur-Yvette – França.

Ao Programa e Colegas de Pós Graduação em Engenharia Mecânica da Universidade Estadual de Campinas pelo apoio, orientações e incentivo.

A Capes pela bolsa de estudos que possibilitou a minha dedicação ao desenvolvimento desta pesquisa.

A todos os profissionais que participaram desta pesquisa e que de uma maneira ou outra contribuíram para a realização deste trabalho.

A todos os meus amigos que me ajudaram a crescer e sempre estiveram ao meu lado. Obrigada Alysson, Carol, Du, Marina e Marininha por estarem ao meu lado no momento mais importante deste trabalho, a defesa.

Aos meus pais e irmão que muito me apoiaram.

Ao meu marido que tanto me incentiva a construir uma carreira acadêmica e de pesquisa.

Resumo

Uma nova proposta de modelo dinâmico da Plataforma de Stewart é apresentada neste trabalho. Enquanto a cinemática inversa é usada para posicionar cada um dos seis braços do robô, o vetor de força que atua em seus deslocamentos é dado pelo torque de motores elétricos. A inércia em cada um dos pontos de apoio é aproximada por uma rigidez mecânica associada ao modulo de Young. Ainda, foi implementado um modelo dinâmico que usa a aproximação de Newton-Euler, amplamente aplicada na dinâmica inversa de robôs seriais, para o caso deste robô paralelo. Em ambas as abordagens, as equações foram implementadas em Matlab/Simulink^{TM}, e os resultados das simulações foram apresentados para validação das aproximações.

Palavras Chave: 1. Plataforma de Stewart. 2. Hexapodos. 3. Robôs Paralelos – Seis Graus de Liberdade.

Abstract

A new proposal for a dynamic model of the Stewart platform is presented. While the inverse kinematics is used to position each of the six arms of the robot, the vector of force acting on the displacement is given by the torque of electric motors. The inertia in each of the support points is approximated by a mechanical stiffness associated with the Young modulus. Still, it was implemented a dynamic model that uses the Newton-Euler approach, widely applied in the inverse dynamics of serial robots, for the case of this parallel robot. In both approaches, the equations were implemented in Matlab/SimulinkTM, and the simulation results were presented for validation of the approaches.

Keywords: 1 Stewart Platforms. 2 Hexapods. 3 Parallel Robots – Six DOF.

Sumário

Lista de figuras

Capítulo 1

Introdução

Os manipuladores robóticos são divididos em dois tipos, os seriais e os paralelos. Robôs seriais possuem as barras montadas seqüencialmente, o que significa que cada barra pode ter no máximo duas juntas e que a mesma pode ser conectada a no máximo duas outras barras, formando ou não uma malha fechada. Os braços acionadores podem ser barras em formato cilíndrico, e as juntas em formato de acionadores angulares. Já os manipuladores paralelos não possuem as barras ligadas diretamente umas às outras, sendo estas geralmente conectadas a uma estrutura rígida (Mazoni, 2003). A forma paralela pode garantir uma maior precisão, um maior torque e um carregamento mais uniformemente distribuído que os manipuladores seriais.

Os desenvolvimentos teóricos, metodológicos e práticos realizados recentemente mostraram o interesse crescente pelas aplicações diretas dos mecanismos paralelos aos problemas industriais reais. Nas situações pelas quais as necessidades em termos de precisão e agilidade são primordiais, os mecanismos paralelos se apresentam como alternativas confiáveis às configurações do tipo série, graças às características que lhes conferem uma grande riqueza de movimentos.

Um manipulador paralelo é composto classicamente de duas plataformas, uma base fixa e uma base móvel, interligadas por vários braços que se deslocam individualmente entre si. Esses braços são conectados às plataformas por intermédio de juntas passivas esféricas e/ou universais. Conseqüentemente, essas ligações não sofrem esforços além da tração ou da compressão, o que aumenta sua exatidão para o posicionamento e permite uma construção mais leve. Além disso, os manipuladores paralelos possuem uma elevada rigidez estrutural, já que a força global se encontra projetada em vários pontos ao mesmo tempo.

1.1 A Plataforma de Stewart-Gough

O objeto de estudo deste trabalho de pesquisa, a Plataforma de Stewart-Gough, é um manipulador paralelo de 6 graus de liberdade. Inicialmente desenvolvida, em 1965, para exercer a função de um simulador de vôo (Dumur, Rosário, & Machado, 2007), esta plataforma possui atualmente diversas aplicações, como por exemplo, no domínio da máquina-ferramenta, da pesquisa submarina, dos telescópios e da cirurgia ortopédica.

O mecanismo em questão, igualmente conhecido como hexapodo, é um manipulador paralelo constituído de seis braços de deslocamento linear, cada qual comandado por um acionador específico; uma junta universal conectando-se à base e uma junta rotativa conectando-se à parte móvel superior. Com a finalidade de alterar a posição da plataforma, esses acionadores agem de tal forma a modificar os comprimentos dos braços. Tal configuração permite a rotação da parte superior de acordo com os valores de deslocamento de cada acionador. Assim, o mecanismo possui seis graus de liberdade: três coordenadas em translação e três ângulos direcionadores, yaw, pitch and roll, (ângulos de Euler). Sua estrutura de malha fechada faz com que o sistema manipulador seja proporcionalmente mais rígido, quando comparados seu tamanho e seu peso com os mesmos de qualquer ligação robótica serial (Chang, Fu, Yu, & Huang, 2004).

Uma vez conhecidas as localizações de todas as juntas de cada uma das plataformas, a fixa e a móvel, e sabendo que é possível modificar e mensurar os comprimentos de cada braço, a movimentação em posição da plataforma superior com relação à base é igualmente passível de estudo. As estratégias existentes se dividem em duas categorias. A primeira, referente à cinemática inversa, visa determinar os comprimentos dos braços a partir da posição e da orientação desejadas da plataforma móvel. Esta estratégia aplicada ao robô paralelo é simples devido à unicidade de sua solução. Em contrapartida a segunda estratégia, referente à cinemática direta, parte dos comprimentos dos braços para encontrar a posição e a orientação da plataforma móvel. Neste caso contam-se mais de 40 possíveis soluções devido a característica singular da matriz Jacobiana resultante da arquitetura do manipulador paralelo (Ma & Angeles, 1991).

1.2 Objetivos do trabalho

A utilização industrial ou médica de um manipulador robótico pode muitas vezes exigir precisão e alto desempenho. Para tanto, não basta conhecer apenas o modelo cinemático do robô. Sem o seu modelo dinâmico, que considera toda a distribuição inercial de sua plataforma atuadora, não é possivel garantir que a força empregada aos braços acionadores seja suficiente para obter o deslocamento de cada braço, calculado pela cinemática inversa, para que o atuator atinja a posição objetivo.

Seguindo esses passos, o objetivo básico deste trabalho é desenvolver o modelo dinâmico da plataforma de Stewart para que seja possível projetar um controlador que, empregado ao modelo cinemático e dinâmico do manipulador, permita a correção de sua trajetória. O desenvolvimento do modelo dinâmico é um desafio do ponto de vista matemático uma vez que a arquitetura do mecanismo faz aparecerem singularidades computacionais. Por esse motivo, estudos dinâmicos vêm sendo feitos baseados na formulação Lagrangeana, na de Newton-Euler (Dasgupta & Mruthyunjaya, 1998) e no princípio do trabalho virtual (Lee, Song, Choi, & Ho, 2003).

1.3 Desenvolvimento do trabalho

O estudo dinâmico da Plataforma de Stewart foi inicialmente abordado utilizando a aproximação de Newton-Euler (Dasgupta & Mruthyunjaya, 1998). Por esta técnica, é feita a implementação de um controlador do tipo proporcional-derivativo (PD) e o resultado de simulações é mostrado a partir da codificação do modelo em Matlab™. A principal desvantagem levantada contra esta abordagem é o alto custo computacional que impossibilita a simulação dos movimentos da plataforma em tempo real.

Uma segunda abordagem de estudo dinâmico da Plataforma de Stewart apresentada neste trabalho, considera que as forças necessárias para mover os braços robóticos são proporcionais às supostas deformações que estes braços sofreriam para atingir o deslocamento desejado. Esta abordagem foi inspirada no trabalho de Schwartz (2003), que se baseia na Lei de Hooke e em propriedades mecânicas de tecidos biológicos para simular o comportamento destes tecidos quando submetidos a aplicações cirúrgicas. Neste caso, as forças necessárias ao acionamento dos braços cilíndricos são oriundas dos torques de motores CC de imã permanente (Butler & Amerongen, 1989). Para a dinâmica da estrutura como um todo, a inércia dos seis braços não é levada em consideração. O estudo é concentrado, portanto, na formulação cinemática estrutural com a participação inercial da parte móvel superior da plataforma, e no controle PID das tensões de cada um de seus seis motores de corrente contínua.

A partir da pesquisa bibliográfica de trabalhos publicados sobre o tema em questão, o trabalho aqui apresentado visa o deslocamento da plataforma a partir de informações fornecidas por um gerador de funções, as quais se referem às coordenadas do centro de sua parte móvel. A conseqüente movimentação dos seis braços é controlada pela variação de comprimento dos mesmos, sendo que o controlador utilizado é do tipo PID.

1.4 Estrutura do trabalho

O Capítulo 2 dessa dissertação apresenta uma pesquisa bibliográfica aprofundada de trabalhos relacionados a modelagem dinâmica e controle de robôs paralelos do tipo Plataforma de Stewart.

O Capítulo 3 tem como objetivo implementar o controle PD na Plataforma de Stewart, a qual teve toda sua estrutura móvel, braços e plataforma, modelados dinamicamente segundo a aproximação de Newton-Euler, através do:

- Estudo cinemático e dinâmico de um dos braços robóticos.
- Estudo cinemático e dinâmico da plataforma móvel.
- Apresentação de resultados das simulações realizadas.

O Capítulo 4 tem como objetivo implementar o controle PID na Plataforma de Stewart, a qual teve a dinâmica aproximada do ponto de vista de seus braços robóticos. Os deslocamentos dos braços robóticos foram calculados considerando que as forças necessárias às movimentações desejadas são proporcionais às deformações que elas causam nos cilindros. Para chegar a esse modelo, o capitulo foi dividido nos seguintes estudos:

- Desenvolvimento da cinemática da plataforma considerando as geometrias hexagonais da base e da parte móvel como aquelas do protótipo existente no Laboratório de Automação e Robótica – Unicamp (figura 2.11). O passo seguinte é a apresentação do modelo geométrico inverso, mais simples para o mecanismo em questão, em que os comprimentos de seus seis braços são determinados a partir do posicionamento central da plataforma superior.

- Modelagem do sistema de acionamento (motor elétrico) que será usado em cada braço do robô, apresentando sob a forma de diagrama de blocos a parte elétrica e mecânica do simulador implementado em Matlab/Simulink™

- Implementação da estrutura de controle do sistema de acionamento das juntas robóticas descrito anteriormente. A partir deste ponto, têm-se todos os requisitos necessários ao estudo dinâmico do mecanismo, muito importante para que se possa corrigir seu movimento através do controlador projetado.

- Equacionamento dinâmico do sistema em estudo, através da aproximação do cálculo das intensidades de força citadas anteriormente, que considera uma força necessária à deformação do braço ao supô-lo um corpo elástico (Módulo de Young).

- Apresentação do acoplamento completo do sistema, o qual consiste dos seguintes elementos: gerador de funções, cinemática inversa, estrutura de acionamento elétrica e modelo dinâmico.

No Capitulo 5, referente a validação computacional através de simulação, são realizados quatro estudos de casos caracterizados através de um deslocamento (vertical simples e em torno do eixo *Z*) e uma função senoidal (no eixo *Z* e em todos os seis graus de liberdade), onde em cada um deles, um gerador de funções forneceu um tipo de entrada desejada aplicada ao centro de massas da mesa superior. As respostas gráficas mostram o movimento desejado no centro da parte móvel, os comprimentos esperados de cada braço para tal objetivo, as forças necessárias em cada braço para o movimento impelido e o comprimento real de cada braço, resultante da ação do controlador do tipo PID. A atualização do sistema é feita em tempo real.

No Capitulo 6 desta dissertação de mestrado são apresentadas as principais conclusões deste trabalho e perspectivas futuras.

Finalmente, para melhor compreensão do leitor, os anexos apresentadam o modelo cinemático direto, a programação feita em Matlab/Simulink™, e artigos publicados em congressos e revistas, pelo autor relacionado, com o estudo desenvolvido nesta dissertação de mestrado.

Capítulo 2

Revisão Bibliográfica

2.1 Classificação dos sistemas robóticos

Os manipuladores robóticos podem ser classificados dependendo de vários critérios, dentre eles, o número de graus de liberdade, estrutura cinemática, geometria do espaço de trabalho e características do movimento. O presente trabalho será enfatizado à classificação de um manipulador robótico paralelo segundo a sua estrutura dinâmica. A figura 2.1 apresenta exemplos industriais de manipuladores do tipo paralelo e serial.

(a) Manipulador Paralelo: Plataforma de Stewart MPS (Andreff; Dallej; Martinet, 2007)

(b) Manipulador Serial KR-3 KS (KUKA robot Group)

Figura 2.1: Manipulador Paralelo e Serial.

Um manipulador pode ser classificado de acordo com a forma de sua estrutura cinemática, ou seja:

- Serial ou manipulador de cadeia aberta: estrutura cinemática possui a forma de cadeia cinemática aberta,

- Paralelo: constituído de cadeias cinemáticas fechadas,

- Hibrido: constituídas de cadeias cinemáticas abertas e fechadas.

A figura 2.2 apresenta um manipulador robótico composto na base de uma estrutura paralela com três graus de liberdade e um braço robótico com quatro graus de liberdade disposto sobre esta base.

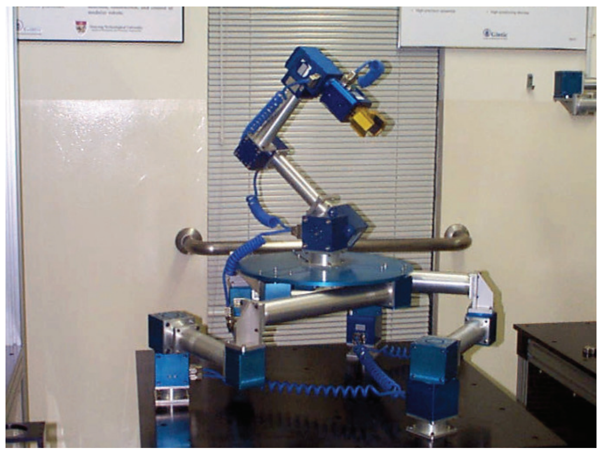


Figura 2.2: Manipulador hibrido.

2.2 Comparação entre um manipulador serial e paralelo

A comparação destes dois tipos de manipuladores é realizada basicamente em termos de suas características mecânicas e problemas de controle para determinar em que tipo de tarefas cada manipulador pode ser utilizado com vantagem.

Em termos mecânicos, os manipuladores seriais são constituídos de atuadores nas suas partes móveis, implicando em massas e momentos de inércia relativamente altos (sistema rígido). Nos manipuladores paralelos, todos os atuadores são montados próximos a base, possibilitando assim numa possível redução da massa nas suas partes móveis, implicando assim que os manipuladores paralelos apresentam características dinâmicas melhores em relação aos manipuladores seriais.

Em relação ao volume de trabalho, o dos manipuladores paralelos é muito menor que o de um manipulador serial. Em compensação, a estrutura mecânica de um manipulador paralelo é mais simples.

Para o controle de posição, a cinemática de um manipulador é um aspecto importante a ser considerado, onde a partir da cinemática direta, a posição e orientação do efetuador final são determinadas em função das variáveis articulares. O modelo cinemático inverso é necessário para o cálculo dos deslocamentos da juntas, quando são considerados os movimentos no espaço de trabalho, como por exemplo, o movimento do efetuador final ao longo de uma trajetória.

Um sensor de força pode ser utilizado a partir do conhecimento das forças e torques obtidos através do modelo estático direto. No controle de força no espaço de trabalho, o modelo estático inverso é requerido no cálculo dos torques e forças nos atuadores correspondentes aos momentos e forças no efetuador final. Os cálculos são muito mais simples, quando realizados de forma direta, sem necessidade de achar a matriz inversa. Assim, o controle no espaço de trabalho é mais fácil de ser realizado para os manipuladores paralelos, enquanto a localização do efetuador final é mais simples nos manipuladores seriais. Já o controle de força no espaço de trabalho nos manipuladores seriais é mais fácil de ser implementado que nos manipuladores paralelos.

Os erros de posicionamento nas articulações são acumulativos nos manipuladores seriais, enquanto que nos manipuladores paralelos, estes erros são a média de todos os erros. A capacidade de carga máxima nos manipuladores seriais é limitada para cada atuador que tem o torque de saída mínimo nos manipuladores seriais, enquanto que nos manipuladores paralelos a capacidade de carga é a soma das forças de todos os atuadores constituintes do manipulador.

Uma das principais diferenças entre manipuladores seriais e paralelos consiste no fato de que um manipulador paralelo possui articulações passivas não associadas a um atuador, enquanto que em um manipulador serial, uma junta sempre está associada a um atuador.

A partir das considerações apresentadas anteriormente, pode-se concluir que a utilização dos manipuladores seriais apresenta vantagens na realização de tarefas que requerem movimentos complexos e controle exato de força, como por exemplo: montagem de pequenos objetos com precisão. Por outro lado, os manipuladores paralelos podem ser aplicados para manipular objetos pesados em movimentos rápidos e sem um grande range de orientação.

Tabela 2.1: Comparação entre Manipuladores Serial e MPS

		MANIPULADOR SERIAL	MANIPULADOR PARALELO
MECANISMO	Inércia	Grande	Pequeno
	Volume de trabalho	Grande	Pequeno
	Aparência	Forma Antropomórfica	Base Estrutura
	Fabricação	Difícil	Fácil
CONTROLE	Controle de posição no espaço de trabalho	Difícil	Fácil
	Controle de força no espaço de trabalho	Fácil	Difícil
	Detecção de Força	Difícil	Fácil
	Erro de Posição	Acumulando	Médio
	Erro de controle de Força	Médio	Acumulado
	Perto de pontos singulares	Degeneração em controle de força	Diminuição de exatidão no posicionamento
		Grande movimento no atuador	Grande força no atuador
	Dinâmica	Complicada	Muito mais complicada

A tabela 2.1 apresenta uma comparação qualitativa das características mecânicas e modelagem dos manipuladores com arquitetura serial e paralela, onde podemos observar as principais diferenças entre essas duas arquiteturas, e as principais vantagens e desvantagens na utilização das mesmas. (Arai & Sheridan, 1988).

2.3 Manipuladores paralelos

Um robô paralelo é constituído de um efetuador com *n* graus de liberdade e de uma base fixa, interligados por, no mínimo, duas cadeias cinemáticas independentes. A atuação é realizada através de *n* acionadores simples.

Um manipulador paralelo generalizado é um mecanismo de cadeia cinemática fechada cujo efetuador está ligado à base através de várias cadeias cinemáticas independentes, podendo incluir mecanismos redundantes com mais atuadores que o número de graus de liberdade controlados do efetuador, bem como manipuladores trabalhando em cooperação. Segundo Merlet (2006), as principais características destes manipuladores são as seguintes:

- No mínimo duas cadeias suportam o efetuador. Cada uma destas cadeias contém, no mínimo, um atuador simples. Há um sensor apropriado para medir o valor das variáveis associadas com a atuação (ângulo de rotação ou movimento linear).

- O número de atuadores é o mesmo que o número de graus de liberdade do efetuador.

- A mobilidade do manipulador é zero quando os atuadores estão travados.

O mesmo autor também apresenta algumas justificativas no interesse da utilização desse tipo de mecanismo:

- Um mínimo de duas cadeias permite uma melhor distribuição da carga sobre as cadeias.

- O número de atuadores é mínimo.

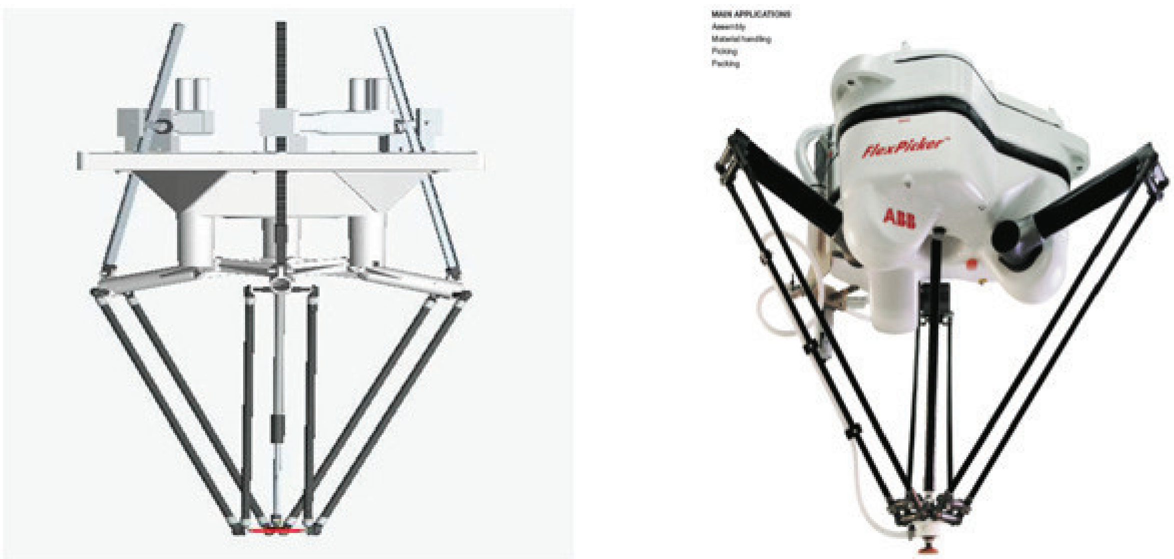
- O número de sensores necessários para o controle de malha fechada do mecanismo é mínimo.

- Quando os atuadores estão travados, o manipulador permanece em sua posição; isto é um aspecto importante de segurança para certas aplicações tais como robótica médica.

A literatura corrente apresenta uma série de arquiteturas mecânicas para robôs paralelos. Merlet (2006) classifica as mesmas em: robôs planares (manipuladores de 2 e 3 graus de liberdade) e robôs de movimento espacial. Nesta revisão são abordados os manipuladores paralelos espaciais.

2.3.1 Manipuladores de 3 graus de liberdade

Manipuladores com três graus de liberdade em translação têm sido desenvolvidos para aplicações de manipulação. O mais conhecido robô deste tipo é o Delta, inicialmente desenvolvido por *Clavel* da *École Polytechnique Federale de Lausanne* em 1988, e atualmente comercializado pela D*emaurex:* Divisão de Tecnologia de Embalagens do Grupo Bosch, e pela ABB (*FlexPicker*) (Merlet, 2006). As aplicações para este tipo de robô estão concentradas na área de embalagem, manipulação de alimentos, indústria farmacêutica, médica e eletrônica, conforme mostra a figura 2.3.



(a) Demaurex (máquinas de embalagem) **(b) Delta (ABB FlexPicker-IRB340)**

Figura 2.3: Manipuladores Industriais com 3 Graus de Liberdade - translação.

Outro manipulador com três graus de liberdade é o *orthoglide* (figura 2.4), que possui três atuadores lineares fixos montados ortogonalmente constituindo em três pernas idênticas e uma plataforma móvel que se movimenta no espaço cartesiano com uma orientação fixa (Pashkevich, Chablat, & Wenger, 2006).

Figura 2.4: Manipuladores com 3 Graus de Liberdade – orthoglide.

Manipuladores para orientação, ou seja, que permitem 3 rotações sobre um ponto, representa uma alternativa para o pulso com três juntas revolutas tendo eixos convergentes normalmente usados para robôs seriais. O mais simples gerador de movimentos rotativos sobre um ponto é constituído de um mastro central ligado à base e possuindo em sua extremidade uma junta bola-e-encaixe sobre a qual a base móvel é articulada. Um gerador de movimentos baseados em cadeias esféricas descrito por Asada foi desenvolvido por Gosselin e equipe na Universidade Laval, no Canadá, e chamado *The Agile Eye* para uma rápida orientação de câmeras (Merlet, 2006) (figura 2.5).

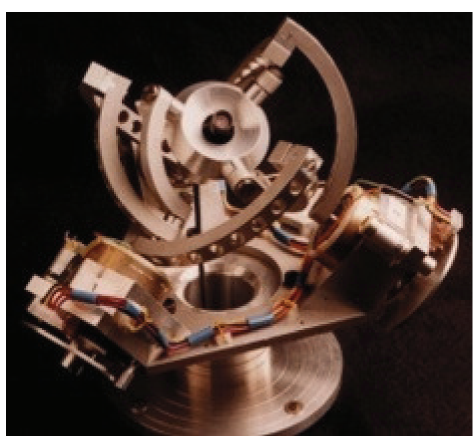
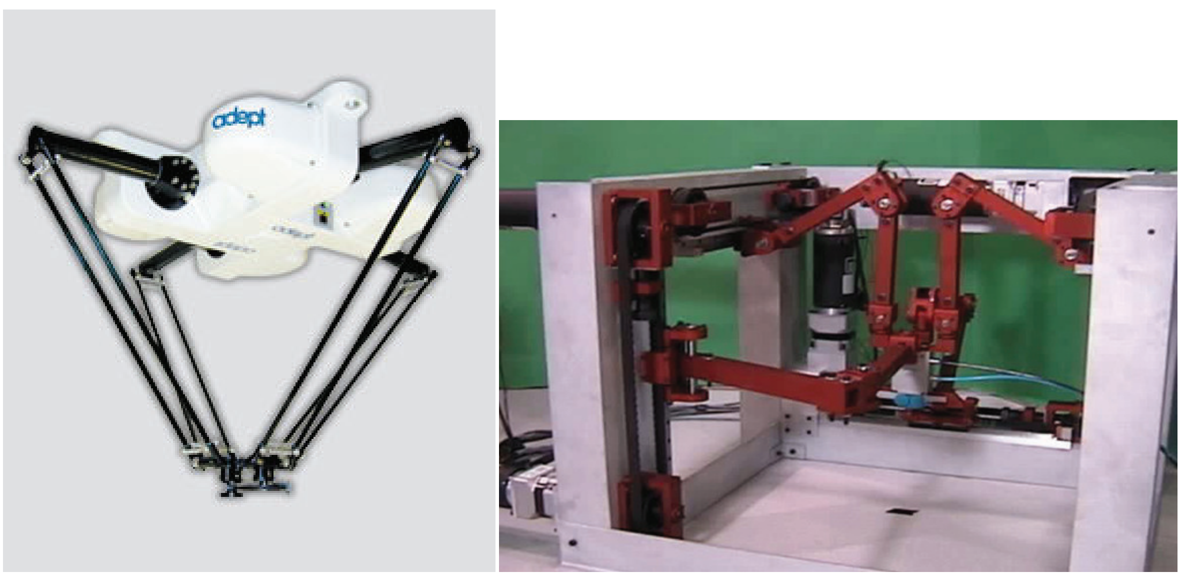


Figura 2.5: Manipuladores com 3 Graus de Liberdade – rotação.

2.3.2 Manipuladores de 4 graus de liberdade

Mecanismos com quatro graus de liberdade são encontrados há muito tempo na literatura. Em 1975, *Koevermans* apresentou um simulador de vôo usando atuadores lineares e com a plataforma móvel submetida a restrições passivas, onde os graus de liberdade eram três rotações e uma translação sobre o eixo Z (Koevermans, 1975).

Uma aplicação para esse mecanismo é o manipulador *The Adept Quattro™ s650 (Adept Technology)*, sendo projetado para ser utilizado em operações Pick-and-place, tais como operações de manufatura, manipulação de materiais e montagem a alta velocidade. O manipulador *Quadrupteron*, criado pelo Dr. François Pierrot e sua equipe (Nabat, Company, & Pierrot, 2006), foi inspirado no *orthoglide*, tem três graus de liberdade translacionais *x-y-z* e um de rotação no eixo *z* (figura 2.6).



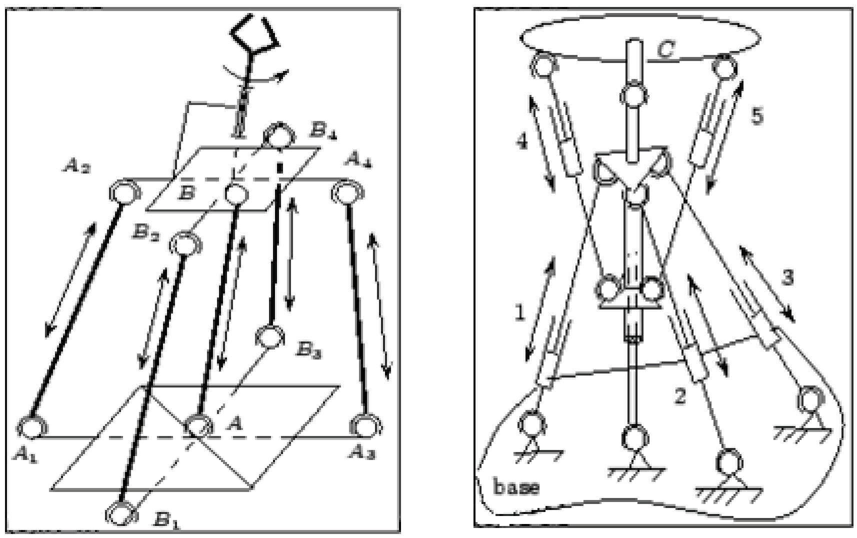
(a)The Adept Quattro™ s650 **(b) Quadrupteron (orthoglide)**

Figura 2.6: Manipuladores com 4 Graus de Liberdade – 3 translações e 1 rotação.

2.3.3 Manipuladores de 5 graus de liberdade

O mecanismo implementado por Austad, representa uma arquitetura híbrida para um manipulador de cinco graus de liberdade que poderia ser interpretado como uma mistura entre um robô serial e um robô paralelo onde, um primeiro dispositivo paralelo controla a posição de um ponto específico do efetuador, e um segundo mecanismo paralelo garante duas rotações da plataforma móvel.

Outro mecanismo que podemos destacar é o de *Zamanov*, que desenvolveu uma arquitetura baseada no acoplamento de dois robôs paralelos planares. Esta estrutura permite o controle dos graus de liberdade da plataforma com exceção da rotação normal à plataforma. Este último grau de liberdade pode ser controlado através de um atuador colocado sobre a plataforma (Merlet, 2006). A figura 2.7 exemplifica estes dois mecanismos paralelos.



a)Austad **b) Zamanov**

Figura 2.7: Manipuladores com 5 Graus de Liberdade.

2.3.4 Manipuladores de 6 graus de liberdade

O manipulador HEXA consiste de seis cadeias cinemáticas muito leves. Todas estas cadeias são idênticas e constituídas de um braço, uma barra e de uma articulação esférica. O manipulador HEXA é fixado no teto, onde os seis motores são fixos na base, e em cada um deles associado diretamente braço. Todas estas seis cadeias cinemáticas terminam no efetuador final da plataforma móvel. O HEXA possui a capacidade de operação em altas velocidades, com baixas cargas na plataforma móvel. Este manipulador foi desenvolvido pela *Toyota Machine Works* ((Pierrot, Uchiyama, & Dauchez, 1990).

Um Manipulador Plataforma de Stewart (MPS) é constituído por uma base fixa unida a uma base móvel por seis atuadores lineares, a distribuição espacial dos atuadores no MPS forma um octaedro (Stewart, 1965). O mecanismo MPS opera com baixas velocidades, altas cargas na plataforma móvel e tem uma rigidez alta (figura 2.8).

a) HEXA **b) Plataforma Stewart**

Figura 2.8: Manipuladores paralelos de seis graus de liberdade

2.4 Exemplos de aplicações dos manipuladores paralelos

Manipuladores paralelos têm sido freqüentemente utilizados nos últimos anos em diferentes aplicações no campo da engenharia, proporcionando soluções de posicionamento na indústria, na medicina e em variadas aplicações espaciais, descritas a seguir.

Aplicações Espaciais

Uma primeira categoria de manipuladores paralelos direcionados a simuladores para aplicações espaciais é a que desenvolve dispositivos terrestres para simulação de micro gravidade. Nesta categoria, temos como exemplo um simulador desenvolvido para testar o atracamento entre um ônibus espacial e uma estação espacial, em que o manipulador é instrumentado com sensores de força para que trabalhe de acordo com o modelo dinâmico de uma estação sujeita à um impacto (Merlet, 2006).

Outro protótipo existente é o robô *CKCM* estudado pela NASA (*Goddard Space Flight Center*) por Nguyen (1989) e sua equipe. O robô foi desenvolvido para o estudo de montagem robotizada no espaço. O uso de atuadores hidráulicos para robôs paralelos é sugerida para a simulação de movimentos do manipulador no espaço (Merlet, 2006).

Um robô hexápode também foi desenvolvido pelo *Instituto Max Planck* para ser usado com o telescópio *UKIRT* (*United Kingdom Infra-Red Telescope*) para movimentos lentos de foco. A companhia alemã Vertex também oferece uma estrutura paralela para uso com telescópios (Merlet, 2006).

Aplicações Médicas

Nesta área podemos destacar um endoscópio com dispositivo de fixação ativo usando um robô com 3 graus de liberdade atuados por fios que foi construído por Wendlandt & Sastry (1994). Outra aplicação médica para a estrutura paralela é o uso na assistência de pessoas deficientes para a movimentação de seus braços, como sugerido por Homma, Hashino, & Arai (1998). Estas estruturas também são utilizadas como suporte para microscópio e auxílio em cirurgias de precisão. A figura 2.9 apresenta algumas aplicações médicas utilizando manipuladores paralelos.

Figura: 2.9 Manipuladores Paralelos - Aplicações Médicas.

Aplicações Industriais

Atualmente, manipuladores paralelos são muito utilizados industrialmente devido às suas características de alta precisão de posicionamento e enorme rigidez. As principais aplicações destes manipuladores são em operações de montagem e desmontagem, manipulação de peças, soldagem a ponto e aplicações usando realimentação de força.

O modelo Delta representa o melhor exemplo de um manipulador de estrutura paralela usada industrialmente em tarefas de manipulação rápida de objetos. Nos dispositivos máquina-ferramenta de precisão um dispositivo paralelo do tipo plataforma de Stewart-Gough pode ser utilizado para posicionamento de peças mecânicas complexas durante a execução de operações de fresagem e usinagem em 3 e 5 eixos. A figura 2.10 apresenta algumas aplicações industriais utilizando esses manipuladores.

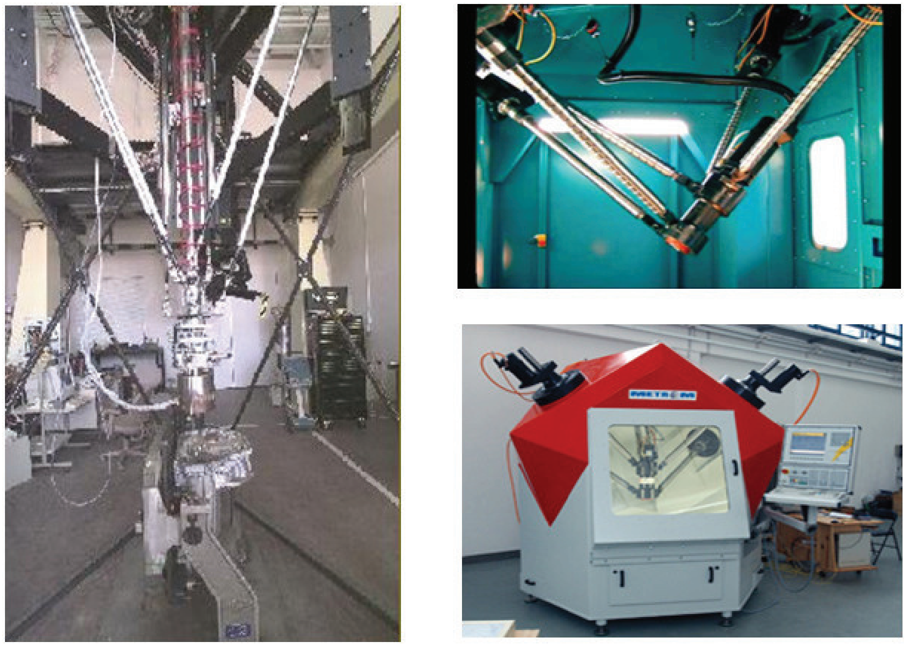


Figura 2.10: Manipuladores Paralelos - Aplicações Industriais.

Finalmente, a configuração do manipulador paralelo modelado neste estudo é ilustrada pela figura 2.11.

Figura 2.11: Exemplo real da plataforma estudada (não acionada – acionada).

2.5 Estudo dinâmico

Diversos trabalhos voltados à solução dinâmica desta plataforma foram publicados. Lee, Song, Choi, & Ho (2003) deriva as equações dinâmicas da plataforma de Stewart também se baseando na aproximação de Newton-Euler e no controle dinâmico inverso (IDC), também chamado de controlador de torque computacional, geralmente usado em manipuladores seriais. Além disso, o controle IDC é usado em combinação com o controle $H_{\infty.}$ O uso destes controladores combinados, com a aproximação dinâmica, mostra melhor resultado a medida que a velocidade do caminho percorrido aumenta, do que a formulação matemática completa da dinâmica submetida ao controlador do tipo PID.

O trabalho de Ma & Angeles, *Architecture Singularities of Platform Manipulators* (1991), classifica e estuda três tipos de singularidades encontradas em manipuladores paralelos: referentes à arquitetura, à configuração e à formulação. Focando na singularidade arquitetural, mostra que a estabilidade numérica da matriz Jacobiana é muito dependente do projeto arquitetural do manipulador, e que se a estrutura passa por uma região de singularidade, torna-se inapta ao trabalho já que falha no balanceamento das cargas sob a parte móvel.

Liu, Li, & Li (2000) trata os braços e a plataforma móvel como sub-estruturas independentes, e suas equações dinâmicas são derivadas usando a forma de Huston da equação de Kane. As equações dinâmicas são acopladas e as restrições que surgem das sub-estruturas são tratadas por multiplicadores de Lagrange. Esta aproximação é mais direta que a aproximação de Newton-Euler e o processo derivativo pode ser feito computacionalmente de forma automática.

Capítulo 3

Modelagem Dinâmica com Aproximação por Newton-Euler

3.1 Introdução

A formulação e a derivação de equações dinâmicas para manipuladores paralelos são consideravelmente complicadas devido a sua estrutura fechada e a restrições cinemáticas. No caso genérico, a formulação de Euler-Lagrange com imposição de restrições pelos multiplicadores de Lagrange resulta em um sistema de equações algébricas diferenciais relativamente complicadas de resolver. Ainda, tal formulação necessita de uma grande quantidade de computação simbólica a fim de encontrar derivadas parciais do Lagrangiano e uma grande quantidade de computação numérica para estimar tais derivadas. Por outro lado, a formulação de Newton-Euler não requer aproximação de derivadas de nenhuma função, como o Lagrangiano e, portanto, evita cálculos onerosos (Dasgupta & Mruthyunjaya, 1998).

Na formulação dinâmica de um manipulador serial de encadeamento aberto, a aproximação de Newton-Euler é restringida pelo cálculo da dinâmica inversa e não é muito favorecida pelo encadeamento fechado das equações dinâmicas. Contudo, no caso de manipuladores paralelos, a aproximação de Newton-Euler também pode ser usada pela vantagem de derivar a equações dinâmicas de encadeamento fechado. No corrente experimento, esta aproximação foi adotada por desenvolver as equações dinâmicas de encadeamento fechado da Plataforma de Stewart, o que é essencial para a dinâmica direta e para o projeto do sistema de controle.

Ainda, nesta aproximação, foi desenvolvida uma eficiente formulação da dinâmica inversa, modelando a dinâmica, gravidade e forças de atrito viscoso nas seis juntas prismáticas-esféricas-universais da Plataforma de Stewart, para uma geometria e distribuição inercial universais.

Nesta abordagem, além da aproximação de Newton-Euler, equações de equilíbrio são manipuladas para originar a forma fechada das equações dinâmicas inseridas no espaço de trabalho das seis juntas citadas acima. A transformação das equações dinâmicas em junta-espaço também é desenvolvida. Adicionalmente, a estrutura da Plataforma de Stewart que possui uma junta esférica em cada extremidade superior de seus braços, e que possui 6 graus de liberdade adicionais, pode ser considerada como um sistema de 12 graus de liberdade. Suas equações dinâmicas completas são originadas levando-se em consideração as rotações passivas dos braços como coordenadas adicionais.

Assim, ao longo deste terceiro capítulo, são originadas as equações de movimento para uma plataforma de Stewart cuja estrutura é composta de uma junta universal na extremidade do braço conectado à base, e de uma junta esférica na extremidade do braço conectado ao topo para cada um de seus seis braços. No entanto, será considerado que não ocorre rotação ao longo de nenhum dos eixos de seus braços. Primeiramente são consideradas a cinemática e a dinâmica de um dos braços, e a expressão para a força de restrição na junta esférica no topo do braço é originada como é mostrado a seguir:

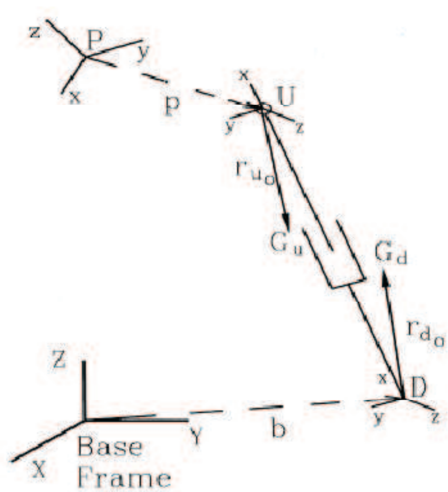


Figura 3.1: Detalhes de um braço.

3.2 Cinemática e dinâmica de um braço robótico

Para a cinemática e a dinâmica de um braço do manipulador, todas as variáveis exceto àquelas da cinemática da plataforma, ou seja, posição, velocidade e aceleração, pertencem a um braço arbitrário e são representados respectivamente por **b**, **p** e $\mathbf{a}_p$ O braço do manipulador mostrado na figura 3.1 tem como referência as coordenadas da base **X**, **Y** e **Z**, para a plataforma móvel os eixo coordenados **P**, e outros dois eixos coordenados de referência, um na parte inferior do braço, **D**, e outro na parte superior do braço, **U**. O eixo coordenado **D** tem sua origem em um ponto da base **X**, **Y** e **Z**, sendo *x* o eixo ao longo do braço, *y* o eixo ao longo do eixo de rotação da junta universal e *z* o eixo perpendicular aos eixos *x* e *y* de acordo com a regra da mão direita. O eixo coordenado **U** possui a mesma orientação de **D**, porém é localizado em um ponto do eixo coordenado da plataforma móvel **P**.

3.2.1 Análise de posição e velocidade

Nesta sessão é feito um resumo das analises de posição, velocidade e transformação inercial do braço.

Vetor representando o braço :

$$\mathbf{S} = \mathbf{q} + \mathbf{t} - \mathbf{b} \tag{3-1}$$

Comprimento do braço :

$$\mathrm{L} = \|\mathbf{S}\| \tag{3-2}$$

Vetor unitário na direção do braço :

$$\mathbf{s} = \mathbf{S}/\mathrm{L} \tag{3-3}$$

Velocidade do ponto de conexão na plataforma :

$$\dot{\mathbf{S}} = \omega \times \mathbf{q} + \dot{\mathbf{t}} \tag{3-4}$$

Velocidade de deslizamento entre as duas partes do braço :

$$\dot{L} = \mathbf{s} \cdot \dot{\mathbf{S}} \tag{3-5}$$

Velocidade angular do braço :

$$\mathbf{W} = \mathbf{s} \times \dot{\mathbf{S}} / L \tag{3-6}$$

3.2.2 Análise da aceleração

A aceleração no ponto de conexão do braço da plataforma é dado por :

$$\ddot{\mathbf{S}} = \ddot{\mathbf{t}} + \alpha \times \mathbf{q} + \omega \times (\omega \times \mathbf{q}) \tag{3-7}$$

Os termos na expressão acima que envolvem a aceleração linear $\ddot{\mathbf{t}}$ e a aceleração angular α são agrupados no vetor :

$$\mathbf{a}_p = \ddot{\mathbf{t}} + \alpha \times \mathbf{q} \tag{3-8}$$

O restante é denotado por :

$$\mathbf{U}_1 = \omega \times (\omega \times \mathbf{q}) , \tag{3-9}$$

que contém a parte da aceleração com dependência da velocidade.

A aceleração $\ddot{\mathbf{S}}$, expressa em termos da aceleração de deslizamento $\ddot{L}$ na junta prismática e da aceleração angular $\mathbf{A}$ do braço, é dada por :

$$\ddot{\mathbf{S}} = \ddot{L} \cdot \mathbf{s} + \mathbf{W} \times (\mathbf{W} \times \mathbf{S}) + 2 \cdot \mathbf{W} \times \dot{L} \cdot \mathbf{s} + \mathbf{A} \times \mathbf{S} \tag{3-10}$$

Fazendo o produto escalar de ambas as equações de $\ddot{\mathbf{S}}$ acima com $\mathbf{s}$ e considerando-se $\mathbf{s}\cdot\mathbf{W}=0$, obtém-se :

$$\ddot{L}=\mathbf{s}\cdot\mathbf{a}_p+u \,, \tag{3-11}$$

onde $u=\mathbf{s}\cdot\mathbf{U}_1+\frac{1}{L}(\dot{\mathbf{S}}-\dot{L}\mathbf{s})^2$.

Similarmente, aplicando o produto vetorial das mesmas equações com $\mathbf{s}$, considerando $\mathbf{s}\cdot\mathbf{W}=0$ e $\mathbf{s}\cdot\mathbf{A}=0$, obtém-se a aceleração angular do braço como :

$$\mathbf{A}=\frac{1}{L}\mathbf{s}\times\mathbf{a}_p+\mathbf{U}_2 \,, \tag{3-12}$$

sendo $\mathbf{U}_2=\frac{1}{L}\cdot(\mathbf{s}\times\mathbf{U}_1-2\dot{L}\mathbf{W})$.

As acelerações dos centros de gravidade das partes inferior e superior dos braços são respectivamente dadas por :

$$\mathbf{a}_d=\frac{1}{L}(\mathbf{s}\times\mathbf{a}_p)\times\mathbf{r}_d+\mathbf{U}_3 \tag{3-13}$$

e

$$\mathbf{a}_u=(\mathbf{s}\cdot\mathbf{a}_p)\cdot\mathbf{s}+\frac{1}{L}(\mathbf{s}\times\mathbf{a}_p)\times\mathbf{r}_u+\mathbf{U}_4 \,, \tag{3-14}$$

sendo $\mathbf{U}_3=\mathbf{U}_2\times\mathbf{r}_d+\mathbf{W}\times(\mathbf{W}\times\mathbf{r}_d)$ e $\mathbf{U}_4=u\mathbf{s}+\mathbf{U}_2\times\mathbf{r}_u+\mathbf{W}\times(\mathbf{W}\times\mathbf{r}_u)+2\dot{L}\mathbf{W}\times\mathbf{s}$.

3.2.3 Análise dinâmica

A análise dinâmica é feita considerando o equilíbrio rotacional do braço inteiro, incluindo as partes inferior e superior. A equação de Euler para o braço é dada por :

$$\begin{aligned}&-m_d\mathbf{r}_d\times\mathbf{a}_d-m_u\mathbf{r}_u\times\mathbf{a}_u+(m_d\mathbf{r}_d+m_u\mathbf{r}_u)\times\mathbf{g}-(\mathbf{I}_d+\mathbf{I}_u)\mathbf{A}\\&-\mathbf{W}\times(\mathbf{I}_d+\mathbf{I}_u)\mathbf{W}+M_u\mathbf{s}+\mathbf{S}\times\mathbf{F}_s-C_u\mathbf{W}-\mathbf{f}=0\end{aligned} \tag{3-15}$$

Na expressão, $\mathbf{F}_s$ representa a força exercida pela junta esférica sobre o braço e $\mathbf{f}$ é o momento de atrito viscoso sobre esta junta :

$$\mathbf{f}=C_s(\mathbf{W}-\omega) \tag{3-16}$$

Considerando-se o equilíbrio translacional, a equação de Newton para a parte superior na direção do eixo do braço é escrita por :

$$F+\mathbf{s}\cdot\mathbf{F}_s-C_p\dot{L}+m_u\mathbf{s}\cdot\mathbf{g}=0 \tag{3-17}$$

A equação que representa $\mathbf{F}_s$, resultado de substituições de equações acima, possui uma grande quantidade de termos, praticamente todos dependentes de $\mathbf{a}_p$. Para incorporar a força $\mathbf{F}_s$ nas equações dinâmicas da plataforma, todos esses termos são agrupados sob a forma $\mathbf{Qa}_p$, sendo $\mathbf{Q}$ uma matriz 3×3 a qual depende de parâmetros do braço e da configuração do manipulador. Para obter este resultado, duas regras algébricas são aplicadas :

Regra 1 :

$$\mathbf{s}\times\mathbf{a}_p=\tilde{\mathbf{s}}\mathbf{a}_p \quad \text{onde} \quad \mathbf{s}=\begin{bmatrix}s_x\\s_y\\s_z\end{bmatrix} \qquad \text{e} \qquad \tilde{\mathbf{s}}=\begin{bmatrix}0&-s_z&s_y\\s_z&0&-s_x\\-s_y&s_x&0\end{bmatrix}$$

Regra 2 :

$(\mathbf{x}\cdot\mathbf{a}_p)\mathbf{y} = (\mathbf{x}^T\mathbf{a}_p)\mathbf{y} = \mathbf{y}(\mathbf{x}^T\mathbf{a}_p) = (\mathbf{y}\mathbf{x}^T)\mathbf{a}_p$, sendo $\mathbf{x}$ e $\mathbf{y}$ vetores tridimensionais no espaço Cartesiano, dada $(\mathbf{y}\mathbf{x}^T)$ uma matriz 3×3.

As duas regras acima referem-se aos vetores nos sentidos geométrico e algébrico e provêem um meio consistente de transladar operações algébricas vetoriais em multiplicações matriciais, permitindo portanto o isolamento do vetor $\mathbf{a}_p$ e a conseqüente simplificação da equação para $\mathbf{F}_s$:

$$\mathbf{F}_s = \mathbf{Q}\mathbf{a}_p + \mathbf{V} - \mathbf{s}\mathbf{F}\,, \tag{3-18}$$

onde

$$\begin{aligned}\mathbf{Q} = {} & \left[m_u \cdot \left(1 + \frac{2\mathbf{s}\cdot\mathbf{r}_u}{L}\right) - \frac{m_d r_d^2 + m_u r_u^2}{L^2} \right]\mathbf{s}\mathbf{s}^T + \frac{m_d r_d^2 + m_u r_u^2}{L^2}\mathbf{E}_3 - \frac{m_u}{L}\left(\mathbf{s}\mathbf{r}_u^T + \mathbf{r}_u\mathbf{s}^T\right) \\ & - \frac{1}{L^2}\left[m_d(\mathbf{s}\times\mathbf{r}_d)(\mathbf{s}\times\mathbf{r}_d)^T + m_u(\mathbf{s}\times\mathbf{r}_u)(\mathbf{s}\times\mathbf{r}_u)^T - \tilde{\mathbf{s}}(\mathbf{I}_d + \mathbf{I}_u)\tilde{\mathbf{s}} \right]\end{aligned}\,, \tag{3-19}$$

e $\mathbf{E}_3$ uma matriz identidade 3×3.

Ao substituir a expressão $\mathbf{a}_p = \ddot{\mathbf{t}} + \alpha\times\mathbf{q}$ na equação de $\mathbf{F}_s$ e indexando-a, obtém-se

$$(\mathbf{F}_s)_i = \mathbf{Q}_i\ddot{\mathbf{t}} - \mathbf{Q}_i\tilde{\mathbf{q}}_i\alpha + \mathbf{V}_i - \mathbf{s}_i F_i \tag{3-20}$$

Os vetores $\mathbf{s}_i$, $\mathbf{q}_i$, $\mathbf{V}_i$ e a matriz $\mathbf{Q}_i$ devem ser computados separadamente para cada um dos 6 braços.

Similarmente, o momento de atrito viscoso no *i*-ésimo ponto de conexão com a plataforma é dado por :

$$\mathbf{f}_i = C_s \cdot (\mathbf{W}_i - \omega_i) \tag{3-21}$$

3.3 Cinemática e dinâmica da plataforma

Fazendo uso das expressões acima para a reação na junta $(\mathbf{F}_s)_i$ e o momento de fricção $\mathbf{f}_i\,(i = 1-6)$, as equações dinâmicas para a plataforma são desenvolvidas conforme abaixo :

Aceleração e inércia da plataforma :

Fazendo $\mathbf{R}_0$ o vetor posição do centro de gravidade da plataforma no eixo ordenado de referência local, o mesmo vetor pode ser expresso no eixo ordenado da base por :

$$\mathbf{R} = \Re \mathbf{R}_0 \tag{3-22}$$

A orientação $\Re$ da plataforma é representada pelos ângulos roll-pitch-yaw:

i) Roll - movimento em torno do eixo longitudinal.

ii) Pitch - movimento em torno do eixo transversal.

iii) Yaw - movimento em torno do eixo vertical.

$$\begin{aligned} \Re^T(\psi,\theta,\varphi) &= \mathrm{rot}(x,\varphi)\,\mathrm{rot}(y,\theta)\,\mathrm{rot}(z,\psi) \\ &= \left[\begin{array}{c|c|c} c\varphi\, c\theta & c\varphi s\theta s\psi - s\varphi\, c\psi & c\varphi\, s\theta\, c\psi + s\varphi\, s\psi \\ s\varphi\, c\theta & s\varphi s\theta\, s\psi + c\varphi\, c\psi & s\varphi\, s\theta\, c\psi - c\varphi\, s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{array}\right] \end{aligned} \tag{3-23}$$

$$\theta = \mathrm{ATAN2}\left[\frac{-n_Z}{c\varphi\, n_x + s\varphi\, n_y}\right],\ \varphi = \mathrm{ATAN2}\left[\frac{n_y}{n_x}\right],\ \psi = \mathrm{ATAN2}\left[\frac{s\varphi\, a_x - c\varphi\, a_y}{-s\varphi\, s_x + c\varphi\, s_y}\right]$$

$\mathbf{n} = \lfloor n_x \quad n_y \quad n_z \rfloor$, $\mathbf{s} = \lfloor s_x \quad s_y \quad s_z \rfloor$, $\mathbf{a} = \lfloor a_x \quad a_y \quad a_z \rfloor$: vetor ortonormal que descreve a orientação.

c e *s* representam as notações *cos* e *sin* respectivamente.

A aceleração do centro de gravidade é dado por :

$$\mathbf{a} = \alpha \times \mathbf{R} + \omega \times (\omega \times \mathbf{R}) + \ddot{\mathbf{t}} \tag{3-24}$$

O momento de inércia $\mathbf{I}_p$ da plataforma é transformado para o momento de inércia da base global por :

$$\mathbf{I} = \Re \mathbf{I}_p \Re^T \tag{3-25}$$

Equações dinâmicas do espaço de trabalho :

Assume-se que o sistema de forças externas atuando na plataforma é $\mathbf{F}_{ext}$, e $\mathbf{M}_{ext}$ o momento no sistema ordenado local. Os vetores podem ser transformados na base global pelas transformações rotacionais.

Considerando o equilíbrio de forças atuando na plataforma, a equação de Newton para a plataforma pode ser escrita por

$$-\mathbf{M}\mathbf{a} + \mathbf{M}\mathbf{g} + \Re \mathbf{F}_{ext} - \sum_{i=1}^{6} (\mathbf{F}_s)_i = \mathbf{0} \tag{3-26}$$

Os momentos com relação ao ponto de referência da plataforma, ou seja, a equação de Euler para a plataforma é

$$-\mathbf{M}\mathbf{R} \times \mathbf{a} + \mathbf{M}\mathbf{R} \times \mathbf{g} - \mathbf{I}\alpha - \omega \times \mathbf{I}\omega + \Re \mathbf{M}_{ext} - \sum_{i=1}^{6} [\mathbf{q}_i \times (\mathbf{F}_s)_i] + \sum_{i=1}^{6} \mathbf{f}_i = \mathbf{0} \tag{3-27}$$

Para a obtenção das equações dinâmicas completas para a plataforma, substitui-se as equações de $\mathbf{a}$ e $(\mathbf{F}_s)_i$ nas equações de Newton e Euler acima, obtendo-se :

$$\mathbf{J} \begin{bmatrix} \ddot{\mathbf{t}} \\ \alpha \end{bmatrix} + \eta = \mathbf{H}\mathbf{F} + \begin{bmatrix} \Re \mathbf{F}_{ext} \\ \Re \mathbf{M}_{ext} \end{bmatrix}, \tag{3-28}$$

onde

$$\mathbf{J} = \mathbf{J}_{plat} + \sum_{i=1}^{6} \mathbf{J}_i \; ; \qquad \eta = \eta_{plat} + \sum_{i=1}^{6} \eta_i \; ; \tag{3-29}$$

e

$$\mathbf{J}_{plat} = \begin{bmatrix} M\mathbf{E}_3 & -M\tilde{\mathbf{R}} \\ M\tilde{\mathbf{R}} & \mathbf{I} + M(R^2\mathbf{E}_3 - \mathbf{R}\mathbf{R}^T) \end{bmatrix} \qquad \mathbf{J}_i = \begin{bmatrix} \mathbf{Q}_i & -\mathbf{Q}_i\tilde{\mathbf{q}}_i \\ \tilde{\mathbf{q}}_i\mathbf{Q}_i & -\tilde{\mathbf{q}}_i\mathbf{Q}_i\tilde{\mathbf{q}}_i \end{bmatrix}, \tag{3-30}$$

$$\eta_{plat} = \begin{bmatrix} M\{\omega \times (\omega \times \mathbf{R}) - \mathbf{g}\} \\ \omega \times \mathbf{I}\omega + M\mathbf{R} \times \{(\omega \cdot \mathbf{R})\omega - \mathbf{g}\} \end{bmatrix} \qquad \eta_i = \begin{bmatrix} \mathbf{V}_i \\ \mathbf{q}_i \times \mathbf{V}_i - \mathbf{f}_i \end{bmatrix}, \tag{3-31}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 & \mathbf{s}_4 & \mathbf{s}_5 & \mathbf{s}_6 \\ \mathbf{q}_1 \times \mathbf{s}_1 & \mathbf{q}_2 \times \mathbf{s}_2 & \mathbf{q}_3 \times \mathbf{s}_3 & \mathbf{q}_4 \times \mathbf{s}_4 & \mathbf{q}_5 \times \mathbf{s}_5 & \mathbf{q}_6 \times \mathbf{s}_6 \end{bmatrix}, \tag{3-32}$$

$$\mathbf{F} = [F_1 \quad F_2 \quad F_3 \quad F_4 \quad F_5 \quad F_6]^T. \tag{3-33}$$

A matriz de inércia **J** mostra o alto grau de acoplamento dinâmico da plataforma de Stewart. A matriz **H** descreve a transformação entrada-saída da força, e finalmente o Jacobiano é dado pela transposição inversa dessa matriz, ou $\mathbf{H}^{-T}$.

Equações dinâmicas das juntas-espaço :

Acima definiu-se a equação dinâmica para simulação e controle do centro de gravidade da plataforma. Já para a simulação e controle de suas juntas no espaço (ou da entrada de informações) requerem-se as equações de movimento dessas juntas no espaço, as quais serão definidas a seguir.

Da equação de aceleração do deslizamento da junta prismática de cada um dos braços tem-se :

$$\ddot{L}_i = \mathbf{s}_i \cdot (\mathbf{a}_p)_i + u_i \tag{3-34}$$

que após a substituição de $\mathbf{a}_p$ torna-se

$$\ddot{L}_i = [\mathbf{s}_i{}^T \quad (\mathbf{q}_i \times \mathbf{s}_i)^T] \cdot \begin{bmatrix} \ddot{\mathbf{t}} \\ \alpha \end{bmatrix} + u_i \tag{3-35}$$

Combinando as expressões já definidas e separando-se o vetor de acelerações para todos os braços :

$$\begin{bmatrix} \ddot{\mathbf{t}} \\ \alpha \end{bmatrix} = \mathbf{H}^{-T}(\ddot{\mathbf{L}} - \mathbf{u}) \tag{3-36}$$

Finalmente as equações dinâmicas poder ser descritas por

$$\mathbf{H}^{-1}\mathbf{J}\mathbf{H}^{-T}\ddot{\mathbf{L}} + \mathbf{H}^{-1}(\eta - \mathbf{J}\mathbf{H}^{-T}\mathbf{u}) = \mathbf{F} + \mathbf{H}^{-1}\begin{bmatrix} \Re\mathbf{F}_{ext} \\ \Re\mathbf{M}_{ext} \end{bmatrix} \tag{3-37}$$

Nota-se que as equações de movimento das juntas da plataforma de Stewart no espaço são mais complicadas quando comparadas com as de seu centro de gravidade no espaço. Conseqüentemente, a análise dinâmica das juntas é computacionalmente mais onerosa que a análise dinâmica do centro de gravidade da mesa. Esta característica é típica de manipuladores paralelos, o que não acontece com manipuladores seriais, que apresentam maior facilidade de análise do ponto de vista de suas juntas.

3.4 Simulação e resultados

Para a simulação do desenvolvimento demonstrado ao longo deste capítulo, considerou-se a Plataforma de Stewart do tipo Universal-Prismatica-Esférica de 6 braços, com ambas a base e a plataforma representadas por hexágonos regulares. No trabalho de Ma & Angeles (1991), é mostrado que para esta arquitetura a matriz de transformação de forças **H** é sempre singular e que o manipulador é singular em todas as configurações. Isso significa que as forças atuantes nos seis braços não são capazes de suportar cargas arbitrarias na plataforma, significando que o grau de restrição do manipulador é perdido em algumas direções. Em outras palavras, o manipulador ganha alguns graus de liberdade e, mesmo que os atuadores sejam bloqueados, a plataforma continua em movimento dependendo das condições iniciais.

No exemplo a ser demonstrado a seguir, a posição inicial da plataforma foi escolhida de forma que ela fique exatamente em cima da base e com a mesma orientação.

Todos os braços são considerados idênticos e os parâmetros cinemáticos e dinâmicos foram escolhidos como a seguir.

Pontos da base:

$$\mathbf{b}_i = \begin{bmatrix} 0.30\cos\left(\pi \, {}^{i}\!/_{3}\right) & 0.30\mathrm{sen}\left(\pi \, {}^{i}\!/_{3}\right) & 0 \end{bmatrix}^T \text{ para } i = 1 \text{ a } 6$$

Pontos da plataforma:

$$\mathbf{p}_i = \begin{bmatrix} 0.15\cos\left(\pi \, {}^{i}\!/_{3}\right) & 0.15\mathrm{sen}\left(\pi \, {}^{i}\!/_{3}\right) & 0 \end{bmatrix}^T \text{ para } i = 1 \text{ a } 6$$

Vetores unitários ao longo dos eixos fixos de juntas universais:

$$\begin{bmatrix} \boldsymbol{\kappa}_1 & \boldsymbol{\kappa}_2 & \boldsymbol{\kappa}_3 & \boldsymbol{\kappa}_4 & \boldsymbol{\kappa}_5 & \boldsymbol{\kappa}_6 \end{bmatrix} = \begin{bmatrix} -0.8141 & 0.2308 & 0.9535 & 1.0000 & 0.7071 & -0.9535 \\ 0.2714 & 0.9231 & 0.2860 & 0.0000 & 0.7071 & 0.2860 \\ 0.0000 & 0.3077 & 0.0953 & 0.0000 & 0.0000 & -0.0953 \end{bmatrix}$$

Massas das partes inferior e superior de cada braço em Kg:

$$m_i = 3.0 \text{ e } m_s = 1.0$$

Centros de gravidade das partes inferior e superior de cada braço, em coordenadas locais:

$$\mathbf{r}_{i_0} = \begin{bmatrix} 0.4 & 0.14 & -0.18 \end{bmatrix}^T \text{ e } \mathbf{r}_{S_0} = \begin{bmatrix} -6.0 & -0.08 & 0.08 \end{bmatrix}^T$$

Momentos de inércia das partes inferior e superior de cada braço, em coordenadas locais:

$$\mathbf{I}_{i_0} = \begin{bmatrix} 0.010 & 0.005 & 0.007 \\ 0.005 & 0.002 & 0.003 \\ 0.007 & 0.003 & 0.001 \end{bmatrix} \text{ e } \mathbf{I}_{s_0} = \begin{bmatrix} 0.005 & 0.002 & 0.002 \\ 0.002 & 0.002 & 0.001 \\ 0.002 & 0.001 & 0.003 \end{bmatrix}$$

Massa da plataforma em Kg:

$$M = 40.0$$

Centro de gravidade da plataforma:

$$\mathbf{R}_0 = \begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$$

Momento de inercia da plataforma em coordenadas locais:

$$\mathbf{I}_p = \begin{bmatrix} 0.050 & 0.003 & 0.004 \\ 0.003 & 0.040 & 0.003 \\ 0.004 & 0.003 & 0.100 \end{bmatrix}$$

Coeficientes de atrito viscoso:

$$C_u = 0.0001$$
$$C_p = 0.001$$
$$C_s = 0.0002$$

Lei de Controle usada:

$$F_i = (K_p)_i \{(L_i)_0 - L_i\} - (K_v)_i \dot{L}_i \tag{3-38}$$

Os valores assumidos para os ganhos, abaixo, foram obtidos empiricamente:

$$(K_p)_i = 5000 \quad (K_v)_i = 500 \text{ para } i = 1 - 6$$

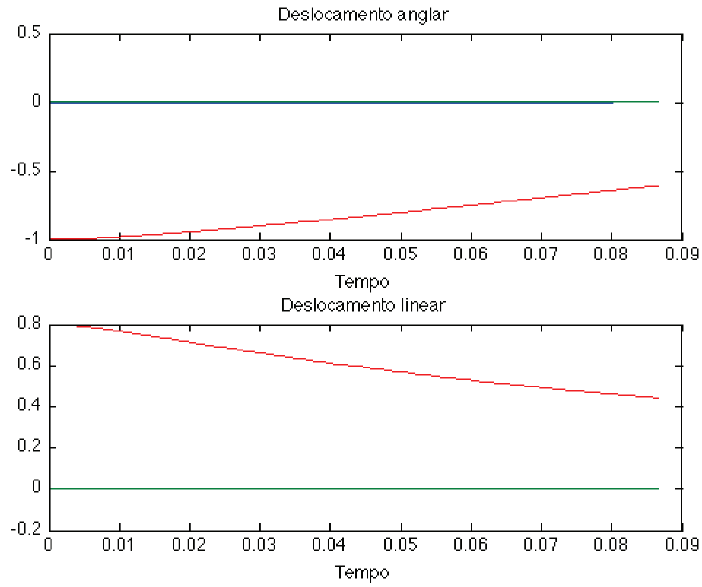


Figura 3.2: Deslocamentos angular e linear da Plataforma

A técnica de integração usada na implementação em Matlab, a *ode45*, baseada nas fórmulas de 4° e 5° ordens de Runge-Kutta, não apresentou bom resultado para a simulação em tempo real. O gráfico obtido (figura 3.2) é inconclusivo já que o complexo acoplamento matemático não permitiu um tempo de simulação adequado, limitando-o a 0.09 segundos, ponto em que é impossível encontrar tolerâncias de integração dentro do menor espaço de tempo possível para o cálculo. Outra técnica de integração também foi empregada usando o Matlab, a *ode15s*, um solucionador de ordem variável baseado em fórmulas de diferenciação numérica, e o mesmo resultado foi obtido.

Frente à dificuldade de integração das equações do modelo matemático da Plataforma de Stewart pela aproximação de Newton-Euler, no Capítulo 4 foi desenvolvida outra técnica que permite a simulação em tempo real do modelo considerando sua Lei de Controle.

Capítulo 4

Modelagem Dinâmica com Aproximação pela Lei de Hooke

4.1 Introdução

Analisadas as dificuldades de simular em tempo real o modelo matemático completo da Plataforma de Stewart pela aproximação de Newton-Euler, neste capitulo substitui-se o deslocamento linear e angular de cada um dos seis braços robóticos por uma deformação equivalente do aço do qual eles são constituídos. Essa deformação é proporcional à força que um motor de corrente continua precisa aplicar ao braço em questão para que ele tenha o deslocamento desejado, de forma a posicionar o centro de gravidade da plataforma em um local geométrico fornecido por um gerador de trajetória.

Neste caso, a parte inferior e a parte superior do manipulador paralelo são hexágonos irregulares, conforme é mostrado nas figuras abaixo.

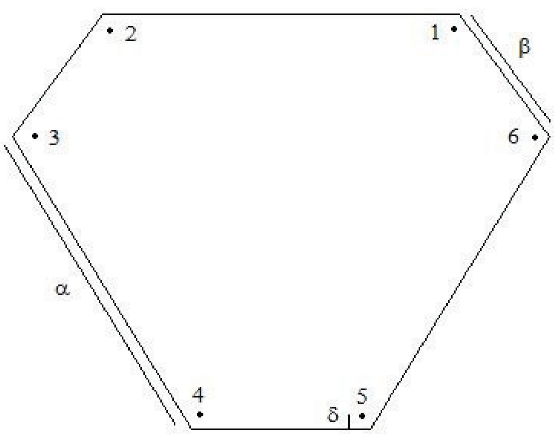


Figura 4.1: Pontos da parte inferior

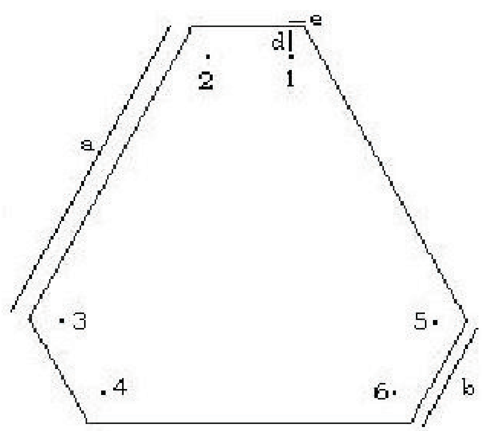


Figura 4.2: Pontos da parte superior

Conhecidas a geometria e a forma de acionamento do robô paralelo, é feito seu estudo cinemático e dinâmico, este ultimo permitindo a simulação do movimento considerando um controlador PID para corrigir os comprimentos dos braços e obter a posição desejada.

A seqüência do desenvolvimento matemático desta implementação é ilustrada no diagrama abaixo:

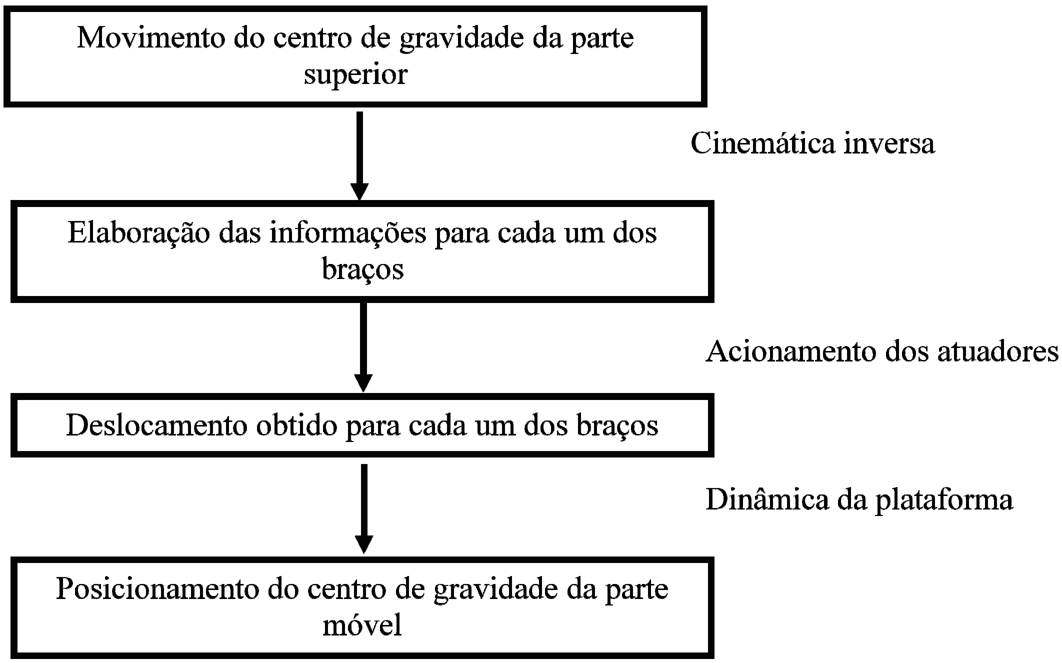


Figura 4.3: Diagrama do desenvolvimento matemático proposto

Nos tópicos a seguir são elaboradas as equações matemáticas e é feita a implementação do modelo e seu controle em Matlab/Simulink™ para enfim se analisar os resultados obtidos.

4.2 Modelagem cinemática

4.2.1 Estudo geométrico

Inicialmente determina-se matematicamente a geometria da plataforma. Para isso são atribuídos pontos numerados às extremidades dos cilindros com relação a um sistema de coordenadas considerado absoluto e fixo no centro da parte fixa e inferior. A numeração dos pontos das partes inferior e superior, com as medidas necessárias, é mostrada nas figuras 4.1 e 4.2.

As geometrias apresentada nas figuras acima resultam nas matrizes geométricas seguintes, em que PI é a representação matricial da parte fixa inferior, ou base, e PS é a representação matricial da parte móvel superior, ou plataforma.

$$\mathbf{PI} = \begin{bmatrix} \alpha/2-\varepsilon & (\alpha+\beta)/2*\cos(t)-\delta & 0 \\ -\alpha/2+\varepsilon & (\alpha+\beta)/2*\cos(t)-\delta & 0 \\ -1/2*\alpha+\varepsilon+(2*\varepsilon-\beta)*\cos(s) & (1/2*\alpha+1/2*\beta)*\cos(t)-\delta-(\beta-2*\varepsilon)*\cos(t) & 0 \\ -\beta/2+\varepsilon & -(\alpha+\beta)/2*\cos(t)+\delta & 0 \\ \beta/2-\varepsilon & -(\alpha+\beta)/2*\cos(t)+\delta & 0 \\ 1/2*\alpha-\varepsilon-(2*\varepsilon-\beta)*\cos(s) & (1/2*\alpha+1/2*\beta)*\cos(t)-\delta-(\beta-2*\varepsilon)*\cos(t) & 0 \end{bmatrix} \tag{4-1}$$

$$\mathbf{PS} = \begin{bmatrix} b/2-e & (a+b)/2*\cos(t)-d & h \\ -b/2+e & (a+b)/2*\cos(t)-d & h \\ -a/2+e+(2*e-b)*\cos(s) & -(a+b)/2*\cos(t)+d+(b-2*e)*\cos(t) & h \\ -a/2+e & -(a+b)/2*\cos(t)+d & h \\ a/2-e & -(a+b)/2*\cos(t)+d & h \\ a/2-e-(2*e-b)*\cos(s) & -(a+b)/2*\cos(t)+d+(b-2*e)*\cos(t) & h \end{bmatrix} \tag{4-2}$$

onde h é a altura da parte superior no início do movimento e $t=\frac{\pi}{6}$ e $s=\frac{\pi}{3}$.

4.2.2 Modelo geométrico inverso

O modelo geométrico inverso é um modelo mecânico utilizado em robótica por braços manipuladores. Permite determinar a configuração das ligações em função da configuração (posição e orientação) do centro da plataforma. Existem dois tipos de ligações, as rotoras e as ligações prismáticas, sendo que as ligações rotoras permitem movimentos de rotação caracterizados por um ângulo, e a ligação prismática permite movimentos de translação e são caracterizadas por uma distância.

O modelo geométrico inverso é utilizado por caracterizar o funcionamento de um braço manipulador. Concretamente, o modelo geométrico inverso permite calcular a posição de cada motor (ligação) do robô em função da posição e da orientação do órgão terminal (ferramenta).

O problema geométrico inverso consiste na determinação das novas coordenadas das extremidades superiores dos cilindros após um movimento aleatório, sendo que as referidas extremidades são fixas em juntas esféricas. O movimento da parte superior, rígida por sua vez é, portanto, composto de uma rotação e de uma translação. Para obter os novos vetores de posição da parte superior, é necessário aplicar uma transformação à sua matriz de posições iniciais, ou seja, multiplicá-la por uma matriz de rotação e em seguida somar o resultado a uma matriz de translação. Seja $\mathbf{X}_i$ uma função de $\mathbf{L}_i$:

$$\mathbf{X}_i = f(\mathbf{L}_i) \tag{4-3}$$

com $\mathbf{L}_i = (L_1 \quad L_2 \quad \cdots \quad L_6)$ o vetor de posições lineares das juntas e $\mathbf{X}_i = (X \quad Y \quad Z \quad \psi \quad \theta \quad \varphi)$ o vetor posição-orientação de um ponto da plataforma.

A matriz de rotação é uma função dos ângulos de Euler, o que significa que a rotação em torno dos eixos segue a ordem XYZ. O método aplicado, de obter a rotação do vetor posição dos pontos e não do sistema, é devido ao fato de que se deve inverter a matriz de rotação em torno de X, Y e Z respectivamente. O inverso desta matriz produto é sua transposta, onde a matriz de rotação $\Re$ é dada pela equação (3-23).

O comprimento solicitado ao cilindro – tamanho original somado ao deslocamento – é a norma do segmento, que vai da extremidade inferior até a extremidade superior, sendo esta última determinada pela transformação de coordenadas.

Desta forma, sabendo-se o comprimento de cada um dos seis cilindros na configuração inicial, dado por:

$$l_k = \sqrt{\sum_{j=1}^{3} (PS^{kj} - PI^{kj})^2} \,, \tag{4-4}$$

k = 1,2,...6; j são os componentes diretores x, y e z de cada cilindro,

o comprimento de cada cilindro após um movimento é dado por :

$$l_k + \Delta l_k = \sqrt{\sum_{j=1}^{3} [\Re_j(\theta,\rho,\psi) PS^k - PI^{jk}]^2}, \qquad (4\text{-}5)$$

onde $\Re_j$ é a j-ésima linha da matriz de rotação e, conforme definido acima, P^{kj} é a j-ésima coordenada do ponto P^k. A estrutura de blocos implementada é mostrada na figura 4.4.

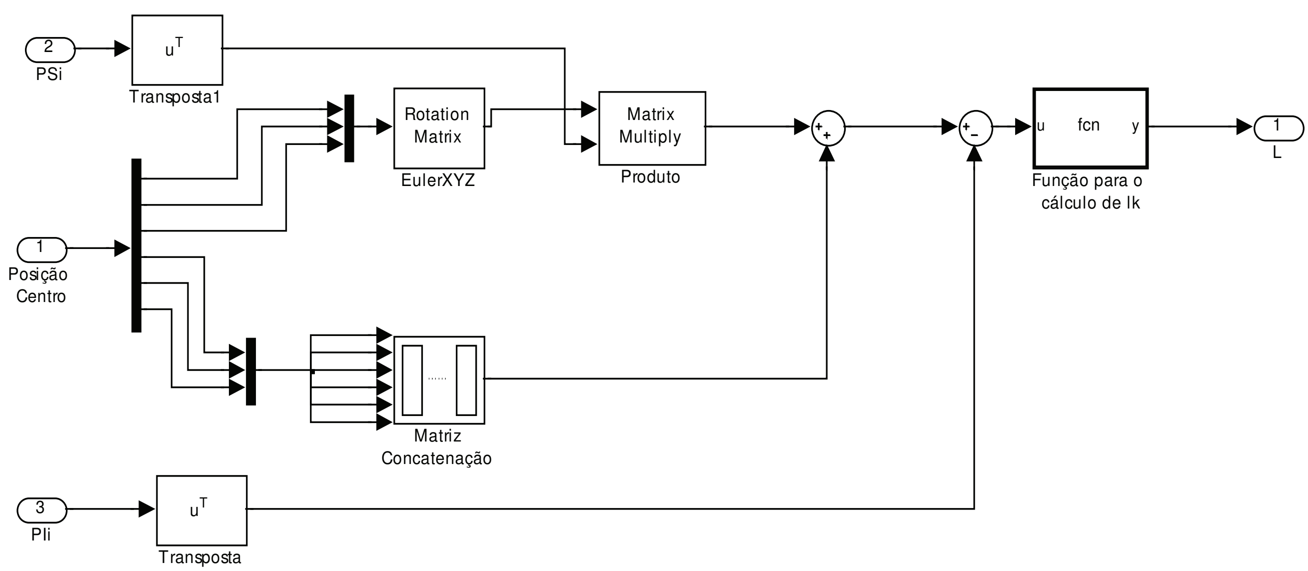


Figura 4.4: Modelo cinemático inverso

Deve-se observar que a configuração inicial da mesa não deve ser tal que os deslocamentos iniciais dos eixos sejam nulos, salvo se o movimento solicitado possa ser atendido apenas pela translação da mesa paralela.

4.3 Acionamento dos atuadores por motores de corrente contínua

Na plataforma real, seis atuadores hidráulicos transmitem o movimento para seis eixos, permitindo mover a base superior. Os sensores de posição situados em cada motor informam a posição associada aos ciclos de acionamento. Para a seqüência do desenvolvimento, devido à ausência de dados precisos sobre as características dos motores hidráulicos, um modelo de motor de corrente contínua foi utilizado, sendo que tal modelo foi usado em cada um dos seis braços com as características adequadas para a plataforma considerada. A vantagem principal das máquinas de corrente contínua consiste em sua adaptação simples aos meios, permitindo regular ou variar suas velocidades, seus torques e suas direções de rotação (Butler & Amerongen, 1989).

Cada eixo é formado, portanto, de um motor de corrente contínua, um sistema de transmissão e um controlador PID que atua na regulação da tensão de cada motor. Para o motor propriamente dito, as três equações clássicas são as seguintes:

$$u(t) = L_{mot}\frac{d\,i(t)}{d\,t} + R_{mot}\,i(t) + K_b\frac{d\,\theta_m(t)}{d\,t}, \tag{4-6}$$

$$T_m(t) = J_{eq}\frac{d^2\,\theta_m(t)}{d\,t^2} + B_{eq}\frac{d\,\theta_m(t)}{d\,t}, \tag{4-7}$$

$$T_m(t) = K_T\,i(t), \tag{4-8}$$

sendo $T_m(t)$ o torque fornecido pelo motor, $\theta_m(t)$ a posição angular da árvore motora, $i(t)$ a corrente induzida, L_{mot}, R_{mot} a indutância e a resistência induzidas, respectivamente, J_{eq}, B_{eq} a inércia e os atritos viscosos fornecidos à árvore motora.

A transformação do movimento de rotação da árvore motora em um movimento de translação dos eixos é efetuada por intermédio de uma rosca de ganho K_e:

$$K_e = \frac{p}{2\pi}, \tag{4-9}$$

onde *p* é o passo da rosca.

O bloco esquemático do funcionamento do motor é dado abaixo:

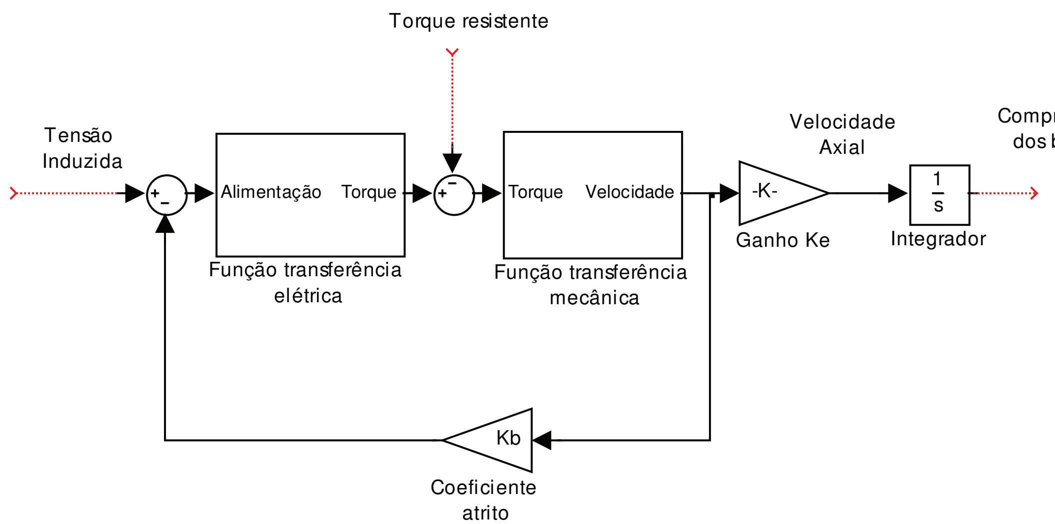


Figura 4.5: Modelo motor de corrente contínua

Em detalhes, as partes elétricas e mecânicas (seis funções idênticas para os seis eixos considerados).

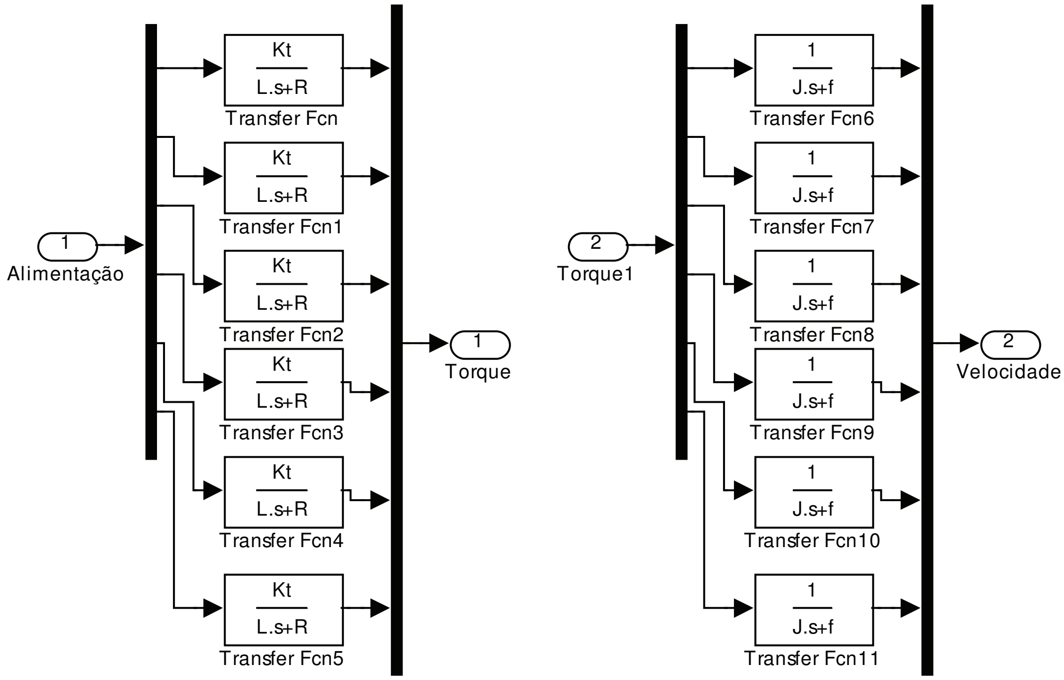


Figura 4.6: Partes elétricas e mecânicas para os seis eixos

Os parâmetros utilizados pelos motores considerados são listados abaixo:

Tabela 4.1 : Parâmetros do motor e da rosca de passo

L (H)	0.0065
R (Ω)	4.2
K_t (Nm/rad)	1.43
K_b (V/rad*sec^{-1})	0.8594
P(W)	0.01
J (Nm*sec^{2})	0.0083
f(Nm/rad*sec^{-1})	0.0035

4.4 Estrutura de acionamento e controle dos eixos

Cada motor no eixo é servido em posição por uma estrutura de regulação clássica, que faz intervir os controladores PID. Complementado o esquema da figura 4.6, o controlador PID é acoplado ao motor de corrente continua, conforme ilustra a figura abaixo. O significado do torque resistente será explicado adiante.

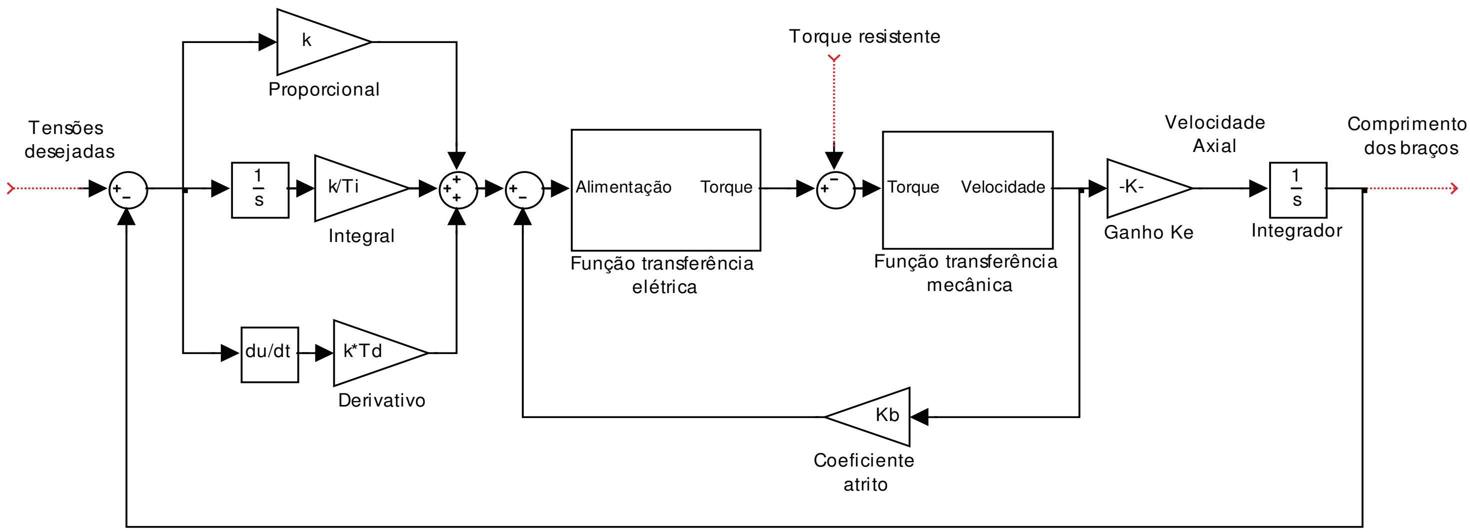


Figura 4.7: Estrutura de alimentação dos motores dos eixos

Os ganhos de sintonia para o controlador PID foram determinados experimentalmente pela resposta ao degrau, usando as regras de Ziegler-Nichols: $k = 5600, T_i = 0{,}5\,s, T_d = 0{,}0333\,s$, garantem ao sistema boa estabilidade. O diagrama de Bode para os parâmetros fornecidos acima é mostrado em seguida, informando, pela elevada margem de fase, a boa regulação do controlador (figura 4.8).

A estrutura de acionamento definida no corrente capítulo será acoplada ao modelo dinâmico da plataforma apresentado no próximo tópico.

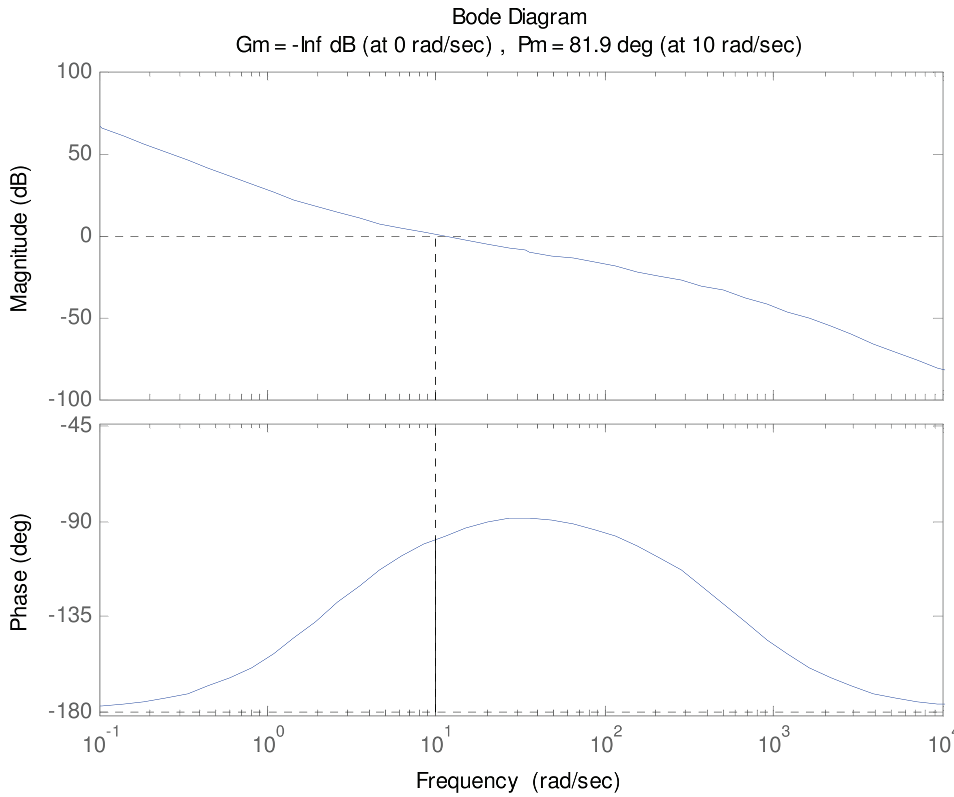


Figura 4.8: Diagrama de Bode da malha aberta corrigida (em posição)

4.5 Modelagem dinâmica

Colocar em prática uma plataforma com boas propriedades necessita da implantação de uma estrutura de comando. Contudo, controlar eficientemente a plataforma de Stewart é um verdadeiro desafio devido à característica fortemente não linear da parte dinâmica, das incertezas paramétricas e da cinemática complexa.

O desenvolvimento matemático das etapas seguintes descreve a elaboração do modelo dinâmico da plataforma utilizando blocos Simulink™, os quais foram acoplados conforme a figura 4.9:

Da equação (4-7), para cada eixo ($j = \overline{1,6}$), o equacionamento dinâmico fornece uma relação da forma:

$$T_{mj}(t) - D(\Delta L_j) = J_{eq} \frac{d^2 \theta_{mj}(t)}{dt^2} + B_{eq} \frac{d\theta_{mj}(t)}{dt} , \tag{4-10}$$

em que T_{mj} é o torque fornecido ao motor e o termo $D(\Delta L_j)$ é o troque resistente induzido pela plataforma, cujo desenvolvimento será detalhado a seguir.

A dinâmica da plataforma foi aproximada considerando que cada eixo pode ser representado por uma rigidez equivalente, a qual é dada por um cilindro de sessão S e de comprimento h_a:

$$K_a = \frac{S\,E}{h_a} , \tag{4-11}$$

E é o módulo de Young intervindo pelos esforços de compressão.

Os valores numéricos escolhidos são os seguintes:

$$\begin{aligned} &S = \pi\,(0{,}2)^2 \text{ m}^2 \\ &h = 0{,}3 \text{ m} \\ &E = 250\,000 \text{ MPa (caso do aço)} \end{aligned}$$

Desta forma, a parte dinâmica da plataforma, representada pela equação de movimento com o tensor de inércia, foi substituída por um termo de rigidez, ou seja, a dinâmica dos motores e da plataforma foi substituída por uma subtração, na equação dinâmica de cada motor, de um termo de torque resistente que é proporcional ao comprimento do cilindro ligado ao motor. Assim, o torque de cada motor, produzido por sua parte elétrica, sustenta sua inércia mecânica e o torque resistente, o qual aumenta quando o cilindro que ele representa se alonga e diminui quando o cilindro recua. O termo $D(\Delta L)$ que aparece na equação dinâmica precedente é justamente este torque resistente. O esquema de blocos abaixo resume tal aproximação.

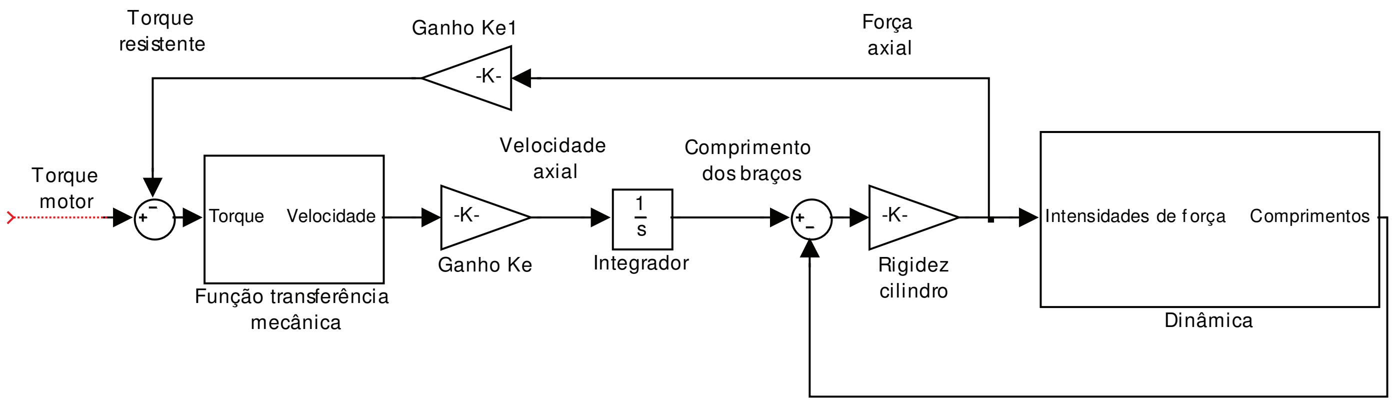


Figura 4.9: Rigidez associada à parte dinâmica

O objetivo é, portando, o de modelar o bloco dinâmico acima, ou seja, representar a relação de esforço ao longo dos eixos.

A primeira etapa consiste em calcular os vetores em *X*, *Y* e *Z* das direções de cada um dos seis braços e em seguida suas normas Euclidianas, função que dá aos vetores um valor positivo, considerando sucessivamente:

$$\mathrm{U}^{kj} = \mathrm{PS}_n^{kj} - \mathrm{PI}_i^{kj}\,, \tag{4-12}$$

$$\mathrm{PS}_n = \text{n-ésimo movimento da plataforma superior,}$$

$$k = \overline{1,6} \;\text{ e }\; j = \overline{1,3}\,,$$

e para a norma :

$$N = \sqrt{\sum_{j=1}^{3} \left(\mathrm{U}^{kj}\right)^2} \tag{4-13}$$

A matriz **U** normalizada é, portanto:

$$\mathbf{Y} = \frac{\mathbf{U}_{kj}}{N} \tag{4-14}$$

$k = 6$ e $j = 3$,

Têm-se assim as direções das forças. Com o objetivo de calcular a força resultante, multiplicam-se as intensidades das forças pelas direções $\mathbf{Y}$ determinadas acima. É importante observar que o cálculo das intensidades de força foi feito de forma aproximada, considerando o valor necessário para deformar o braço, supondo corpos elásticos, via Módulo de Young (Schwartz, 2003).

$$\mathbf{i_F} = \frac{SE\Delta\mathbf{l}}{h_a} \tag{4-15}$$

Assim, $\mathbf{i_F}$ é o vetor 1×6 das intensidades de força. Ainda, $\Delta\mathbf{l}$ é o deslocamento do cilindro, subtraído do movimento linear imposto por cada motor, resultando nos componentes de força sobre os eixos *X*, *Y* e *Z*:

$$\sum \mathbf{f} = \mathbf{i_F} \times \mathbf{Y}, \tag{4-16}$$

onde $\sum \mathbf{f}$ é o vetor força 1×3.

A divisão pelo inverso da massa da plataforma, de 5000 kg neste caso, fornece as componentes de aceleração axial de cada eixo:

$$\ddot{\mathbf{X}} = \frac{\sum \mathbf{f}}{\mathrm{m}} \tag{4-17}$$

Em seguida é necessário determinar a aceleração angular do centro de gravidade da parte superior da plataforma. A matriz $\mathbf{U}$ permite especificar o ponto médio da parte superior, definido por seus componentes sobre os eixos *X*, *Y* e *Z*:

$$c_j = \frac{\sum_{k=1}^{6} \mathrm{U}^{kj}}{6} \quad \text{(4-18)}$$

Seja $c = \begin{bmatrix} c_x & c_y & c_z \end{bmatrix}$ o vetor para o ponto médio, o vetor abaixo determina a distância de cada extremidade da plataforma ao ponto médio:

$$\mathbf{W}_k = \begin{bmatrix} \mathrm{U}_x^k - c_x & \mathrm{U}_y^k - c_y & \mathrm{U}_z^k - c_z \end{bmatrix} \quad \text{(4-19)}$$

A matriz $\mathbf{W}$ é uma matriz 6×3 que corresponde aos braços de alavanca utilizados para calcular os momentos.

Para que seja possível fazer a multiplicação vetorial entre o vetor das forças e o dos braços de alavanca a fim de satisfazer a equação do torque:

$$\mathbf{T} = \mathbf{f} \times \mathbf{W}, \quad \text{(4-20)}$$

convém transformar o vetor de intensidade de força em uma matriz diagonal de intensidades de força, assim :

$$\mathbf{f} = \mathbf{i}_{\mathbf{F}} \mathbf{Y} \quad \text{(4-21)}$$

Para solucionar a incompatibilidade matricial do produto vetorial, $\mathbf{f} \times \mathbf{W}$, utiliza-se o bloco Simulink™ *Matrix Concatenation* para as duas entradas, e o bloco Simulink *Matlab function* chama em seguida a função *cross_m_sum(u),* que faz corretamente o produto vetorial.

Para o cálculo da matriz de inércia da parte superior, considerou-se que esta possui a geometria de um hexágono sólido e chato na direção *Z* e de um cilindro sólido de altura igual a sua espessura nas direções *X* e *Y*.

Assim, sobre o eixo *Z*:

$$I_z = \frac{m}{6}\frac{\sum_{n=1}^{N}\|\mathbf{p}_{n+1}\times\mathbf{p}_n\|\left(\mathbf{p}_{n+1}\mathbf{p}'_{n+1}+\mathbf{p}_{n+1}\cdot\mathbf{p}_n+\mathbf{p}_n\mathbf{p}'_n\right)}{\sum_{n=1}^{N}\|\mathbf{p}_{n+1}\times\mathbf{p}_n\|} \tag{4-22}$$

Sobre os eixos X e Y :

$$\mathrm{I_x} = \mathrm{I_y} = \frac{1}{12}\mathrm{m}\left(3\mathrm{r}^2 + \mathrm{h}^2\right) \tag{4-23}$$

A matriz de inércia é, portanto:

$$\mathbf{I} = \begin{bmatrix} \mathrm{I_x} & 0 & 0 \\ 0 & \mathrm{I_y} & 0 \\ 0 & 0 & \mathrm{I_z} \end{bmatrix} \tag{4-24}$$

O momento cinético é calculado utilizando a matriz de inércia e a matriz de rotação usada anteriormente, mas com ângulos que se modificam a cada instante da simulação:

$$\boldsymbol{\sigma} = \Re(\theta,\rho,\psi)_n \mathbf{I}^{-1}\Re^T(\theta,\rho,\psi)_n \tag{4-25}$$

Por fim calcula-se os componentes da aceleração angular com a seguinte equação:

$$\ddot{\boldsymbol{\Theta}} = \boldsymbol{\sigma}\mathbf{T} \tag{4-26}$$

O esquema completo desta etapa é mostrado na figura 4.10.

Nota-se que é preciso ainda dois integradores para obter sucessivamente as velocidades e as posições lineares e angulares do ponto médio considerado (centro de gravidade da parte superior). Será necessário também reconstituir os comprimentos dos eixos correspondentes pelo sistema que representa a cinemática inversa, sistema este definido no tópico *4.2.2 Modelo geométrico inverso*.

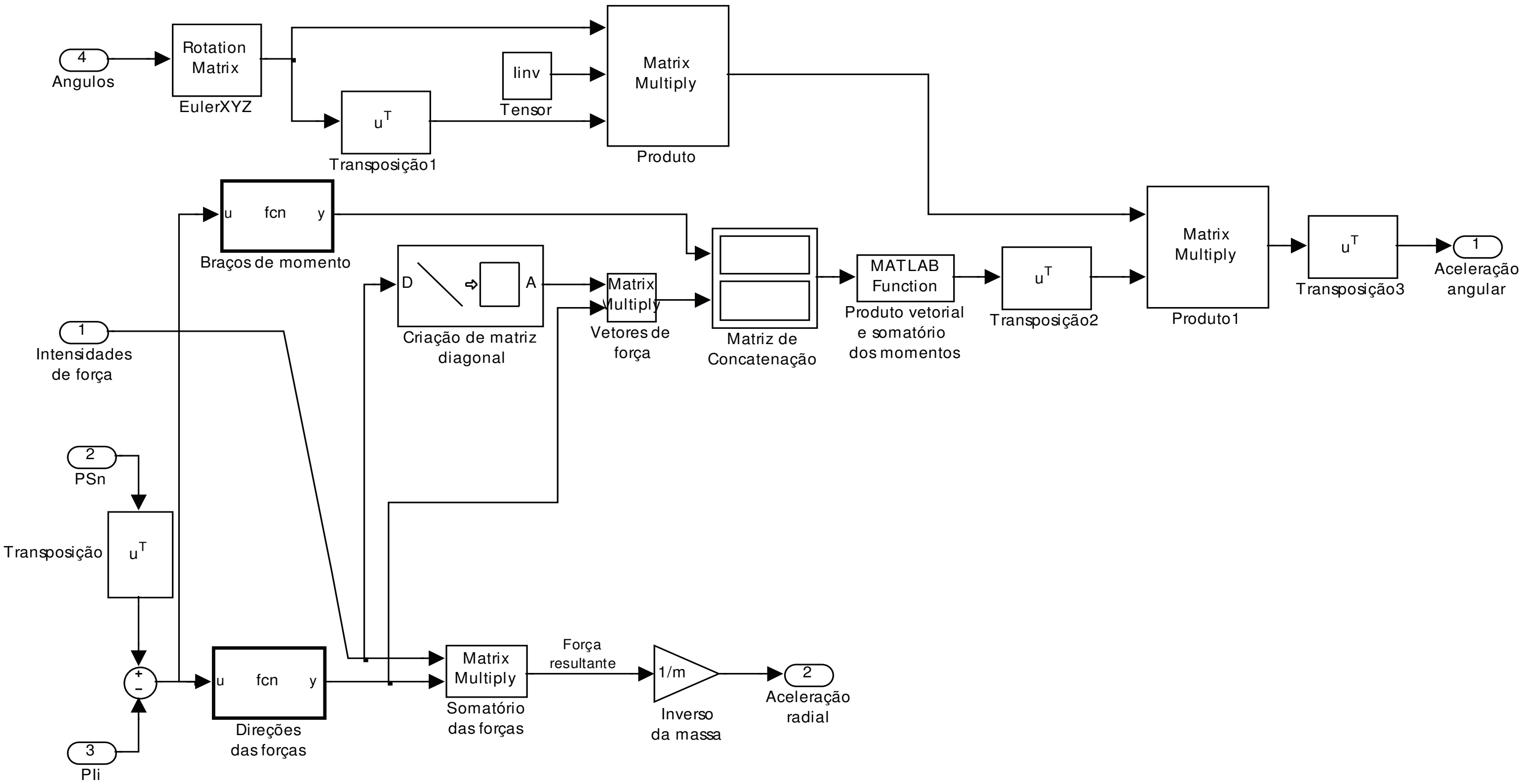


Figura 4.10: Modelo dinâmico

4.6 Acoplamento completo do sistema

A implementação do sistema completo consiste em conectar os módulos de sistema analisados nos tópicos anteriores (cinemática inversa, acionamento dos atuadores, rigidez do ligamento, dinâmica da plataforma) de modo a obter os comprimentos de cada eixo, fazendo a mesa superior seguir a trajetória desejada para o seu centro de gravidade em termos de posição e orientação. A estrutura completa é dada pela figura 4.11.

Para atingir o objetivo acima, é necessário converter os dados de entrada relacionados à trajetória do centro de gravidade em deslocamentos dos braços do hexapode, o que é feito pelo sistema *Cinemática Inversa 1* da figura 4.11. Em seguida, dos valores de comprimento dos braços obtidos, os valores de deslocamento são fornecidos, através do acionamento dos seis motores (supondo-os perfeitamente desacoplados), para o bloco *PID*. Estes valores são subtraídos dos valores atuais para que os esforços necessários sejam calculados.

O torque resistente, definido no tópico *4.5*, é representado na figura 4.11 pela entrada nomeada *Ciclo* do bloco *Motor CC*.

A saída do bloco *Rigidez* são as seis intensidades de força calculadas pela equação (4-15). As mesmas intensidades, transformadas pelo ganho do passo da rosca, fornecem o torque resistente, o qual é subtraído do torque da parte mecânica do motor, equação (4-10).

A partir das intensidades de força, da nova configuração da parte superior da plataforma e de sua parte inferior, calcula-se os novos dados de entrada, que são a rotação e o deslocamento. As novas entradas são usadas pelo bloco *Cinemática Inversa 2* para que sejam feitas as comparações de comprimentos e o sistema possa atingir em seguida a posição de equilíbrio correspondente à trajetória desejada.

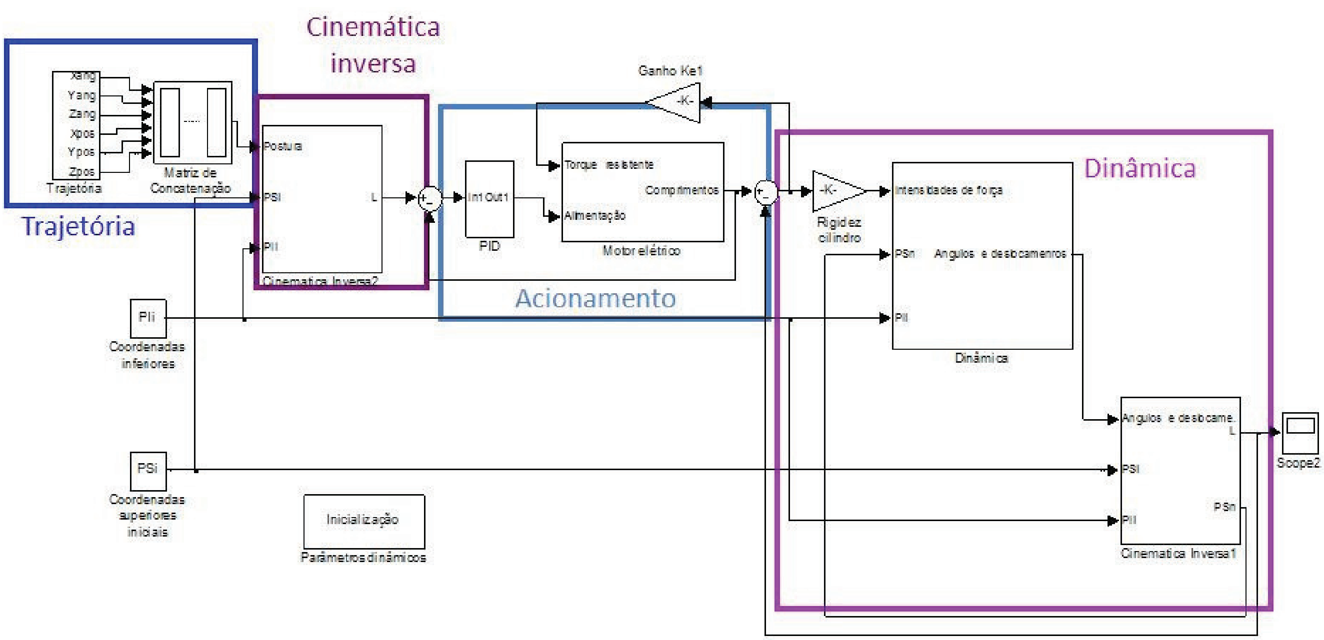


Figura 4.11: Modelo Simulink da plataforma de Stewart-Gough

Capítulo 5

Simulação Dinâmica e Validação

5.1 Teste n°1: movimento de translação em Z

A estrutura desenvolvida no Capítulo 4 foi testada sob diversas configurações. Apresenta-se abaixo, uma configuração de simples interpretação; um movimento de translação vertical do centro de gravidade de uma altura de 0,543 m. Este movimento induz um posicionamento para os seis braços, ilustrado na figura 5.1. A figura 5.2 e figura 5.4 ilustram, respectivamente, as variações de força necessárias para a realização do movimento e os comprimentos obtidos após a translação, levando em consideração a dinâmica da plataforma.

Para um movimento de translação em *Z*, todos os atuadores intervêm, constatando-se que o acionamento é dado de forma satisfatória, com um erro quase nulo e um tempo de máximo pico de cerca de 5 s. A estabilização do movimento é dada aos 40 s aproximadamente.

As posições fornecidas pelos ciclos de acionamento são exatamente as mesmas impostas pelos valores de referência, o que é natural dada à presença do controlador *PID*.

As variações das intensidades de força tornam-se nulas após a mudança dos valores de referência, sendo que o inicio da estabilização se dá aos 25 s.

Em contrapartida, existe nos comprimentos resultantes da dinâmica da plataforma, um erro sutil, provavelmente devido a complexidade e a não-unicidade da reconstituição da posição do centro de gravidade (o que é retratado pelo modelo cinemático direto equivalente, mostrado no Anexo A). Além disso, mesmo que as referências desejadas sejam simétricas duas a duas, a resposta obtida mostra uma leve dissimetria.

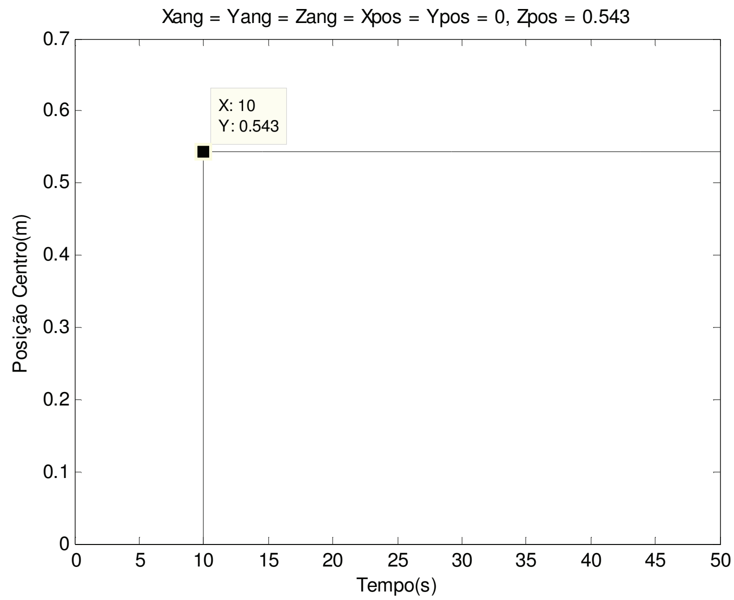


Figura 5.1: Trajetória de referência – Deslocamento em Z

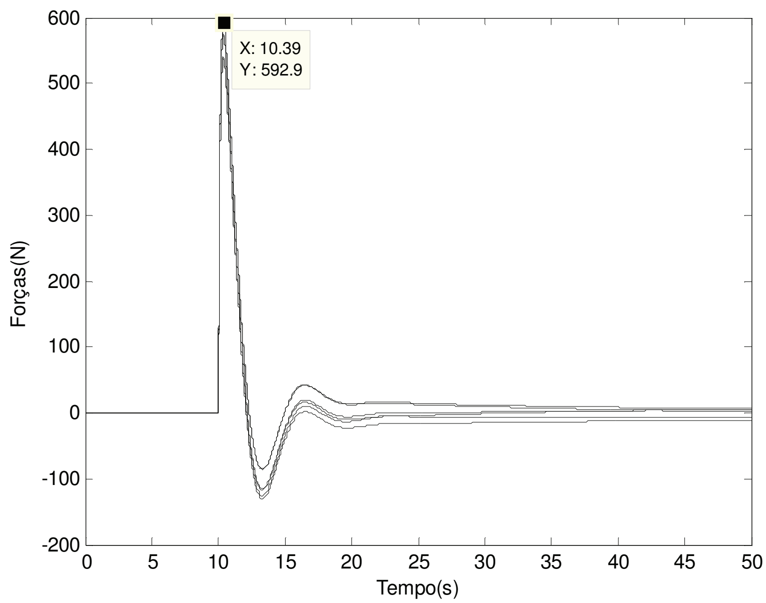


Figura 5.2: Esforço de controle produzido nos cilindros – Deslocamento em Z

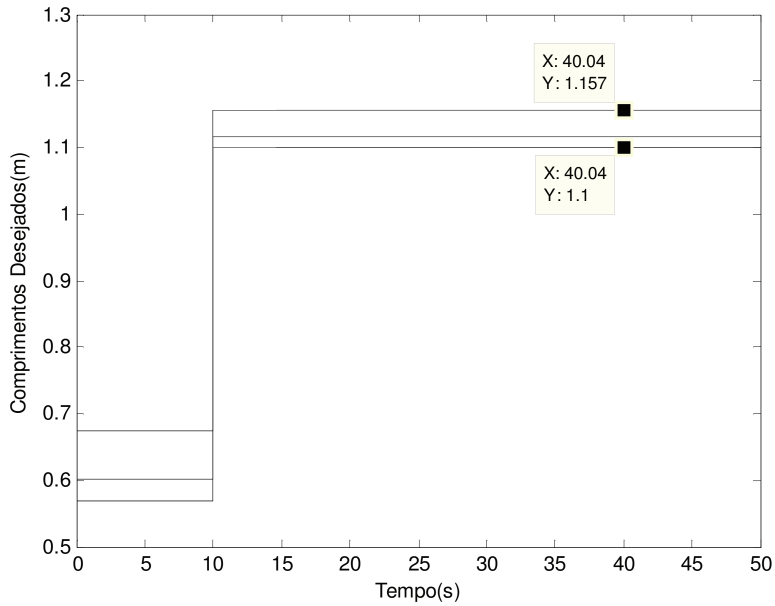


Figura 5.3: Trajetória desejada dos cilindros - Deslocamento em Z

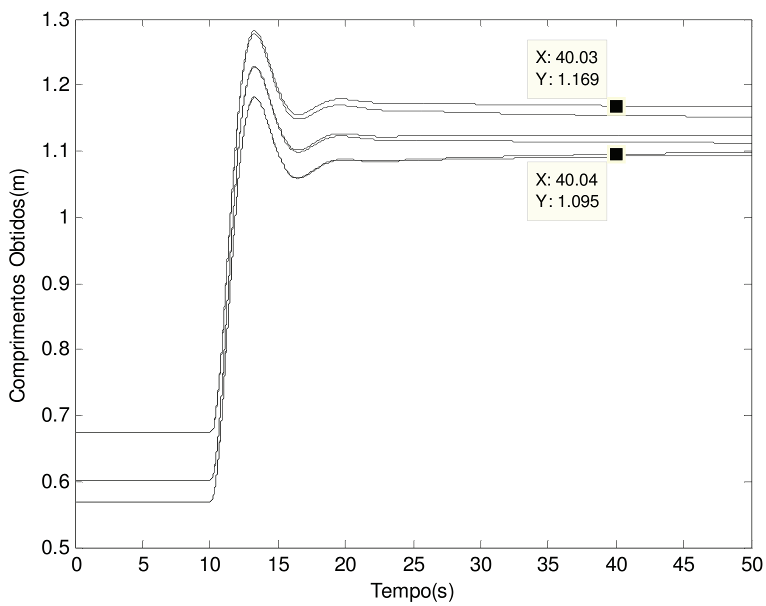


Figura 5.4: Trajetória realizada pelos cilindros - Deslocamento em Z

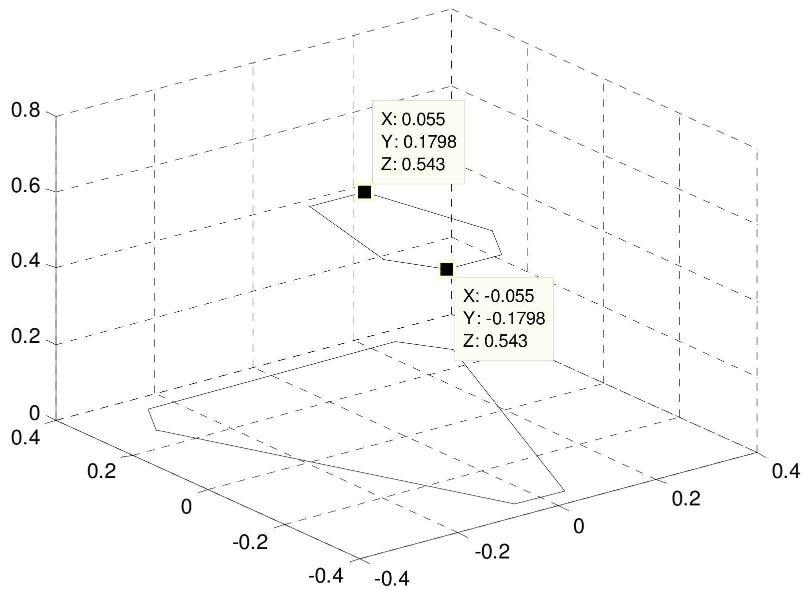


Figura 5.5: Representação tridimensional do movimento - Configuração inicial

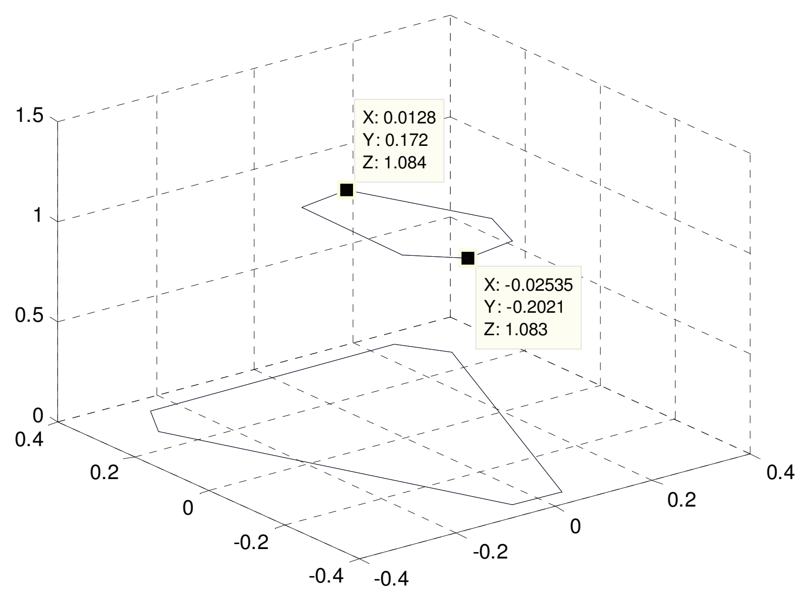


Figura 5.6: Representação tridimensional do movimento - Deslocamento em Z

A representação tridimensional vista na figura 5.6 mostra que dentro dos 50 s simulados a plataforma quase atinge a altura exata desejada, que deveria ser de 1,086m. Observa-se também uma leve inclinação da parte superior em relação às suas extremidades diametralmente opostas.

5.2 Teste n°2 : movimento de rotação em Z

O teste visa observar o comportamento dos cilindros na situação de rotação da plataforma em torno de seu eixo vertical.

Pelas figuras resultantes da simulação observa-se que o sistema precisou de um tempo maior para estabilizar-se. Os comprimentos desejados nos cilindros, aos 45s de simulação, apresentam erro de precisão de 4,5%.

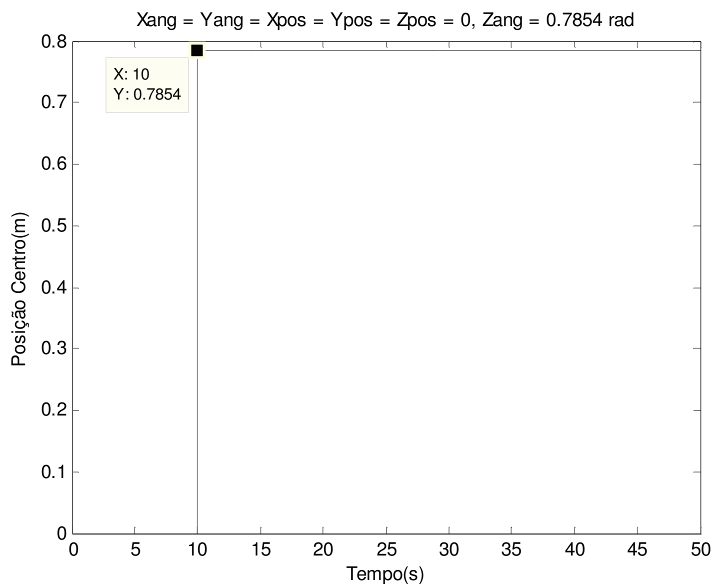


Figura 5.7: Trajetória de referência - Rotação em Z

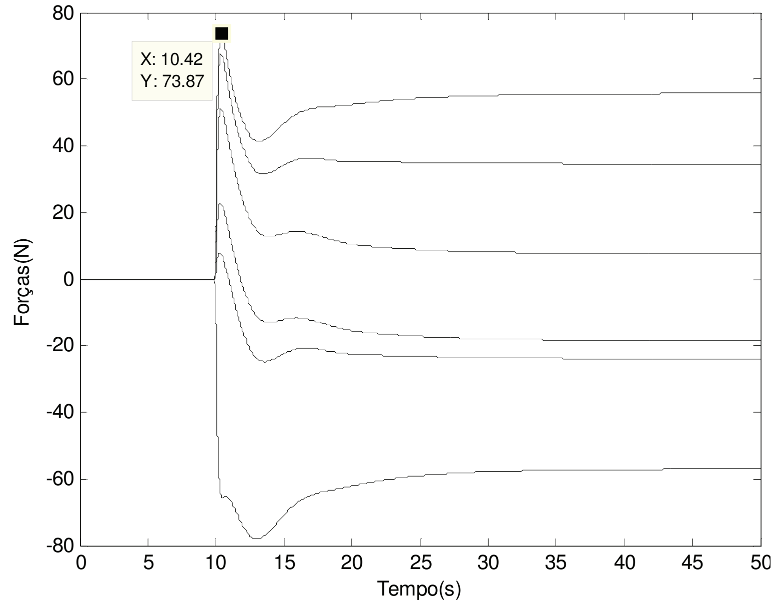


Figura 5.8: Esforço de controle produzido nos cilindros - Rotação em Z

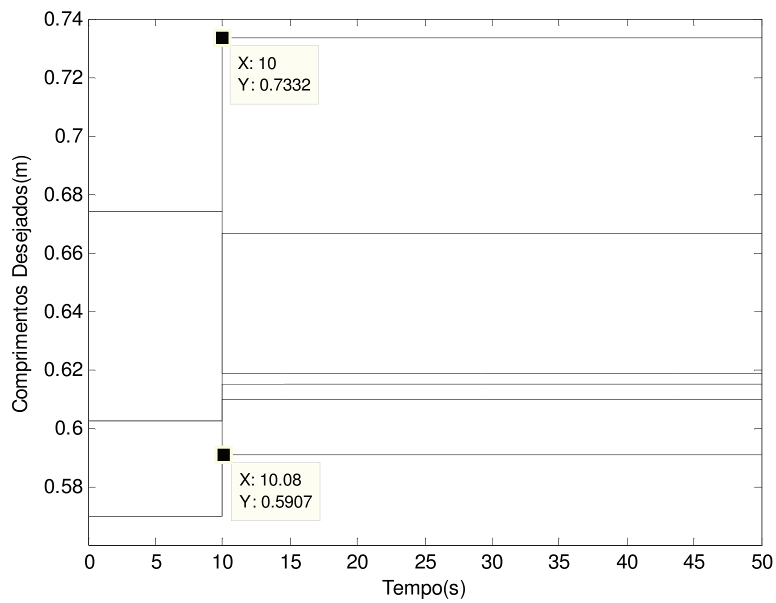


Figura 5.9: Trajetória desejada dos cilindros - Rotação em Z

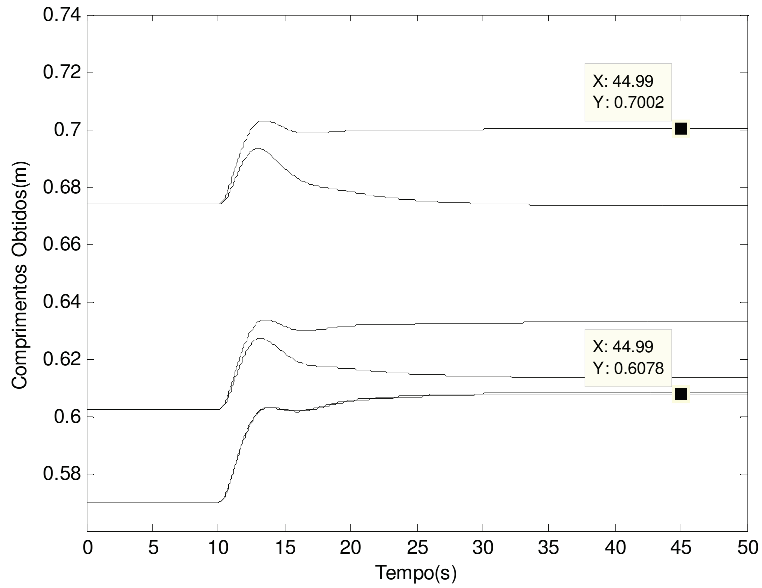


Figura 5.10: Trajetória realizada pelos cilindros - Rotação em Z

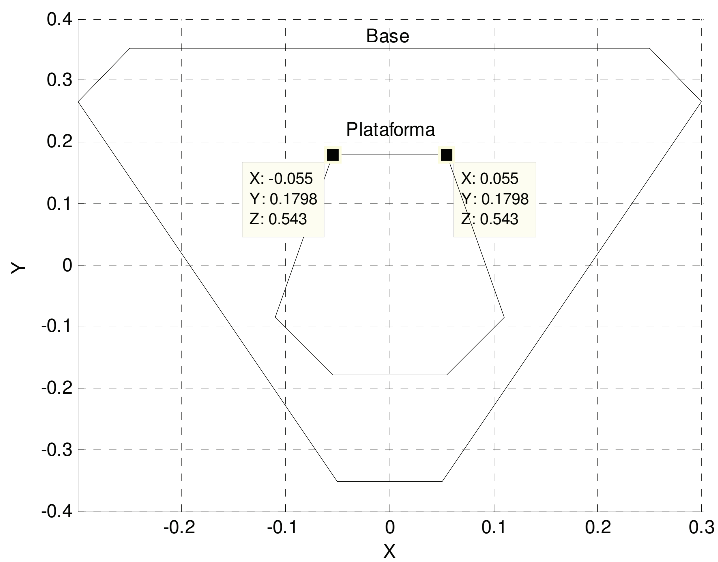


Figura 5.11: Visão transversal do robô em sua configuração original

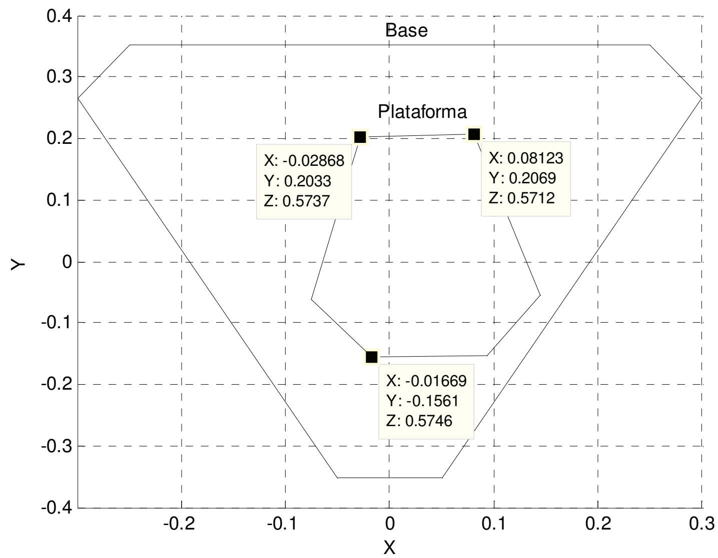


Figura 5.12: Visão transversal do robô após a rotação em Z

Para esta simulação, o desempenho do controlador PID não foi satisfatório. A figura 5.12 mostra que além da rotação em *Z*, houve também uma leve translação da plataforma em Z, a qual deveria ser nula. Embora o tempo de simulação tenha sido de 50 s, a estabilidade é atingida aos 30 s. Porém, o deslocamento angular atingido (0,0327 rad ou 1,8760°) não é o esperado (0,7854 rad ou 45°).

5.3 Teste n°3 : movimento senoidal em Z

Neste ensaio observa-se que a simetria do problema é conservada, sendo que os atuadores têm dois a dois o mesmo movimento.

Os valores calculados não são, contudo, exatamente os mesmos das entradas desejadas, oscilando em uma vizinhança muito próxima do valor de referência, sendo que o erro máximo apresentado para as amplitudes de oscilação desejadas é de 2%.

Mais uma vez esta diferença pode ser originária da não-unicidade da reconstituição da posição do centro da plataforma a partir das forças aplicadas.

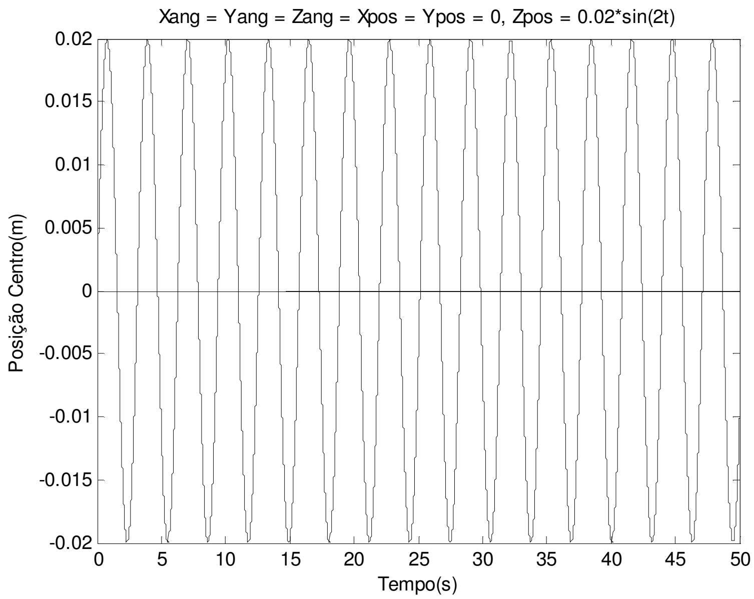


Figura 5.13: Trajetória de referência - Senoide em Z

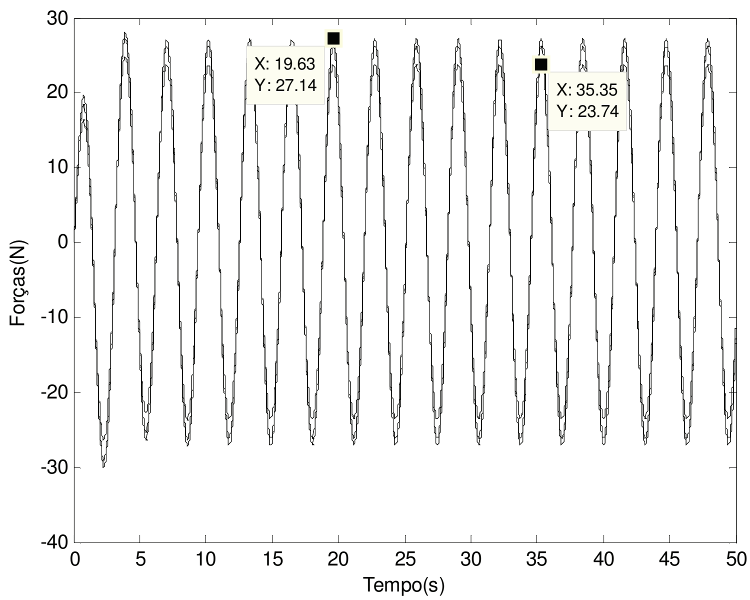


Figura 5.14: Esforço de controle produzido nos cilindros - Senoide em Z

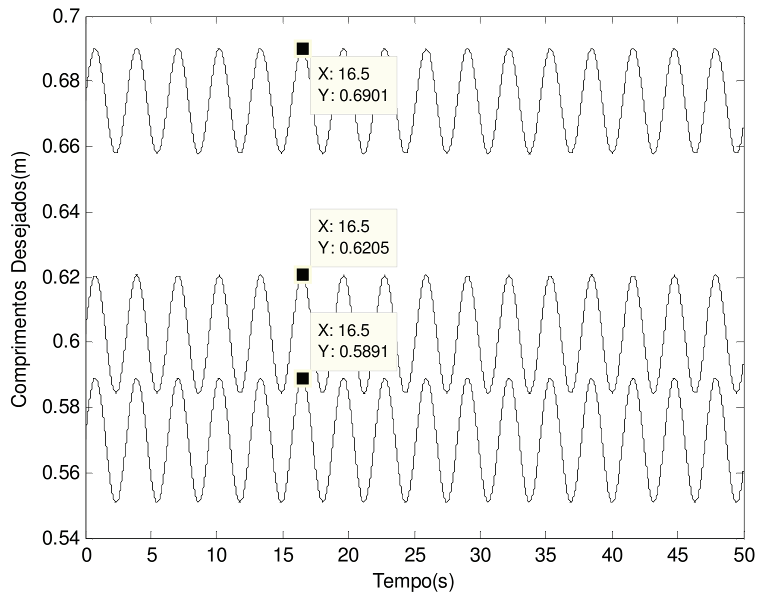


Figura 5.15: Trajetória desejada dos cilindros - Senoide em Z

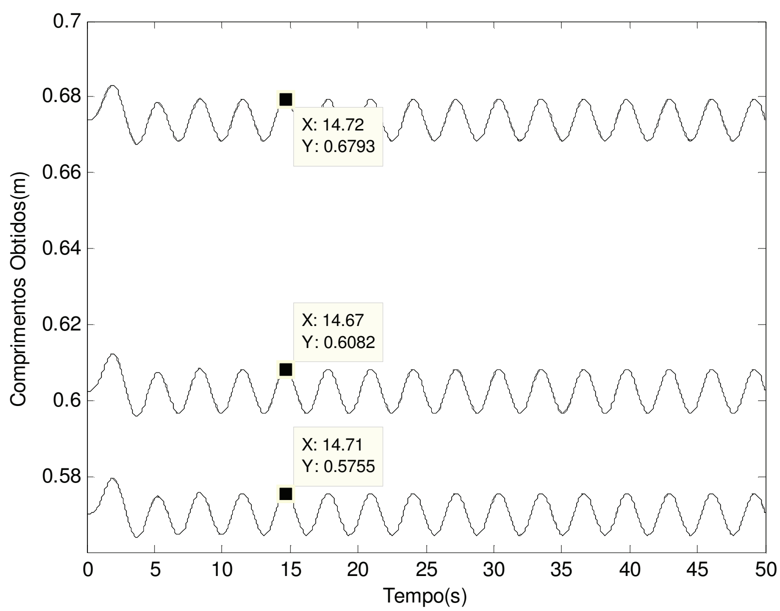


Figura 5.16: Trajetória realizada pelos cilindros - Senoide em Z

5.4 Teste n°4: movimento senoidal nas seis coordenadas

A análise feita neste ensaio é muito similar àquela feita no ensaio precedente. A amplitude de oscilação desejada apresentou um erro um pouco maior, da ordem de 4%. Percebe-se ainda que ha uma defasagem no tempo de atualização do movimento, já que o tempo necessário para os deslocamentos atingirem os picos de oscilação é um pouco maior que o esperado. A diferença espectral dos deslocamentos desejados para os realizados deve-se ao atraso no tempo de resposta da simulação em tempo real, resultado da complexa arquitetura do robô paralelo.

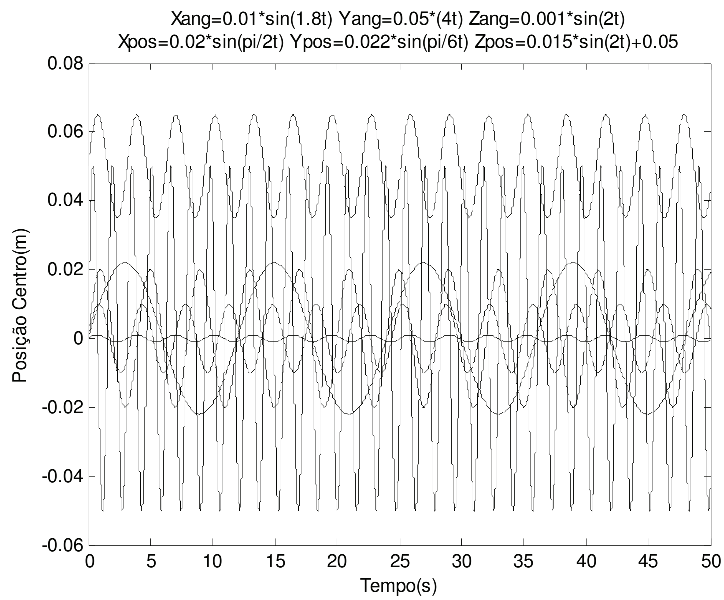


Figura 5.17: Trajetória de referência – 6 Senoides

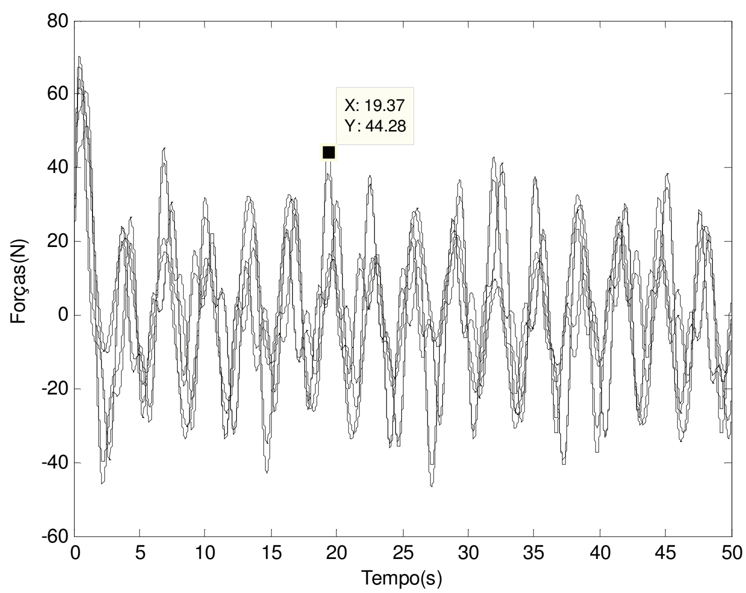


Figura 5.18: Esforço de controle produzido nos cilindros – 6 Senoides

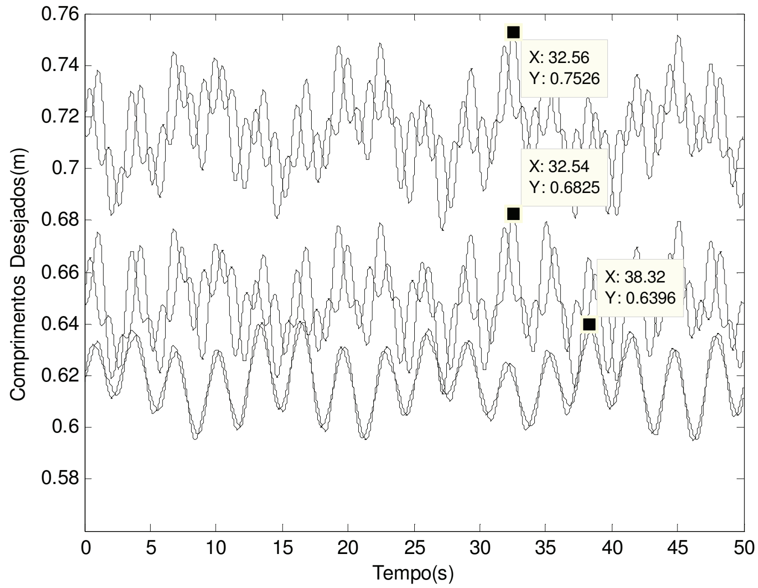


Figura 5.19: Trajetória desejada dos cilindros – 6 Senoides

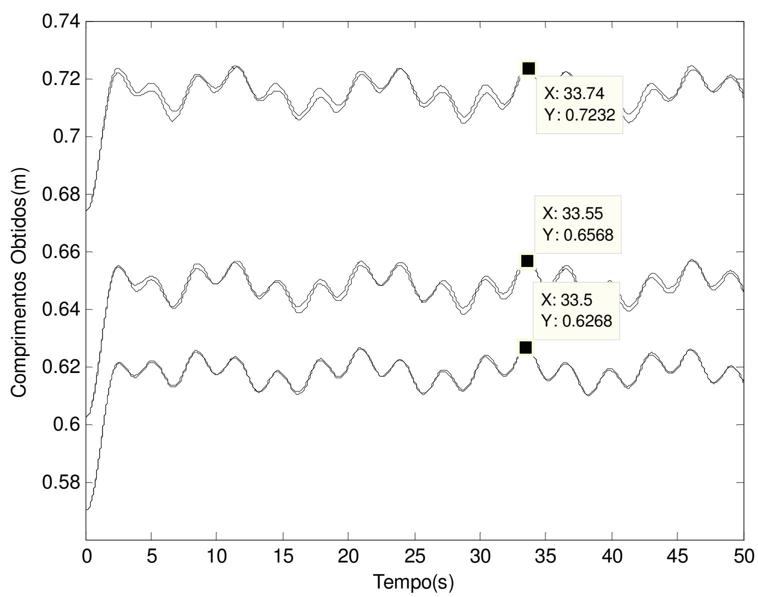


Figura 5.20: Trajetória realizada pelos cilindros - 6 Senoides

Capítulo 6

Conclusões e Perspectivas Futuras

Modelos dinâmicos que descrevem as equações de corpo rígido e o acionamento da Plataforma de Stewart foram estudados neste trabalho sob dois aspectos diferentes, com a finalidade de simular seu movimento segundo a implementação de um sistema de controle.

O primeiro estudo, de completa estruturação por abordar a cinemática e dinâmica tanto dos braços acionadores quanto da plataforma, resultou em um modelo implementável pouco prático do ponto de vista da simulação devido ao custo computacional envolvido na solução das equações de forma fechada desenvolvidas neste trabalho.

O segundo estudo consistiu na elaboração de um simulador para a plataforma de Stewart, que teve sua arquitetura desenvolvida em módulos. Estes incluem aspéctos relacionados à cinemática inversa, aos acionamentos dos braços e à dinâmica da plataforma fazendo interferir uma rigidez equivalente dos cilindros de acionamento. O simulador da plataforma de Stewart foi elaborado empregando motores de corrente contínua e controles de posição à eles associados (do tipo PID).

Foi possível portanto, considerando como referência o movimento do centro de gravidade da plataforma, analisar os comprimentos calculados para os eixos, os quais são necessários para atingir esta posição, e em seguida examinar as posições dos eixos efetivamente atingidas considerando-se a dinâmica do sistema.

A convergência dos comprimentos obtidos pela simulação versus àqueles desejados não é perfeita. Isso certamente é devido a não-unicidade ligada à cinemática direta, mesmo que o controlador PID resulte saídas aceitáveis para as referências consideradas nos testes.

Para o funcionamento do simulador resta ainda ajustar os valores numéricos para os valores reais do protótipo da plataforma situada na Unicamp, uma vez que a massa da plataforma e os dados característicos dos motores usados nesta simulação não são os mesmos medidos na figura 2.11. Poderá também ser feita a substituição dos motores de corrente contínua, considerados no modelo deste trabalho, pelos motores hidráulicos do protótipo real.

Finalmente, as perspectivas deste trabalho consideram o estudo de leis de comando na plataforma, conectando módulos não somente do controlador PID, mas também das estruturas de comandos avançados, eventualmente multivariáveis, que considerem os ciclos fechados formados pelos braços do manipulador paralelo.

Como um último passo, poderá ser feita a validação da lei de controle na plataforma real.

Referências Bibliográficas

Arai, T., & Sheridan, T. (1988). Characteristics and Mechanism Analysis of Parallel Link Manipulator. *Proc. 2nd IFAC Symp. SYROCO*, (pp. 119-124).

Berger, D. R., Schulte-Pelkum, J., & Bulthoff, H. H. (2010). Simulating Believable Forward Accelerations. *ACM Transactions on Applied Perception , 7* (1), 27 pages.

Butler, G. H., & Amerongen, J. v. (1989). Model reference adaptative control of a direct-drive dc motor. *IEEE Control Systems Magazine* .

Chang, C.-F., Fu, L.-C., Yu, M.-Y., & Huang, C.-I. (2004). Sliding-mode tracking control of the Stewart platform. (pp. 561-568). Taipei, Taiwan: 5th Asian Control Conference.

Dasgupta, B., & Mruthyunjaya, T. (1998). Closed-form Dynamic Equations of the General Stewart Platform Through the Newton-Euler Approach. *33* (7), 993-1012.

Davliakos, I., & Papadopoulos, E. (2009). Impedance Model-Based Control for an Electrohydraulic Stewart. *European Journal of Control , 5*, 1-18.

Dumur, D., Rosário, J., & Machado, J. (2007). *Control of a 6 dof parallel manipulator through a mechatronic approach.* Unicamp, Laboratory of Automation and Robotics, DPM-FEM. Campinas: Journal of Vibration and Control, Vol. 13, No. 9-10.

Homma, K., Hashino, S., & Arai, T. (1998). An Upper Limb Motion Assist System: Experiments with Arm Models. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 758-763).

Huang, X., Liao, Q., & Wei, S. (2010). Closed-form forward kinematics for a symmetrical 6-6 Stewart platform using algebraic elimination. *Mechanism and Machine Theory , 45*, 327–334.

Ibrahim, O., & Khalil, W. (2010). Inverse and direct dynamic models of hybrid robots. *Mechanism and Machine Theory* (45), 627–640.

Jakobovic, D., & Jalenkovic, L. (2002). *The forward and inverse kinematics problems for Stewart parallel mechanisms.* Zagreb: University of Zagreb, FEE&C, Unska 3, 10000.

Khalil, W., & Guegan, S. (2004). Inverse and Direct Dynamic Modeling of Gough-Stewart Robots. *20* (4).

Khalil, W., & Kleinfinger, J. (1986). A New Geometric Notation for Open and Closed-Loop Robots. *IEEE Conference on Robotics and Automation*, (pp. 1174-1180). San Francisco.

Koevermans, W. (1975). Design and performance of the four d.o.f. motion system of the NLR research flight simulator. *In AGARD Conf. Proc. No 198* (pp. 17-1/17-11). La Haye: Flight Simulation.

Lallemand, J., Goudali, A., & Zeghloul, S. (1997). The 6-Dof 2-Delta Parallel Robot. *Robotica , 15*, pp. 407-416.

Lee, S.-H., Song, J.-B., Choi, W.-C., & Ho, D. (2003). Position Control of a Stewart Platform Using Inverse Dynamics Control with Approximate Dynamics. *Mechatronics* (13), pp. 605-619.

Lee, T.-Y., & Shim, J.-K. (2001, February). Forward Kinematics of the General 6-6 Stewart Platform Using Algebric Elimination. 1073-1085.

Liu, M.-J., Li, C.-X., & Li, C.-N. (2000, February). Dynamics Analysis of the Gough-Stewart Platform Manipulator. *16* (1).

Ma, O., & Angeles, J. (1991). Architecture Singularities of Platform Manipulators. *IEEE International Conference on Robotics and Automation.* Sacramento.

Mahmoodi, A., Menhaj, M., & Sabzehparvar, M. (2009). An efficient method for solution of inverse dynamics of Stewart platform. *Aircraft Engineering and Aerospace Technology: An International Journal , 81* (5), 398–406.

Mazoni, A. F. (2003). *Projeto e implementação de programa computacional para modelagem, simulação e controle de movimentos de uma plataforma de posicionamento.* Technical, Unicamp, DPM-FEM, Campinas.

Meirovitch, L. (1970). Methods of Analytical Dynamics. (M. Hill, Éd.)

Merlet, J.-P. (2006). *Parallel Robots (Solid Mechanics and Its Applications).* Springer.

Nabat, V., Company, O., & Pierrot, F. (2006). Dynamic modeling and identification of Part 4, a very high speed parallel manipulator., (pp. 496-501). Beijing.

Nguyen, C., & Pooran, F. (1989). Dynamic analysis of a 6 d.o.f. CKCM robot end-effector for dual-arm telerobot systems. Dans *Robotics and Autonomous Systems* (pp. 377-394).

Pashkevich, A., Chablat, D., & Wenger, P. (2006). Kinematics and workspace analysis of a three-axis parallel manipulator: the Orthoglide. *Robotica , 24* (01), 39-49.

Pierrot, F., Uchiyama, M., & Dauchez, P. (1990). A new design of a 6-DOF parallel robot. *Robotics and Mechatronics , 2*, 308-315.

Rosario, J., Lara, F., Moretti, M., & Uribe, A. (march 2010). Supervision and Control Strategies of a 6 DOF Parallel Manipulator using Mechatronic Approach. Dans A. Lazinica (Éd.), *Robot Manipulators* (pp. 42-49). Sciyo.

Rosario, J., Moretti, M., & Dumur, D. (may 2007). Control of a 6 DOF Parallel Manipulator using Rapid Prototyping. Dans S. Pennachio (Éd.), *Emerging Technologies, Robotics and Control Systems* (Vol. 2, pp. 42-49). International Society for Advanced Research.

Schwartz, J.-M. (2003). *Calcul rapide de forces et de déformations mécaniques non-linéaires et visco-élastiques pour la simulation de chirurgie.* Technical, Université Laval, Département de génie électrique et de génie informatique, Québec.

Stewart, D. (1965). A platform with 6 degrees of freedom. *In Proc. Institution of Mechanical Engineers*, *v.180*, pp. 371-386.

Terrier, M., Dugas, A., & Hascoët, J.-Y. (2004). Qualification of Parallel Kinematics Machines in High-Speed Milling on Free Form Surfaces. *International Journal of Machine Tools & Manufacture , 44*, 865-877.

Wang, Y.-X., Li, Y.-T., & Guo, R.-Q. (2007). Study on the method for 6-SPS Gough–Stewart platforms to pass through type-II singular points with its original configuration. *Mechanical Engineering Science , 222*, 723-736.

Wendlandt, J. M., & Sastry, S. S. (1994). Design and Control of a Simplified Stewart Platform for Endoscopy. *Proceedings of the Conference on Decision and Control*, *1*, pp. 357-362.

Yang, C., He, J., Han, J., & Liu, X. (2009). Real-time state estimation for spatial six-degree-of-freedom. *Mechatronics , 19*, 1026–1033.

Yang, C., Huang, Q., Jiang, H., Peter, O. O., & Han, J. (2010). PD Control With Gravity Compensation For Hydraulic 6-DOF Parallel Manipulator. *Mechanism and Machine Theory , 45* (4), pp. 666-677.

ANEXO A: Modelo Geométrico Direto

O objetivo do modelo geométrico direto é de reconstituir a posição e orientação da parte móvel a partir do conhecimento dos comprimentos dos braços atuadores. Tal problema não possui uma solução analítica única definida para a maioria dos manipuladores 6-6 que tem a forma de hexapode – com seis juntas na base e seis juntas na parte móvel.

Pode-se, contudo, reconstituir esta solução através da otimização, eliminando-se a necessidade de utilizar o método de Newton para a resolução do problema cinemático direto já que a matriz Jacobiana é singular para a estrutura de hexapode, ou seja, é não-inversível por ter o determinante nulo.

Em contrapartida, as relações envolvidas na cinemática direta de um modelo robótico tipo hexapodo, podem ser formuladas matematicamente de várias formas e em seguida tratadas numericamente. Neste tópico é definida uma função matemática a ser otimizada e esta representação será analisada graficamente, uma vez considerado um algoritmo de otimização (Jakobovic & Jalenkovic, 2002)

A posição e a orientação atuais da plataforma móvel são definidas pelo intermédio de três coordenadas representando a posição do centro da parte superior móvel e de três ângulos que definem sua orientação.

Sejam as coordenadas do centro móvel:

$$\vec{\mathbf{t}} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{A-1}$$

Os valores dos ângulos (θ, ρ e ψ) representam as rotações sucessivas em torno dos eixos *X*, *Y* e *Z* respectivamente.

A geometria do hexapodo é definida por seis vetores para a base e seis vetores para a parte móvel. Tais vetores são representados por um sistema local de coordenadas de cada parte, figura 4.1 e figura 4.2, e possuem valores constantes.

Para uma posição e orientação dadas, o erro pode ser expresso como a soma dos quadrados das diferenças entre os valores dos comprimentos calculados e os comprimentos atuais (Jakobovic & Jalenkovic, 2002). Deduz-se dessa forma a função a ser otimizada, incluindo as variáveis procuradas:

(A-2)

$$F_{opt} = \sum_{i=1}^{6} [(N(\Re \mathbf{PS}_i + \mathbf{t} - \mathbf{PI}_i))^2 - (l + \Delta l)^2]^2 ,$$

onde N é a distância Euclidiana entre os pares de vetores, e W_a é o vetor das variáveis de otimização a serem determinadas:

(A-3)

$$\mathbf{W_a} = \begin{bmatrix} t_x \, t_y \, t_z \\ \theta \, \rho \, \psi \end{bmatrix}$$

O problema de otimização foi solucionado usando a função do Matlab *fminunc*, a qual encontra o mínimo de uma função multivariável escalar com uma condição inicial estimada. Trata-se de uma otimização não-linear sem restrição.

Simulação do ciclo cinemático inverso e direto

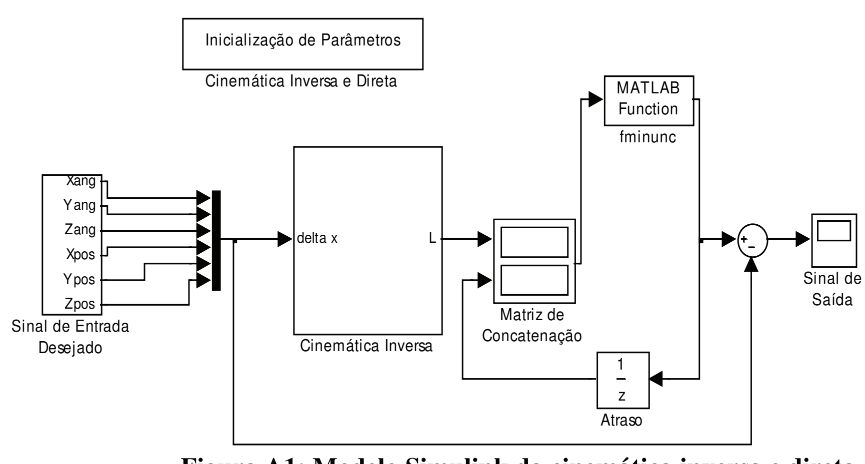


Figura A1: Modelo Simulink da cinemática inversa e direta

A figura A2 abaixo ilustra os resultados obtidos no encadeamento cinemática inversa – cinemática direta, que parte de uma posição horizontal para voltar à esta mesma posição. A função de otimização atinge globalmente o objetivo de minimizar a diferença entre os valores de referência e os valores obtidos. Os resultados são fortemente dependentes da escolha dos parâmetros da função utilizada para a otimização.

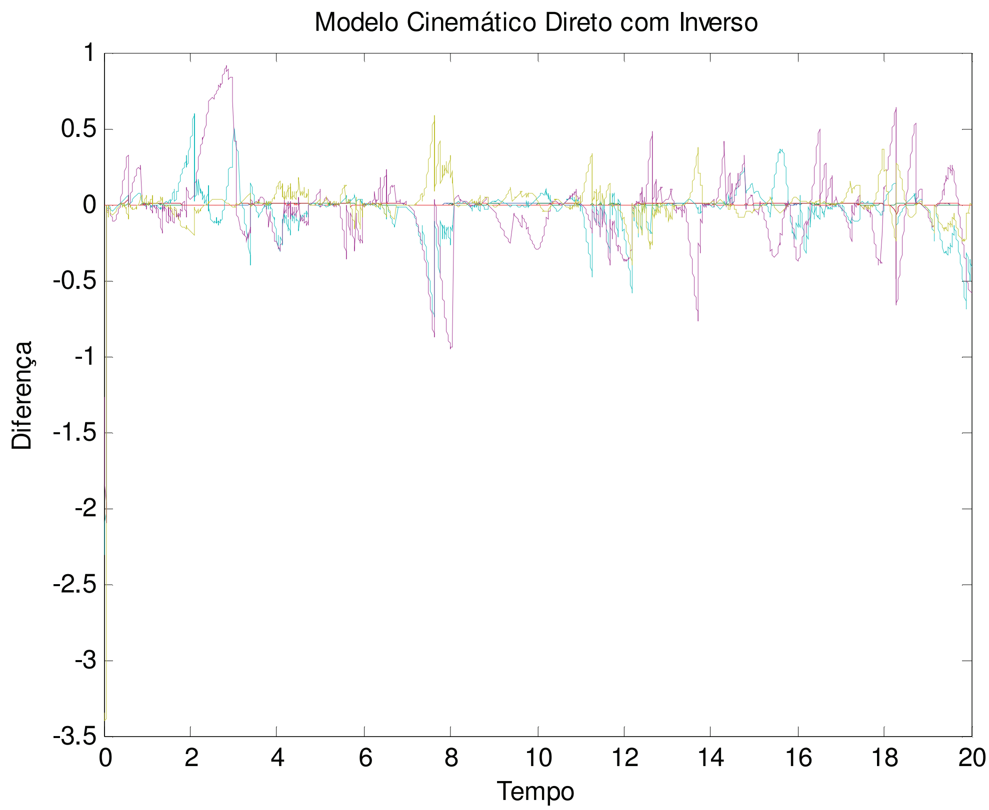


Figura A2: Cinemática inversa com cinemática direta

A cinemática direta pode ser aplicada principalmente para verificar e validar a cinemática inversa, fazendo o cálculo do erro das instruções fornecidas por esta última (Jakobovic & Jalenkovic, 2002).

ANEXO B: Programação em Matlab™ – Capítulo 3

Equações do modelo dinâmico da plataforma de Stewart:

```
function dw = modelo_controle(tempo,w)
global M R0 Ip mu md Cu Cp Cs g R0 rd0 ru0 Iu0 Id0 k b p F L0 L
theta = w(1:3);
t = w(4:6);
om = w(7:9);
dt = w(10:12);
RR = matriz_rotacao(theta);
I = RR*Ip*RR.';
R = RR*R0;
r = norm(R);
Rt = matriz_vetorial(R);
q = RR*p;
La = L;
[s,L] = vetores_cilindros(RR,t);
eta_soma = zeros(6,1);
J_soma = zeros(6,6);
H = [];
dLv = [];
for i = 1:6
xh = s(:,i);
aux = cross(k(:,i),s(:,i));
yh = aux/norm(aux);
if (isnan(yh))|(norm(aux)<1e-10)
yh = xh-k(:,i);
yh = yh/norm(yh);
```

```
end
zh = cross(xh,yh);
T = [xh yh zh];
rd = T*rd0;
nrd = norm(rd);
v = [L(i); 0; 0];
ru = T*(v+ru0);
nru = norm(ru);
Id = T*Id0*T.';
Iu = T*(Iu0+mu*L(i)*L(i)*[0 0 0; 0 1 0; 0 0 1])*T.';
st = matriz_vetorial(s);
dS = cross(om,q(:,i))+dt;
dL = s(:,i).'*dS;
dLv(i,1) = dL;
A1 = mu*(1+2*s(:,i).'*ru/L(i))-(md*nrd*nrd+mu*nru*nru)/(L(i)*L(i));
A2 = (md*nrd*nrd+mu*nru*nru)/(L(i)*L(i));
B1 = -(mu/L(i))*(s(:,i)*ru.'+ru*s(:,i).');
srd = cross(s(:,i),rd);
sru = cross(s(:,i),ru);
B2 = -(1/(L(i)*L(i)))*(md*srd*srd.'+mu*sru*sru.'-st*(Id+Iu)*st);
Q(:,:,i) = A1*s(:,i)*s(:,i).'+A2*eye(3)+B1+B2;
qt = matriz_vetorial(q(:,i));
J(:,:,i) = -[Q(:,:,i)     -Q(:,:,i)*qt;
qt*Q(:,:,i)  -qt*Q(:,:,i)*qt];
W = cross(s(:,i),dS/L(i));
f(:,i) = Cs*(W-om);
U1 = cross(om,cross(om,q(:,i)));
u = s(:,i).'*U1+1/L(i)*(dS-dL*s(:,i)).'*(dS-dL*s(:,i));
U2 = 1/L(i)*(cross(s(:,i),U1)-2*dL*W);
U3 = cross(U2,rd) + cross(W,cross(W,rd));
U4 = u*s(:,i) + cross(U2,ru)+cross(W,cross(W,ru))+2*dL*cross(W,s(:,i));
```

```
U5 = md*cross(rd,U3)+mu*cross(ru,U4)+(Id+Iu)*U2+cross(W,(Id+Iu)*W)...
-cross((md*rd+mu*ru),g)+Cu*W+f(:,i);
V(:,i) = (mu*s(:,i).'*U4+Cp*dL-mu*s(:,i).'*g)*s(:,i)...
-(1/L(i))*cross(s(:,i),U5);
eta(:,i) = -[V(:,i);
cross(q(:,i),V(:,i))-f(:,i)];
eta_soma = eta_soma + eta(:,i);
J_soma = J_soma + J(:,:,i);
H = [H [-s(:,i); -cross(q(:,i),s(:,i))]];
end
Kp = 5000;
Kv = 500;
F = Kp*(L0.'-L.')-Kv*dLv;
Jplat = [M*eye(3) -M*Rt; M*Rt I+M*(r^2*eye(3)-R*R.')];
etaplat = [M*(cross(om,cross(om,R))-g);
cross(om,I*om) + M*cross(R,((om.'*R)*om-g))];
eta_soma = eta_soma + etaplat;
J_soma = J_soma + Jplat;
dd_t_alpha = (J_soma)\(H*F - eta_soma);
dw(1:3,1) = w(7:9);
dw(4:6,1) = w(10:12);
dw(7:9,1) = dd_t_alpha(1:3);
dw(10:12,1) = dd_t_alpha(4:6);
```

Simulação do modelo:

```
clear all
parametros
global F
%F = 10*ones(6,1);
theta = [0 0 -1.0].';
```

```
t = [0 0 .8].';
om = [.0 .0 0.1].';
dt = [0 0 0.0].';
y0 = [theta;t;om;dt];
dw = modelo_controle(0,y0);
[t,y] = ode45('modelo_controle',[0 1],y0);
subplot(2,1,1)
plot(t,y(:,1:3))
xlabel('Temps')
ylabel('Position angulaire')
subplot(2,1,2)
plot(t,y(:,4:6))
xlabel('Temps')
ylabel('Déplacement')
modelo_controle_simulink([y0; 0])
```

Obtêm os vetores diretores e comprimentos dos cilindros.

```
% [s,L] = vetores_cilindros(R,t)
% R - matriz de rotação
% t - translação
% s - vetores diretores como colunas
% L - comprimentos
function [s,L] = vetores_cilindros(R,t)
global p b
S = R*p + t*ones(1,6) - b;
L = sqrt(sum(S.^2));
s = S*diag(1./L);
```

ANEXO C: Programação em Matlab™ – Capítulo 4

Parâmetros de inicialização do sistema.

```
% Valores geométricos da plataforma:
const = 10^-3;
a=300*const;
b=300*const;
d=80*const;
e=95*const;
alpha=874*const;
beta=500*const;
delta=40*const;
epsilon=60*const;
h=543*const;
r = 639*const;
esp = 0.02;
t = pi*30/180;
s = pi*60/180;

% Matrizes Geométricas
p1i=[alpha/2-epsilon, (alpha+beta)/2*cos(t)-delta, 0];
p2i=[-alpha/2+epsilon (alpha+beta)/2*cos(t)-delta 0];
p3i=[-1/2*alpha+epsilon+(2*epsilon-beta)*cos(s), (1/2*alpha+1/2*beta)*cos(t)-delta-(beta-
2*epsilon)*cos(t), 0];
p4i=[-beta/2+epsilon -(alpha+beta)/2*cos(t)+delta 0];
p5i=[beta/2-epsilon -(alpha+beta)/2*cos(t)+delta 0];
p6i=[1/2*alpha-epsilon-(2*epsilon-beta)*cos(s), (1/2*alpha+1/2*beta)*cos(t)-delta-(beta-
2*epsilon)*cos(t), 0];
```

```
p1s=[b/2-e (a+b)/2*cos(t)-d h];
p2s=[-b/2+e (a+b)/2*cos(t)-d h];
p3s=[-a/2+e+(2*e-b)*cos(s) -(a+b)/2*cos(t)+d+(b-2*e)*cos(t) h];
p4s=[-a/2+e -(a+b)/2*cos(t)+d h];
p5s=[a/2-e -(a+b)/2*cos(t)+d h];
p6s=[a/2-e-(2*e-b)*cos(s) -(a+b)/2*cos(t)+d+(b-2*e)*cos(t) h];
global PIi PSi
global L0 L
PIi=[p1i; p2i; p3i; p4i; p5i; p6i];
PSi=[p1s; p2s; p3s; p4s; p5s; p6s];
L0 = sqrt(sum(((PSi-PIi)').^2))

% Peso
m = 5000;
g = 9.81;

% Cálculo do tensor de inércia
soma_num = 0;
soma_den = 0;
for n = 1:5;
    soma_num = soma_num + (norm(cross(PSi((n+1),:),PSi(n,:)))) *
    (PSi((n+1),:)*PSi((n+1),:)' + dot(PSi((n+1),:),PSi(n,:)) + PSi((n),:)*PSi((n),:)');
    soma_den = soma_den + norm(cross(PSi((n+1),:),PSi(n,:)));
end
soma_num = soma_num + (norm(cross(PSi((1),:),PSi(6,:)))) * (PSi((1),:)*PSi((1),:)' +
dot(PSi((1),:),PSi(6,:)) + PSi((6),:)*PSi((6),:)');
soma_den = soma_den + norm(cross(PSi((1),:),PSi(6,:)));

% Geometria: hexagono fino irregular
Iz = m/6*soma_num/soma_den;
Ix = 1/12*m*(3*r^2+esp^2);
```

```
% Geometria: cilindro pleno
Iy = Ix;
tensor_diag = [Ix 0 0; 0 Iy 0; 0 0 Iz]
I = tensor_diag;
Iinv = inv(I);

% Parâmetros dos motores
L = 0.0065;
R = 4.2;
Kt = 1.43;
Kb = 0.8594;
p = 0.01;
J = 0.0083;
f = 0.0035;

% Parâmetros dos atuadores
S = pi*20^2*10^-6;
E = 250000;
ha = 0.3;

% Parâmetros do controlador
Ti=1/2;
k=5600;
Td=1/30;
```

Função que calcula o somatório dos momentos.

```
function C = cross_m_sum(A);
% Produto vetorial de todos os vetores sob a forma de matrizes e somatórios
% de vetores.
```

```
% C = cross_m(A,B);
% A - matriz de 2*n linhas. As primeiras n são os vetores
% que são os primeiros fatores do produto. As linhas seguintes
% são os segundos fatores.
% C – vetor do somatório dos produtos vetoriais.
C = sum(cross(A(1:end/2,:),A(end/2+1:end,:)));
```

Função de otimização – Anexo A.

```
function w = cin_dir3(u);
global PSi PIi
global L
L = u(1,:);
wa = u(2,:);
options = optimset('MaxIter',5e4,'MaxFunEvals',1e5,'TolFun',1e-8,'TolX',1e-8);
w = fminunc('f_obj',wa,options);
```

Cálculo da função a ser minimizada – Anexo A.

```
% Função objetivo :
function z = f_obj(x);
global PSi PIi
global L
theta = x(1);
rho = x(2);
psi = x(3);
T(1,1) = x(4);
T(2,1) = x(5);
T(3,1) = x(6);
Rx=[cos(theta) sin(theta) 0; -sin(theta) cos(theta) 0; 0 0 1];
Ry=[cos(rho) 0 sin(rho); 0 1 0; -sin(rho) 0 cos(rho)];
```

```
Rz=[1 0 0; 0 cos(psi) sin(psi); 0 -sin(psi) cos(psi)];
R=transpose(Rx*Ry*Rz);
z = 0;
for i = 1:6
z = z + (norm(R*PSi(i,:).'+T-PIi(i,:).')^2-L(i)^2)^2;
end
```

ANEXO D: Textos Publicados Durante o Desenvolvimento do Trabalho

Control of a 6 DOF Parallel Manipulator using Rapid Prototyping

J.M. Rosario(1) H. A. Hermini(1) M. Moretti(1) D. Dumur(2)
(1)UNICAMP – Faculty of Mechanical Engineering - Laboratory of Automation and Robotics
13083-970 – Campinas, SP, Brazil
(2)SUPELEC – Department of Automatic Control
91192 – Gif-sur-Yvette, France
rosario@fem.unicamp.br http://www.fem.unicamp.br/~lar

Abstract: **This work presents a practical implementation, using reconfigurable computing applied to robotic problems. Through the proposal a hierarchical architecture, distributing the several control actions in growing levels of complexity is possible to take into account the easiness of future modifications, updates and improvements in the robotic applications. A practical example is presenting using a Stewart-Gough platform control, where the developed software and hardware are structured in independent blocks, through open architecture implementation allowing the easy expansion of the system, better adapting the platform to the tasks associated to it. This open architecture implementation allows an easy expansion of the system and a better adaptation of the platform to its related tasks.**

Keywords: **Parallel Manipulator, Rapid Control Prototyping, Simulation, Mechatronics.**

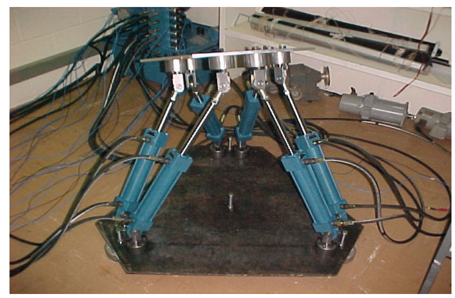
1. Introduction

The Stewart-Gough platform corresponds to a classical design for positioning and motion control, originally proposed in 1965 as a flight simulator, and still commonly used for that purpose. It is a parallel mechanism, with six linearly actuated legs with varying combinations of leg-platform connections applied in a large variety of industrial problems like manufacturing of complex forms, aerospace, automotive, nautical, and machine-tool technology [1]. Among many types of motion control platforms, this one appears of most interest being a widely accepted design for a motion control device.

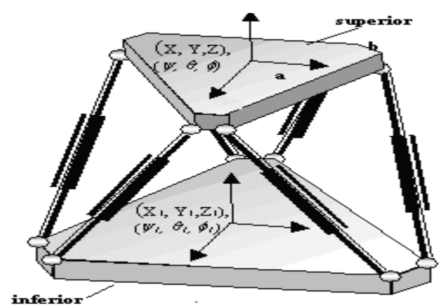
Usually, six legs are spaced around the top plate and share the load on the top plate. This differs from serial designs, such as robot arms, where the load is supported over a long moment arm. The position and orientation of the mobile platform varies depending on the lengths to which the six legs are adjusted.

This device can be used to position the platform in six degrees of freedom. In general, the top plate is triangularly shaped and is rotated 60 degrees from the bottom plate, allowing all legs to be equidistant from one another and each leg to move independently of the others.

This work presents a practical implementation based on reconfigurable computing applied to a Stewart-Gough platform. This system is used to simulate the movement of a sea tanker and within studies of cooperative robots.



a) Hydraulic actuators.



b) Referential Coordinate system.

Figure 1: Stewart-Gough Platform.

This article focuses on the study of a virtual robotic environment for the modelling and simulation of mechatronics devices, emphasizing the dynamical modelling and control development and implementation of joint position controllers of a parallel manipulator. It is organized as follows. Section II provides a mathematical description of this platform, including the kinematics and dynamics modelling and the actuator control. Section III proposes the axis control structure. Section IV is dedicated to the supervision and control architecture. Section V presents simulation results, experimental implementation and preliminary tests. Finally, section VI outlines the main conclusions.

2. Mathematical Description

The Stewart-Gough platform can accomplish a large number of complex tasks [2]. It is a 6 dof parallel mechanism that consists of a rigid body top plate or mobile plate, connected to a fixed base plate through six independent kinematics legs. These legs are identical kinematics chains, composed of a universal joint, a linear electrical actuator, and a spherical joint.

2.1 Kinematics Model

Typically, the legs are designed with an upper and lower body that can be adjusted, so that each leg has a variable length. The geometrical model of a platform expresses the position and orientation with respect to a fixed coordinate system linked at the base of the platform (Fig. 1), as function of its generalized coordinates (joints linear movements), that is:

$$X_i = f(L_i) \tag{1}$$

where $L_i = (L_1 \ L_2 \ \cdots \ L_6)$ are the joints linear position, $X_i = (X \ Y \ Z \ \psi \ \theta \ \phi)$ the position-orientation vector of a point of the platform, and

$$T(\psi,\theta,\phi) = \mathrm{rot}(x,\phi)\,\mathrm{rot}(y,\theta)\,\mathrm{rot}(z,\psi)$$

$$= \begin{bmatrix} c\phi\, c\theta & -c\phi s\theta s\psi - s\phi\, c\psi & c\phi\, s\theta\, c\psi + s\phi\, s\psi \\ s\phi\, c\theta & -s\phi s\theta\, s\psi + c\phi\, c\psi & s\phi\, s\theta\, c\psi - c\phi\, s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix} \tag{2}$$

with:

$$\theta = ATAN2\left[\frac{-n_z}{c\phi n_x + s\phi n_y}\right] \quad \phi = ATAN2\left[\frac{n_y}{n_x}\right]$$

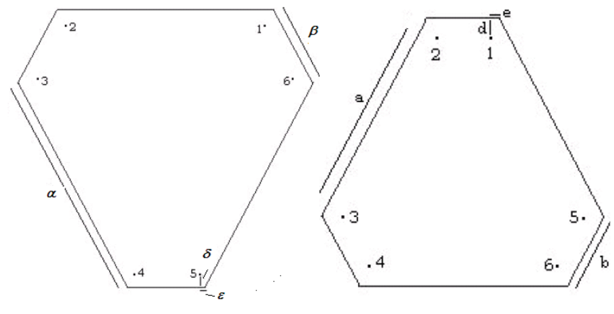
$$\psi = ATAN2\left[\frac{s\varphi a_x - c\varphi a_y}{-s\varphi s_x + c\varphi s_y}\right]$$

And the ortho-normal vector that describes the orientation:

$$\mathbf{n} = \lfloor n_x \ n_y \ n_z \rfloor \quad \mathbf{s} = \lfloor s_x \ s_y \ s_z \rfloor \quad \mathbf{a} = \lfloor a_x \ a_y \ a_z \rfloor$$

This transformation matrix (T) can be interpreted as the one that transforms the vector associated with each linear actuator into a new configuration, with the addition of a corresponding term related to the translation movement [3]. To derive the kinematics model, the superior part of the base has been idealized as an irregular hexagon, each vertex of this hexagon corresponding to an actuator connection.

This transformation matrix (T) can be interpreted as the one that transforms the vector associated with each linear actuator into a new configuration, with the addition of a corresponding term related to the translation movement [3]. To derive the kinematics model, the superior part of the base has been idealized as an irregular hexagon, each vertex of this hexagon corresponding to an actuator connection, as shown in Fig. 2.



a) Inferior base ***b) Superior base***

Figure 2: Platform model – Actuators reference points.

Further, the points that determine the movement of the superior base are the extremities of the six linear actuators settled in the inferior base of the platform. Therefore, assuming that the linear actuators have reached a final position and orientation, the problem consists in calculating the centre of mass coordinates of the superior base and the RPY angles of orientation (roll, pitch and yaw), in relation to this reference system (Fig. 2).

The relative positions of each point of attachment of the linear actuators can be derived from the parameters of movement and orientation, leading to new positions for the superior extremities of the linear actuators through an analytical calculation procedure.

The position vector of the linear actuator for the upper/lower base, P_i, P_s, determined in relation to the reference system fixed at the center of mass of the inferior part, is described through equations 3 and 4.

$$P_i = \begin{bmatrix} A_i + \varepsilon & D_i - \delta & 0 \\ -A_i + \varepsilon & D_i - \delta & 0 \\ -A_i + \varepsilon + C_i & D_i - \delta + C_i & 0 \\ -B_i + \varepsilon & D_i + \delta & 0 \\ B_i + \varepsilon & -D_i + \delta & 0 \\ A_i - \varepsilon - C_i & D_i - \delta + C_i & 0 \end{bmatrix} \tag{3}$$

$$P_s = \begin{bmatrix} A_s - e - C_s & D_s - d + C_s & h \\ B_s + e & -D_s + d & h \\ -B_s + e & D_s + d & h \\ -A_s + e + C_s & D_s - d + C_s & h \\ -A_s + e & D_s - d & h \\ A_s + e & D_s - d & h \end{bmatrix} \tag{4}$$

with $A_i = \frac{1}{2}\alpha$, $A_s = \frac{1}{2}b$, $B_i = \frac{1}{2}\beta$, $B_s = \frac{1}{2}a$

$$C_i = 2(\varepsilon - B_i)\cos(t), C_s = 2(e - B_s)\cos(t)$$

$$D_i = (A_i + B_i)\cos(t), \ D_s = (A_s + B_s)\cos(t)$$

The parameters $\alpha, \beta, \delta, \varepsilon, a, b, d, e$ are reported in Fig. 2, h represents the position of the center of mass of the superior base in the initial configuration, and each line of P_i, P_s represents the coordinates of the inferior ($A_1 \cdots A_6$) and superior ($B_1 \cdots B_6$) extremities of the actuators.

Each linear actuator is associated to a position vector $\underline{X_i}$ considering inferior extremity of the position vector for each actuator and the value of the distension associated with actuator i. With $T(\psi, \theta, \phi)$ the previous transformation matrix, $\underline{X_i^T}$ is the new associated position vector for each upper position i:

$$\underline{X_i} = T(\psi, \theta, \phi)\ \underline{X_i^T} \tag{5}$$

From the knowledge of the position of the superior base, the coordinates of the superior extremities of the linear actuators are determined by the procedures previously described, resulting in a new position, whose norm corresponds to the new size of the actuator. If $\underline{X_0}$ is the reference point, the difference between the current sizes and the desired ones is the distension that must be imposed to each actuator to reach the new position:

$$\Delta L = \left|\underline{X_i^T} - \underline{X_0}\right| - \left|\underline{X_i} - \underline{X_0}\right| \tag{6}$$

Thus, the distance of the inferior extremity of the linear actuator up to the superior extremity is calculated, where the same one is determined from the transformation of coordinates. The kinematics model of the platform needs to receive the translation information in the form of a vector and the rotation matrix in RPY angles.

This model enables to determine the appropriate axes lengths for the linear actuators so that the platform acquires the desired positioning (x, y and z coordinates, variable $j = 1, \cdots, 3$). Eqs. 7 and 8 describe respectively the length of each linear actuator k connected to the upper mobile base before and after movement:

$$L = \sqrt{\sum_{j=1}^{3} (P_s^{kj} - P_i^{kj})^2} \quad \text{with} \quad k = 1, \cdots, 6 \tag{7}$$

$$L + \Delta L = \sqrt{\sum_{j=1}^{3} (T_j^{-1}(\psi, \theta, \phi) P_s^{kj} - P_i^{kj})^2} \tag{8}$$

2.2 Inverse Kinematics

The reference input is defined through a set of displacements associated to position/orientation of the center of the platform. After interpolation, these displacements will act as reference signals for positioning controllers located at each joint, that compare the signals derived from the position sensors of the joints [4, 5]. The calculation of references in angular coordinates, referring to the tasks defined in the Cartesian space, is expressed mathematically by the inversion of the kinematics model, that is:

$$\Delta L_i = J^{-1}(\Delta \underline{X}) \tag{9}$$

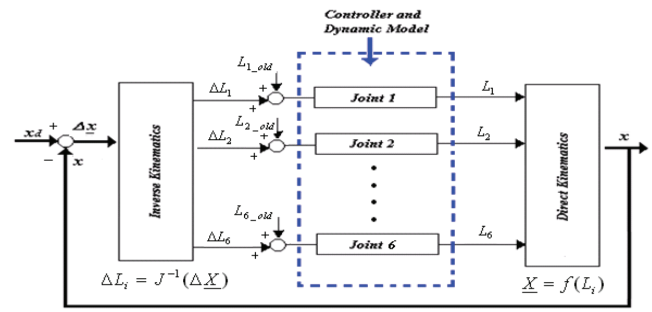


Figure 3: Kinematics Control Structure.

The controller makes corrections based on the dynamic model of the studied platform. The control structure of the joints, including the kinematics model and the control algorithms, is presented on the block diagram of Fig. 3.

The kinematics conditions may generate a system of nonlinear equations resulting in complex solutions [6,7]. Simplifications in the inverse and direct kinematics model are usually approached in the attempt of accomplishing the control of this category of manipulators. In this work, the direct kinematics is solved without coupling the equations associated to each joint movement.

2.3 Dynamic Model

The control of movements can be accomplished by the composition of individual movements of each electrical actuator; the study of the dynamical and control systems is consequently realized for each joint. To take the coupling effects into account, and to solve the trajectory problem, the dynamic control involves the determination of the inputs, so that the drive of each joint moves its links to the position values with the required speed.

The dynamic model of a 6-DOF platform can be derived through the Euler-Lagrange formulation that expresses the generalized torque [8]. The dynamic model is described by a set of differential equations called dynamic equations of motion:

$$\tau_i = J_i \ddot{L}_i + F_i \dot{L}_i + \Gamma_i \qquad i = 1, \cdots, 6 \tag{10}$$

where $\tau_i(t)$ is the generalized torque vector, $L_i(t)$ the generalized frame vector (linear joints), $J_i(t)$ the inertial matrix, $F_i(t)$ the non-linear forces (for example centrifugal) matrix, Γ_i the gravity force matrix.

Starting from the simulations of the electric actuator with its joints, a reference trajectory is generated. The controller makes the corrections taking into account the platform's dynamic model developed above. These corrections are transmitted to the system through the linear actuators described in the next subsection, including mechanical transmissions characterized by their ratio, inertia, stiffness and damping of input and output shafts. The mechanical transmission output shafts are connected to the other parts of the platform structure, which results in the effective torque reflected to each joint.

2.4 Actuator Model

Each robotic joint commonly includes a DC motor, a mechanical transmission system and an encoder. Considering the DC motor, the three classical equations are the following:

$$u(t) = L_{mot}\frac{\mathrm{d}\,i(t)}{\mathrm{d}\,t} + R_{mot}\,i(t) + K_E\frac{\mathrm{d}\,\theta_m(t)}{\mathrm{d}\,t}$$

$$T_m(t) = J_{eq}\frac{\mathrm{d}^2\,\theta_m(t)}{\mathrm{d}\,t^2} + B_{eq}\frac{\mathrm{d}\,\theta_m(t)}{\mathrm{d}\,t} \tag{11}$$

$$T_m(t) = K_T\,i(t)$$

where $T_m(t)$ is the torque, $\theta_m(t)$ the angular position of the motor axes, $i(t)$ the current, L_{mot}, R_{mot} respectively the inductance, resistance, J_{eq}, B_{eq} the inertia, friction of axis load calculated on the motor side.

A specific library has been elaborated, which includes complete axis models with controllers, motor drive, gearboxes and mechanical parts. This library enables easy change of controller's structure or motor specification.

3. AXIS CONTROL STRUCTURE

One advantage of the virtual environment that can be developed based on the previous model is the possibility of implementing and testing advanced axis control strategies, in particular Predictive Control, well known structure providing improved tracking performances. This philosophy, aiming at creating an anticipative effect using the explicit knowledge of the trajectory in the future, can be summarized as follows [9,10]:

- Definition of a numerical model of the system, to predict the future system behaviour,
- Minimization of a quadratic cost function, over a finite future horizon, using future predicted errors,
- Elaboration of a sequence of future control values; only the first value is applied both on the system and on the model,
- Repetition of the whole procedure at the next sampling period according to the receding horizon strategy.

3.1 Model Definition

The CARIMA (Controlled Autoregressive Integrated Moving Average Model) form is used as numerical model of the system, in order to cancel the steady state error, in case of a step input or disturbance, by introducing an integral term in the controller [11]. The predictive control law uses an external input-output representation form, given by the polynomial

$$A(q^{-1})\,y(k) = B(q^{-1})u(k-1) + \frac{\xi(k)}{\Delta(q^{-1})} \tag{12}$$

where u is the control signal applied to the system, y the output of the system, $\Delta(q^{-1}) = 1 - q^{-1}$ the difference operator, *A*, *B* polynomials in the backward shift operator q^{-1}, of respective order n_a and n_b, ξ an uncorrelated zero-mean white noise.

3.2 Prediction Equation

The predictive methodology requires the definition of an optimal *j*-step ahead predictor which enables to anticipate the behavior of the process in the future over a finite horizon. From the input-output model Eq. 12, a polynomial predictor is designed under the following form:

$$\hat{y}(k+j) = \underbrace{F_j(q^{-1})\,y(k) + H_j(q^{-1})\Delta u(k-1)}_{\text{free response}} + \underbrace{+\,G_j(q^{-1})\Delta u(k+j-1) + J_j(q^{-1})\xi(k+j)}_{\text{forced response}} \tag{13}$$

F_j, G_j, H_j and J_j, unknown polynomials, corresponding to the expression of the past and of the future, are derived solving Diophantine equations, with unique solutions [7].

3.3 Cost function

The GPC strategy minimizes a weighted sum of square predicted future errors and square control signal increments:

$$J = \sum_{j=N_1}^{N_2}\left(\hat{y}(k+j) - w(k+j)\right)^2 + \lambda\sum_{j=1}^{N_u}\Delta u(k+j-1)^2 \tag{14}$$

Assuming $\Delta u(t+j) = 0$ for $j \geq N_u$. Four tuning parameters are required: N_1, the minimum prediction horizon, N_2 the maximum prediction horizon, N_u the control horizon and λ the control-weighting factor.

3.4 Cost function minimization

The optimal j-step ahead predictor Eq. 13 is rewritten in a matrix form:

$$\hat{\mathbf{y}} = \mathbf{G}\,\tilde{\mathbf{u}} + \mathbf{if}(q^{-1})\,y(t) + \mathbf{ih}(q^{-1})\,\Delta u(t-1) \qquad (15)$$

with:

$$\mathbf{if}(q^{-1}) = \left[F_{N_1}(q^{-1}) \quad \cdots \quad F_{N_2}(q^{-1})\right]'$$
$$\mathbf{ih}(q^{-1}) = \left[H_{N_1}(q^{-1}) \quad \cdots \quad H_{N_2}(q^{-1})\right]'$$
$$\tilde{\mathbf{u}} = \left[\Delta u(t) \quad \cdots \quad \Delta u(t+N_u-1)\right]'$$
$$\hat{\mathbf{y}} = \left[\hat{y}(t+N_1) \quad \cdots \quad \hat{y}(t+N_2))\right]'$$

$$\mathbf{G} = \begin{bmatrix} g_{N_1}^{N_1} & g_{N_1-1}^{N_1} & \cdots & \cdots \\ g_{N_1+1}^{N_1+1} & g_{N_1}^{N_1+1} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ g_{N_2}^{N_2} & g_{N_2-1}^{N_2} & \cdots & g_{N_2-N_u+1}^{N_2} \end{bmatrix}$$

The future control sequence is obtained minimizing the criterion Eq. 14 [7]:

$$\tilde{\mathbf{u}} = \mathbf{M}\left[\mathbf{w} - \mathbf{if}(q^{-1})\,y(t) - \mathbf{ih}(q^{-1})\,\Delta u(t-1)\right] \qquad (16)$$

with: $\mathbf{M} = \mathbf{Q}\ \mathbf{G}'$, $N_u \times (N_2 - N_1 + 1)$ matrix,

$\mathbf{Q} = \left[\mathbf{G}'\mathbf{G} + \lambda\,\mathbf{I}_{N_u}\right]^{-1}$ and $\mathbf{w} = \left[w(t+N_1) \quad \cdots \quad w(t+N_2)\right]'$

Only the first value of the sequence Eq. 15 is finally applied to the system according to the receding horizon strategy.

3.5 RST form of the Controller

The minimization of the previous cost function [11] results in the predictive controller derived in the RST form according to Fig. 4 and implemented through a difference equation:

$$S(q^{-1})\Delta(q^{-1})u(t) = -R(q^{-1})\,y(t) + T(q)w(t) \qquad (17)$$

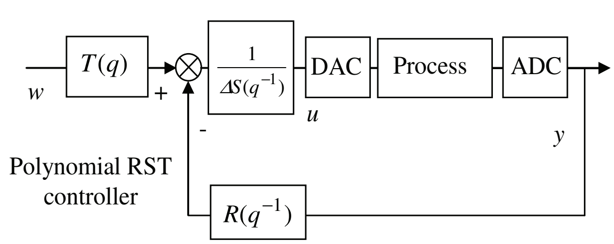


Figure 4: GPC in a RST form.

The main feature of this RST controller is the non-causal form of the T polynomial, creating the anticipative effect of this control law [1]. The degrees of the three polynomials are as follows.

$$\begin{aligned} &\text{degree}\left(R(q^{-1})\right) = \text{degree}\left(R(q^{-1})\right) \\ &\text{degree}\left(S(q^{-1})\right) = \text{degree}\left(B(q^{-1})\right) \\ &\text{degree}\left(T(q)\right) = N_2 \end{aligned} \qquad (18)$$

The GPC has shown to be an effective strategy in many fields of applications, with good time-domain and frequency properties (small overshoot, improved tracking accuracy and disturbance rejection ability, good stability and robustness margins), able to cope with important parameters variations.

4. Supervision and Control Architecture

4.1 Rapid Prototyping

The objective of this reconfigurable architecture concept is thus to enable an easy and quick adaptation and expansion of the system to these technological evolutions, for a better portability and interchangeability of the final system. Through the division of the structure in small functional blocks, with very specific dedicated interfaces, the modularization of the project becomes efficient.

Among all fields related to the complete achievement of an embedded project, hardware and software technologies have rapidly improved. That is particularly true for the evolution of motors, sensors, microprocessors, communication interfaces and power interfaces. From this, the idea is then to elaborate open structures, which may adapt very easily to the developments of all these technologies. The consequence of this requirement is the design of small independent modules, with communication interfaces, included within an open architecture oriented structure.

Using parameters of the above system, the global viability of the project has been assessed first through a dedicated virtual environment before experimental validation. However, the process of developing and implementing control strategies, including tuning phases, for this type of complex mechatronics system is extremely time-consuming [12].

In this direction, the rapid prototyping tools allow the design of integrated environments for modeling, simulating, and rapid prototyping algorithm development, with components that a) simulate the dynamic models of the complex systems, b) perform a complex simulation of the overall mechatronics system and environment, c) automatically generate code for embedded robot control and d) communicate with the platform and control it remotely.

4.2 Supervision and Control

The proposed control architecture is a set of hardware and software modules, implemented with emphasis in rapid prototyping systems, integrated to give support to development of the platform tasks [13]. The architecture is organized in several independent blocks, connected like a hierarchical structure in three control levels (Fig. 5):

- **Supervisory control:** in this higher control level, the supervision of a generic platform task can be carried out, through the execution of global control strategies. This level also allows establishing corrections in the task realization according to the sensors data information.
- **Embedded control:** this level is dedicated to the embedded software for control. The control strategies allow decision making to be performed at a local level, with occasional corrections from the supervisory control level.
- **Local control:** this area is restricted to local control strategies associated with the sensors and actuators interfaces. The strategies in this level can be implemented under the rapid prototyping framework, through FPGA, as described below. The embedded controllers may be implemented under difference equations (RST form), which appear to be a very general and useful structure in an open architecture environment, including, for example, classical PID controllers as well as more advanced control techniques such as predictive control for instance.

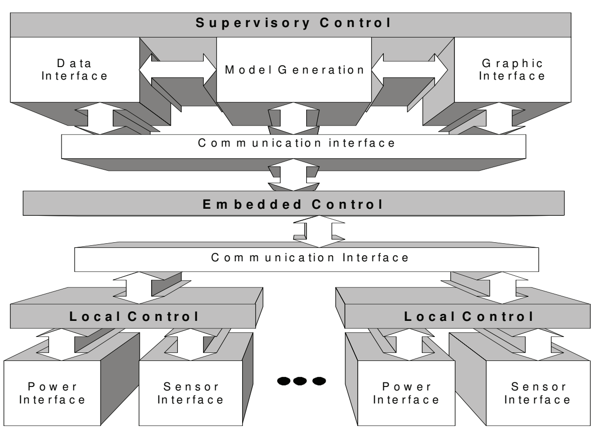


Figure 5: Stewart Platform – Control Architecture.

5. SIMULATION AND EXPERIMENTAL IMPLEMENTATION

5.1 Position Controller using FPGA

An alternative to controllers implemented by software are using reconfigurable logic [14]. The proposed controller has for objective the control of a platform with linear actuators. This programmable controller is able to process the digital signals originating from an encoder coupled to each linear actuator (ENCODER) and the digital signals of a trajectory (TRAJECTORY). For example, a PID digital controller written in a RST form can be implemented in PLD, with the gain parameters fitted through external programming. The controller's output is a digital signal for the PWM power block.

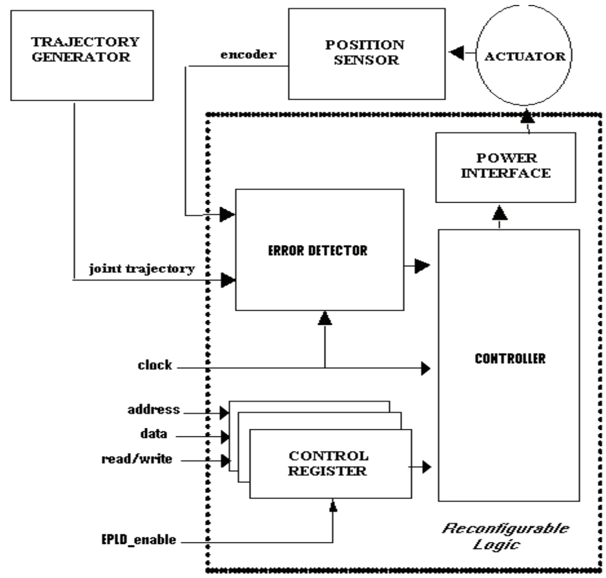
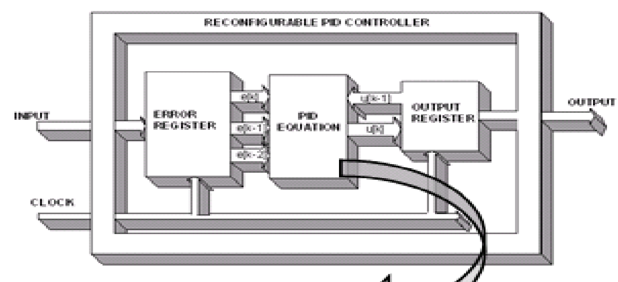


Figure 6: Implemented system - Block Diagram.




```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_signed.all;
ENTITY pid IS
  PORT (ek : IN std_logic_vector (7 DOWNTO 0);
          ekone : IN std_logic_vector (7 DOWNTO 0);
     ektwo : IN std_logic_vector (7 DOWNTO 0);
      ki : IN std_logic_vector (7 DOWNTO 0);
      kd : IN std_logic_vector (7 DOWNTO 0);
      kp : IN std_logic_vector (7 DOWNTO 0);
      ukone : IN std_logic_vector (7 DOWNTO 0);
      uk : OUT std_logic_vector (7 DOWNTO 0));

END pid;
ARCHITECTURE piddig OF pid IS
BEGIN
uk <= ukone + ek * kp + ek * ki + ek * kd + ekone * kp
     - ekone * ki  - ekone * kd + ektwo * kd;
END piddig;
```

Figure 7: Digital Controller implemented in FPGA.

The control of just one actuator is represented in Figure 7, but the synchronized control of whole actuators can be easily achieved through the same PLD. Four main blocks are observed:

- **Error Detecting Block:** comparison of the signs ENCODER and TRAJETORY.
- **PID Controller Block:** PID digital controller, using the gain parameters incorporated in the control registers.
- **Control Register Block:** responsible for the parameters programming in PLD.

- **Power Interface Block:** it converts the binary word supplied by PID controller digital signals for further use by PWM power block.

5.2 Prototyping Environment

A simulation scenario was developed for the environment related to the 6 DOF parallel manipulator, including motor drives, gear boxes, kinematic and dynamic models, and design of the control system for three axes. Simulations described below consider trajectories issued from the path generation module.

The model was tested first in Matlab-Simulink™ language and the final control hardware implementation was performed in visual programming using LabVIEW™ software (Fig. 8). This last one is used for communication purposes between the program and the control hardware of the prototype.

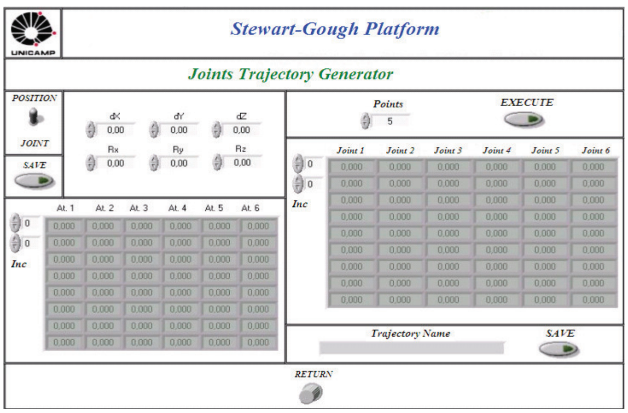
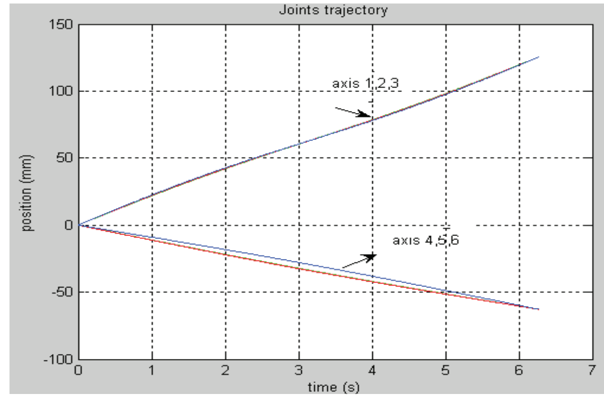


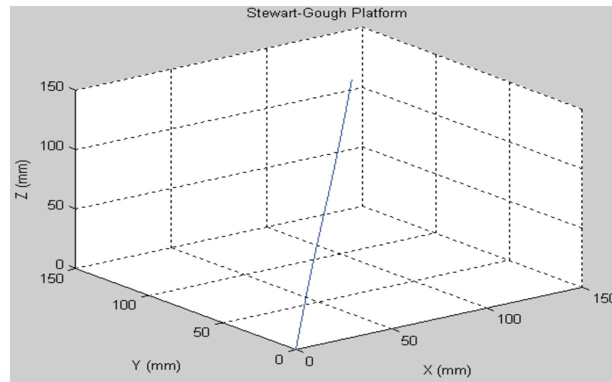
Figure 8: Model implemented in LabVIEW™.

5.3 Kinematics Model Analyis

The development of a numerical algorithm [8], which aims at finding the linear positions for a task defined with respect of the platform center in the Cartesian Space, contains the solution of the inverse kinematics through the use of recursive numerical methods based on the calculation of the kinematics model and of the inverse Jacobian matrix of the manipulator. This algorithm has been validated through different simulations, assessing the behavior of the trajectory (joint coordinate). For this purpose the kinematics model of the platform was used, with six linear joints. Fig. 9a shows the joints movements of each linear actuator and the translation displacement (45 degrees, approximately) of one point of the upper base of this platform obtained through the inverse kinematics model (Fig. 9b). Fig. 10 shows results of the proposed simulation, obtained with PID axis controllers implemented through FPGA, considering general sea movements and LABVIEW™ experimental platform.



a) Joints evolutions.



b) Trajectory description

Figure 9: Kinematics model - Simulation results.

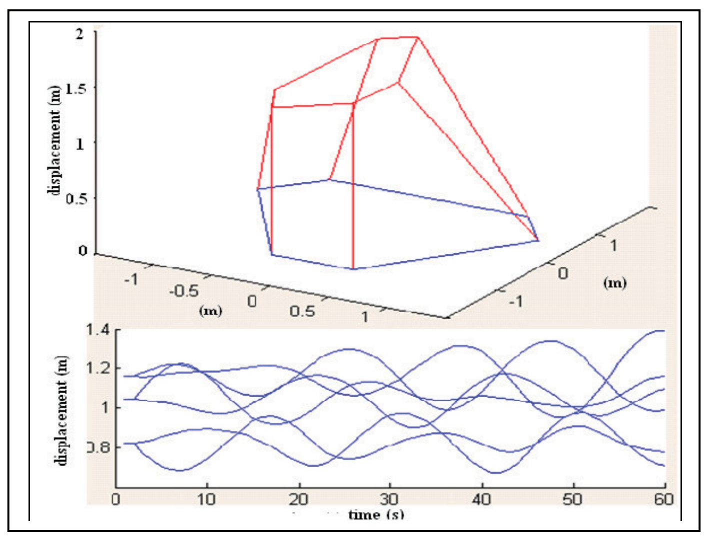


Figure 10: Joints outputs and corresponding spatial platform movements.

6. Conclusion

This paper presents the study of kinematics, dynamics and control of a Stewart-Gough platform, under a reconfigurable architecture concept, considering the division of the system in small functional blocks. This implementation consisted in merging knowledge acquired in multiple areas, and appears as a very promising design strategy for a better reconfigurability and portability of systems.

This platform also becomes a powerful benchmark for many research activities, such as the validation of controllers and supervision strategies, model generation and data transmission protocols, among others. For example, the implementation of predictive controllers on this prototype may enable the test of this advanced control strategy under severe conditions of use.

To simplify tests, implementation and future modifications, the use of rapid prototyping functions in the implementation of the interfaces and other logical blocks is emphasized in the proposed prototype. The control block, for example, can benefit of the characteristics of low consumption, high-speed operations, integration capacity, flexibility and simple programming. Some promising aspects of this architecture are:

- Flexibility, as there is a large variety of possible configurations in the implementation of solutions for several problems,
- It is a powerful tool for prototype design, allowing simple solution to control the several sensors and actuators usually present in this kind of projects,
- Possibility of modification of control strategies during operation of the platform,
- The open architecture of this platform enables the use for educational and researches activities.

ACKNOWLEDGEMENTS

CNPq supported this work and CAPES – Brazilian Council of Research, and ARCUS program, France, through a scientific collaborative work between the Control Department of SUPELEC, France and the Laboratory of Automation and Robotics, UNICAMP, Brazil

REFERENCES

[1] A. Ollero, S. Boverie, R. Goodal, “Mechatronics, Robotics and Components for Automation and Control”, *Annual Reviews in Control IFAC Journal*, No 26, 2005, pp 203-228.

[2] T.Y. Lee, J.K. Shim. Improved analytic elimination algorithm for the forward kinematics of the general Stewart–Gough platform, *Mechanism and Machine Theory*, pp. 563–577, vol. 38, 2003.

[3] J.M. Rosario, E. Oliveira, D. Dumur. Conception of Stewart-Gough Platform with Reconfigurable Control, *Proceedings of the International Symposium on Power Electronics, Electrical Drives, Automation and Motion – SPEEDAM 2006*, pp. S10.13-S10.18, 2006.

[4] M.W. Spong, M. Vidyasagar. Robot Dynamics and Control, *John Wiley & Sons*, New York, 1989.

[5] K.B. Pimenta, J.P. Souza, J.M. Rosario, D. Dumur, “Control of Robotic Joints with Generalized Predictive Control (GPC)”, *RADD’2001*, Vienna, Austria, 2001.

[6] A. Karger. Architecture singular planar parallel manipulators, *Mechanism and Machine Theory*, pages 1149-1164, v.38, 2003.

[7] I.A. Bonev. A closed-form solution to the direct kinematics of nearly general parallel manipulators with optimally located three linear extra sensors, *IEEE Trans. on Robotics and Automation*, pp. 148-156, Vol. 17, No 2, 2003.

[8] S. David, J.M. Rosario, “Modelling, Simulation and Control of Flexible Robots”, *CONTROLO’98,* pp. 532-539, Coimbra, Portugal, 1998.

[9] D. W. Clarke, C. Mohtadi, P.S. Tuffs, “Generalized Predictive Control”, Part I "The Basic Algorithm", Part II "Extensions and Interpretation, *Automatica*, 23(2): 137-160, 1987.

[10] P. Boucher, D. Dumur, “Predictive Motion Control”, in *Journal of Systems Engineering, Special Issue on Motion Control Systems*, vol. 5, Springer-Verlag, London, pp.148-162, 1995.

[11] D. Dumur, P. Boucher. New Predictive Techniques - Control Axis Solutions, *Proceedings of the 3rd IEEE Conference on Control Applications*, pp. 1663-1668, vol. 3, 1994.

[12] E.R. Cassemiro, J.M. Rosário, D. Dumur, “Robot Axis Dynamics Control using a Virtual Robotics Environment“, *Preprints of IEEE Transactions on Systems, Man and Cybernetics - ETFA’05*, pp. 305-311, Catania, Italy, 2005.

[13] C.R.E Lima, N.C. Silva, J.M. Rosário. A Proposal of Flexible Architecture for Mobile Robotics, *7th Forum International Conference and Mechatronics Education Workshop*, Atlanta, 2000.

[14] J.M. Rosário, J.M., C.R. Erig Lima, C.R., H. Ferasolli, R. Perogaro. Reconfigurable Architecture Proposal to Application on Mobile Embedded Systems Prototypes, *7th IFAC Symposium of Robot Control – SYROCO´03*, pp. 89-94, vol. I, 2003.

Chapter Number

Supervision and Control Strategies of a 6 DOF Parallel Manipulator Using a Mechatronic Approach

João Mauricio Rosário[1], Didier Dumur[2], Mariana Moretti[1], Fabian Lara[1], Alvaro Uribe[1]
[1]UNICAMP, Campinas, SP, Brazil
[2]SUPELEC, Gif-sur-Yvette, France

1. Introduction

Currently the Stewart Platform is used in different engineering applications (machine tool technology, underwater research, entertainment, medical applications surgery, and others), as an alternative solution to conventional robots with low mechatronic cost implementation, due to this trend using parallel manipulators the need for development an open supervision and control architecture is born. This chapter presents the mathematical analysis, simulation and supervision and control implementation of the six degree of freedom (DOF) parallel manipulator commonly known as the Stewart platform. The existing studies in the field are critically examined to ascertain the trends of research in the field. An analytical study of the kinematics, dynamics and control of this manipulator, includes the derivation of closed form expressions for the inverse Jacobian matrix of the mechanism and its time derivative, the evaluation of a numerical iterative scheme for on-line solution of the forward kinematics, the effects of various configurations of the unpowered joints on the angular velocities and accelerations of the links are considered, and finally the Newton-Euler formulation is used to derive the rigid body dynamic equations.

The contents if this chapter are presented as follows:

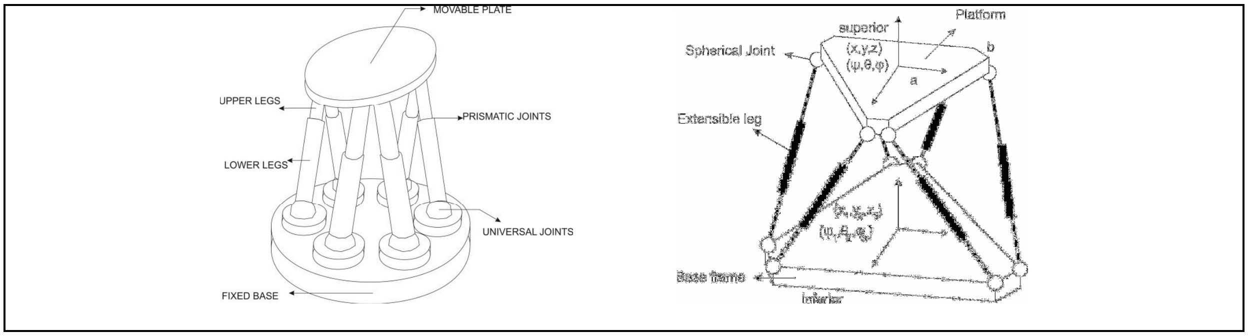
- Section II presents the features of a Stewart Platform manipulator, describing its spatial movements and applications.
- Section III covers the mathematical description, detailing on the kinematics and dynamics modelling, and the actuator control using a mechatronic prototyping approach.
- Section IV details the control structure, and compares two different control strategies, the PID joint control structure, and the Generalized Predictive Control (GPC) strategy. Both controllers structured in the polynomial RST form, as a generic framework for numerical control laws, satisfying open architecture requirements.
- Section V describes the supervision and control architecture. In particular the spatial tracking error is analyzed for both controllers, showing better accuracy with the GPC one.
- Section VI provides time domain simulation results and performance comparison for several scenarios (linear and circular displacements, translational or rotational movements), using reconfigurable computing applied to a Stewart-Gough platform.
- Section VII presents the supervisory control and hardware interface based in Labview™ environment implemented, and its experimental implementation with tests.
- Finally, the section VII presents the conclusions and contributions.

2. Stewart Platform Manipulator

The Stewart platform is a 6 DOF mechanism with two bodies connected together by six extendable legs. The manipulation device is obtained from generalisation of the mechanism proposed as a flight simulator in (Stewart, 1965)(Gough and Whitehall, 1962)(Karger, 2003)(Cappel, 1967). Its legs are connected through spherical joints at both ends, or, a spherical joint at one end a universal joint at the other. The structure with spherical joints at both ends is referred to, as the 6-SPS (spherical-prismatic-spherical) Stewart platform (Fig. 1), while the other, with an universal joint at the base and a spherical joint at the top is referred to, as the 6-

UPS (universal-prismatic-spherical) Stewart platform (Dasgupta, 1998)(Bessala, Philippe and Ouezdou, 1996).

The spatial movements of the six-axis parallel manipulator provide three translational and three rotational DOF of the movable plate, this allows position accuracy, stiffness and payload-to-weight ratio to exceed conventional serial manipulators performances. Due to these mechanical advantages, the Stewart platform manipulator is used in many applications such as flight simulators, parallel kinematics machines-tools, biped locomotion systems and surgery manipulators (Sugahara et al, 2005)(Wapler et al, 2003)(Wentlandt and Sastry, 1994).



a) Mathworks™ description b) The 6-UPS Stewart Platforms

Fig. 1 Schematic Representation of the Stewart-Gough Platform.

3. Mathematical Description

The mathematical model has to response to the following tasks: given a desired trajectory, actuate the forces to properly move the mobile plate of the platform to the target position and orientation. The referential coordinated system for analyzing the manipulator is presented in Fig. 1.

3.1 Geometric Model

Given the accomplishment of numerous tasks due to its configuration, the platform legs are identical kinematics chains, which movement varies accordingly to the typ of joint used (Fasse and Gosselin, 1998)(Boney, 2003). Typically, the legs are designed with an upper and lower adjustable body, so each one has a variable length (Fig. 1). The geometrical model of a platform is expressed by its (X, Y, Z) position and the (ψ, θ, ϕ) orientation due to a fixed coordinate system linked at the base of the platform. The obtained function of this generalized coordinates (joints linear movements), is presented in (1).

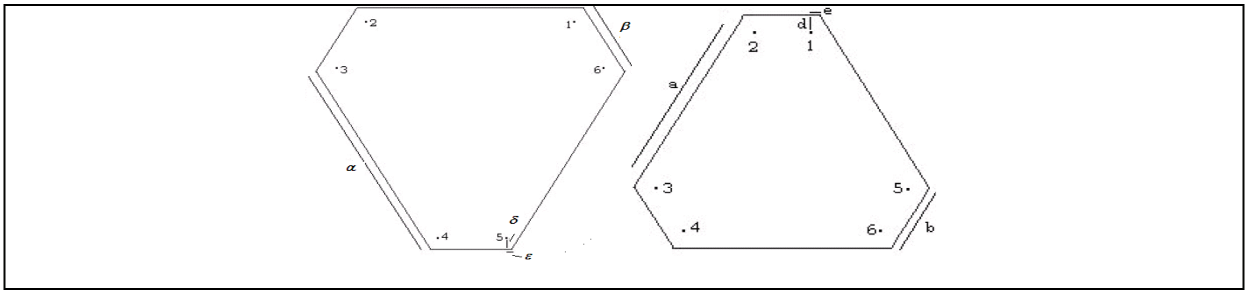
$$X_i = f(L_i) \tag{1}$$

where $L_i = (L_1 \quad L_2 \quad \cdots \quad L_6)$ are the joint's linear position, $X_i = (X \quad Y \quad Z \quad \psi \quad \theta \quad \phi)$ the position-orientation vector of a point of the platform. Then the transformation matrix for rotations can be organised as shown in (2).

$$T(\psi,\theta,\phi) = \mathrm{rot}(x,\phi)\mathrm{rot}(y,\theta)\mathrm{rot}(z,\psi) = \begin{bmatrix} c\phi\, c\theta & -c\phi s\theta s\psi - s\phi\, c\psi & c\phi\, s\theta\, c\psi + s\phi\, s\psi \\ s\phi\, c\theta & -s\phi s\theta\, s\psi + c\phi\, c\psi & s\phi\, s\theta\, c\psi - c\phi\, s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix} \tag{2}$$

where, $\theta = ATAN\,2\left[\dfrac{-n_z}{c\phi\, n_x + s\phi\, n_y}\right]$, $\phi = ATAN\,2\left[\dfrac{n_y}{n_x}\right]$, $\psi = ATAN2\left[\dfrac{s\varphi\, a_x - c\varphi\, a_y}{-s\varphi\, s_x + c\varphi\, s_y}\right]$ and

$\mathbf{n} = \lfloor n_x \quad n_y \quad n_z \rfloor$, $\mathbf{s} = \lfloor s_x \quad s_y \quad s_z \rfloor$, $\mathbf{a} = \lfloor a_x \quad a_y \quad a_z \rfloor$: are the orthonormal vectors that describe the platform's orientation.



a) Inferior base b) Superior base

Fig. 2: Platform Geometric Model – Actuators reference points.

This transformation matrix allows each actuator to change its position into a new configuration (Kim, Chungt and Youmt, 1997)(Li and Salcudean, 1997). In order to define the kinematic model, the superior base has been idealized as an irregular hexagon of which, each vertex corresponds to an actuator connection, as shown in Fig. 2.

The points that define the upper base motion are located at the extremities of the six linear actuators fixed at the lower base of the platform. Therefore, assuming that actuators have reached their final position and orientation, the problem is calculating the coordinates of the centre of mass on the superior base, and the orientation RPY angles(roll, pitch and yaw).

The relative positions can be calculated from the position and orientation analysis (using the transformation matrix), this leads to new positions within the platform workspace
The position vector of the actuator for the upper/lower base, P_i, P_s, is determined in relation to the fixed reference system at the centre of mass of the inferior part, this is described in (3). The parameters $\alpha, \beta, \delta, \varepsilon, a, b, d, e$ are reported in Fig.2, h represents the position of the center of mass of the upper base in its initial configuration, and each line of P_i, P_s represents the lower ($A_1 \cdots A_6$) and superior ($B_1 \cdots B_6$)coordinate extremities of the actuators.

$$P_i = \begin{bmatrix} A_i + \varepsilon & D_i - \delta & 0 \\ -A_i + \varepsilon & D_i - \delta & 0 \\ -A_i + \varepsilon + C_i & D_i - \delta + C_i & 0 \\ -B_i + \varepsilon & D_i + \delta & 0 \\ B_i + \varepsilon & -D_i + \delta & 0 \\ A_i - \varepsilon - C_i & D_i - \delta + C_i & 0 \end{bmatrix} \quad P_s = \begin{bmatrix} A_s + e & D_s - d & h \\ -A_s + e & D_s - d & h \\ -A_s + e + C_s & D_s - d + C_s & h \\ -B_s + e & D_s + d & h \\ B_s + e & -D_s + d & h \\ A_s - e - C_s & D_s - d + C_s & h \end{bmatrix} \tag{3}$$

where, $A_i = 0.5\alpha$, $A_s = 0.5b$, $B_i = 0.5\beta$, $B_s = 0.5a$, $C_i = 2(\varepsilon - B_i)\cos(t)$, $C_s = 2(e - B_s)\cos(t)$, $D_i = (A_i + B_i)\cos(t)$, $D_s = (A_s + B_s)\cos(t)$

Each actuator is associated to a position vector $\underline{X_i}$ considering the inferior end of the position vector and the value of the distension associated with the *ith* actuator. With transformation matrix, $\underline{X_i^T}$ is the new associated position vector for each upper position *ith*, it is obtained in (4).

$$\underline{X_i} = T(\psi, \theta, \phi)\ \underline{X_i^T} \tag{4}$$

From the known position of the upper base, the coordinates of its extremities are calculated using the previous equations, this results in new positions, whose norm corresponds to the new size of the actuator. If $\underline{X_0}$ is the reference point, then the difference between the current sizes and the desired ones is the distension that must be imposed to each actuator in order to reach its new position, this relation is presented in (5)

$$\Delta L = \left|\underline{X_i^T} - \underline{X_0}\right| - \left|\underline{X_i} - \underline{X_0}\right| \tag{5}$$

Thus, the distance between the extremities are calculated, using the transformation matrix and the known coordinates. The kinematic model of the platform receives the translation information in vector form and the rotation in a matrix one with the RPY angles.

This analysis allows defining each axes lengths for the so that the platform gets the desired positioning (x, y and z coordinates, variable $j = 1, \cdots, 3$). Eqs. 6 and 7 describe respectively the length of each linear actuator k connected to the upper mobile base before and after movement:

$$L = \sqrt{\sum_{j=1}^{3} (P_s^{kj} - P_i^{kj})^2} \quad \text{with} \quad k = 1, \cdots, 6 \tag{6}$$

$$L + \Delta L = \sqrt{\sum_{j=1}^{3} (T_j^{-1}(\psi, \theta, \phi) P_s^{kj} - P_i^{kj})^2} \tag{7}$$

The links of the platform are defined by:

$$\mathrm{A_i} = [\mathrm{rp}\cos(\alpha_\mathrm{i}), \mathrm{rp}\,\mathrm{sen}(\alpha_\mathrm{i}), 0]^\mathrm{T} = [\mathrm{A_{ix}}, \mathrm{A_{iy}}, \mathrm{A_{iz}}]^\mathrm{T} \; \alpha_\mathrm{i} = \frac{\mathrm{i}\pi - ap}{2} \; \text{for} \; \mathrm{i} = 1,3,5 \; \alpha_\mathrm{i} = \alpha_{\mathrm{i-1}} + \mathrm{ap} \; \text{for} \; \mathrm{i} = 2,4,6 \tag{8}$$

And the links of the base by:

$$\mathrm{B_i} = [\mathrm{rb}\cos(\beta_\mathrm{i}), \mathrm{rb}\,\mathrm{sen}(\beta_\mathrm{i}), 0]^\mathrm{T} = [\mathrm{B_{ix}}, \mathrm{B_{iy}}, \mathrm{B_{iz}}]^\mathrm{T} \; \beta_\mathrm{i} = \frac{\mathrm{i}\pi - ab}{2} \; \text{for i} = 1,3,5 \; \beta_\mathrm{i} = \beta_{\mathrm{i-1}} + \mathrm{ab} \; \text{for i} = 2,4,6 \tag{9}$$

Where r_p: radius of platform; r_b: radius of base; a_p: angle of platform and a_b: angle of base

3.2 Kinematic Model

The Stewart Platform Manipulator changes its position and orientation in function of its linear actuator's length. Fig. 3 shows the corresponding geometric model from the top, where the bottom base geometry is formed by the B1 to B6 points, and the upper by *P1* to *P6* points.

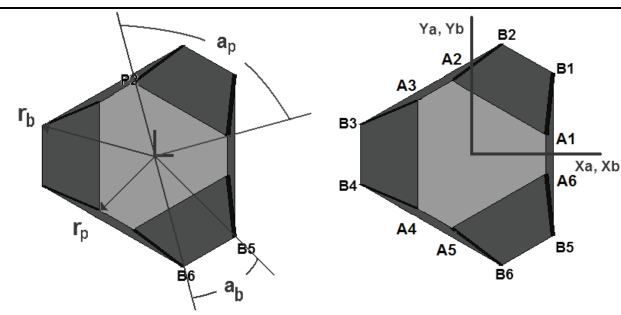


Fig. 3. Geometric model.

3.3 Inverse Kinematics

The inverse kinematics model of the manipulator expresses the joints linear motion as position and orientation function due to a fixed coordinate system at the base of the platform (Wang, Gosselin and Cheng, 2002)(Zhang and Chen, 2007), that is:

$$l = f(x) \tag{10}$$

Where, $l=(l_1,l_2,l_3,l_4,l_5,l_6)$ is the linear position of the joints, $x=(X,\ Y,\ Z,\ \psi,\ \theta,\ \varphi)$ is the position vector of the platform, X,Y,Z terms represents the cartesian position and $\psi,\ \theta,\ \varphi$ represents the orientation of the platform. The reference systems are fixed to $A(u,v,w)$ and $B(x,y,z)$ at the base, as shown in Fig. 4.

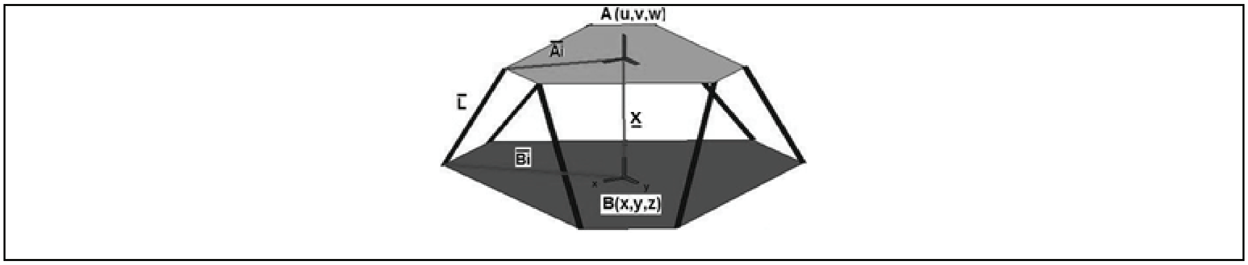


Fig. 4. Vector representation of the manipulator.

The transformation for the mobile platform´s centroid to the base, is described by the position vector x and the rotation matrix ${}^{B}R_{A}$, where,

$$ {}^{B}R_{A} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{11} $$

The angular motions are expressed as Euler angle rotations with respect to x-axis, y-axis, and z-axis, i.e. roll, pitch and yaw, in sequence.

$$ {}^{B}R_{A} = \begin{bmatrix} c\psi c\varphi & c\psi s\varphi s\theta - s\psi c\theta & c\psi s\varphi c\theta + s\psi s\varphi c\theta \\ c\psi c\varphi & s\psi s\varphi s\theta + c\psi c\theta & s\psi s\varphi c\theta - c\psi s\theta \\ -s\varphi & c\varphi c\theta & c\varphi c\theta \end{bmatrix} \tag{12} $$

Where, $c\psi$: *cos* ψ, $s\psi$: *sin* ψ

The vector-loop equation for the *ith* actuator of the manipulator is as follows:

$$ l_i = {}^{A}R_{B}\, A_i + x - B_i \tag{13} $$

Substituting the terms for each actuator, (14) describes the platform motion due to its base.

$$ \begin{aligned} l_i^{\,2} &= X^2 + Y^2 + Z^2 + r_p^{\,2} + r_b^{\,2} + 2\left(r_{11}A_{ix} + r_{12}A_{iy}\right)\left(X - B_{ix}\right) \\ &+ 2\left(r_{21}A_{ix} + r_{22}A_{iy}\right)\left(Y - B_{iy}\right) + 2\left(r_{31}A_{ix} + r_{23}A_{iy}\right)\left(Y - B_{iy}\right) + 2Z\left(r_{31}A_{ix} + r_{23}A_{iy}\right) - 2\left(XB_{ix} + YB_{iy}\right) \end{aligned} \tag{14} $$

3.4 Dynamics Study

The dynamic equations are derived for the Stewart Platform with a universal joint at the base and a spherical joint at the top of each leg. For this study, it is assumed that no rotation of any leg is allowed about its axis. Then, the kinematics and dynamics of one leg is considered and the expression for the constraining force over the spherical joint at the top of the leg is calculated. Following, this expression is simplified and for suitable derivation of the dynamic equations. Finally, kinematics and dynamics of the platform are considered so the spherical joint forces from all the six legs, allow completing the dynamic equations.

The motion control can be accomplished by the composition of individual movements of each electrical actuator; the study of the dynamic and control systems is consequently implemented for each joint (Guo and Li, 2006). For considering the coupling effects into account, and to solve the trajectory problem, the dynamic control involves the determination of the inputs, so that the drive of each joint moves its links to the position values with the required speed.

The dynamic model of a 6-DOF platform can be calculated with the Euler-Lagrange formulation that expresses the generalized torque (Jaramillo et al, 2006)(Liu, Li and Li, 2000).The dynamic model is described by a set of differential equations called dynamic equations of motion as shown in (15).

$$\tau_i = J_i \ddot{L}_i + F_i \dot{L}_i + \Gamma_i \qquad i = 1, \cdots, 6 \tag{15}$$

where $\tau_i(t)$ is the generalized torque vector, $L_i(t)$ the generalized frame vector (linear joints), $J_i(t)$ the inertial matrix, $F_i(t)$ the non-linear forces (for example centrifugal) matrix, Γ_i the gravity force matrix.

3.5 Actuator Model

Each joint has a motor, a transmission system and an encoder. Considering the DC motor (Ollero, Boverie and Goodal, 2005), the three classical equations are the following:

$$\begin{aligned} u(t) &= L_{mot} \frac{\mathrm{d}i(t)}{\mathrm{d}t} + R_{mot}\, i(t) + K_E \frac{\mathrm{d}\theta_m(t)}{\mathrm{d}t} \\ T_m(t) &= J_{eq} \frac{\mathrm{d}^2\theta_m(t)}{\mathrm{d}t^2} + B_{eq} \frac{\mathrm{d}\theta_m(t)}{\mathrm{d}t} = K_T\, i(t) \end{aligned} \tag{16}$$

where $T_m(t)$ is the torque, $\theta_m(t)$ the angular position of the motor axes, $i(t)$ the current, L_{mot}, R_{mot} respectively the inductance, resistance, J_{eq}, B_{eq} the inertia, friction of axis load calculated on the motor side.

4. Control Structure

Simulation environments allow implementing and testing advanced axis control strategies, such as particular Predictive Control, which is a well known structure for providing improved tracking performance. The purpose of the control structure is to obtain a model of the system that predicts the future system's behaviour, calculates the minimization of a quadratic cost function over a finite future horizon using future predicted errors, it also elaborates a sequence of future control values; only the first value is applied both on the system and on the model, finally the repetition of the whole procedure at the next sampling period happens accordingly to the preceding horizon strategy (Li and Salcudean, 1997)(Nadimi, Bak and Izadi, 2006)(Remillard and Boukas, 2007)(Su et al, 2004).

4.1 Model

The Controlled Autoregressive Integrated Moving Average Model (CARIMA) form is used as numerical model for the system so the steady state error is cancelled due to a step input or disturbance, by introducing an integral term in the controller (Clarke, Mohtadi and Tuffs, 1987). The predictive control law uses an external input-output representation form, given by the polynomial relation:

$$A(q^{-1})\, y(k) = B(q^{-1})\, u(k-1) + \frac{\xi(k)}{\Delta(q^{-1})} \tag{17}$$

where u is the control signal applied to the system, y the output of the system, $\Delta(q^{-1}) = 1 - q^{-1}$ the difference operator, A, B polynomials in the backward shift operator q^{-1}, of respective order n_a and n_b, ξ an uncorrelated zero-mean white noise.

4.2 Predictive Equation

The predictive method requires the definition of an optimal j-step ahead predictor which is able to anticipate the behaviour of the process in the future over a finite horizon. From the input-output model, the polynomial predictor is designed under the following form:

$$\hat{y}(k+j) = \underbrace{F_j(q^{-1})y(k) + H_j(q^{-1})\Delta u(k-1)}_{\text{free response}} + \quad + \underbrace{G_j(q^{-1})\Delta u(k+j-1) + J_j(q^{-1})\xi(k+j)}_{\text{forced response}} \tag{18}$$

F_j, G_j, H_j and J_j, unknown polynomials, corresponding to the expression of the past and of the future, are derived solving Diophantine equations, with unique solutions controller (Clarke, Mohtadi and Tuffs, 1987).

4.3 Cost Function

The GPC strategy minimizes the weighted sum of the square predicted future errors and the square control signal increments:

$$J = \sum_{j=N1}^{N2} \left(\hat{y}(k+j) - w(k+j)\right)^2 + \lambda \sum_{j=1}^{Nu} \Delta u(k+j-1)^2 \tag{19}$$

Assuming $\Delta u(t+j) = 0$ for $j \geq N_u$. Four tuning parameters are required: N_1, the minimum prediction horizon, N_2 the maximum prediction horizon, N_u the control horizon and λ the control-weighting factor.

4.4 Cost Function minimization

The optimal j-step ahead predictor (20) is rewritten in matrix form:

$$\hat{\mathbf{y}} = \mathbf{G}\,\tilde{\mathbf{u}} + \mathbf{if}(q^{-1})\,y(t) + \mathbf{ih}(q^{-1})\,\Delta u(t-1) \tag{20}$$

ith:

$$\mathbf{if}(q^{-1}) = \begin{bmatrix} F_{N_1}(q^{-1}) & \cdots & F_{N_2}(q^{-1}) \end{bmatrix}', \quad \tilde{\mathbf{u}} = \begin{bmatrix} \Delta u(t) & \cdots & \Delta u(t+N_u-1) \end{bmatrix}'$$
$$\mathbf{ih}(q^{-1}) = \begin{bmatrix} H_{N_1}(q^{-1}) & \cdots & H_{N_2}(q^{-1}) \end{bmatrix}', \quad \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(t+N_1) & \cdots & \hat{y}(t+N_2)) \end{bmatrix} \tag{21}$$

$$\mathbf{G} = \begin{bmatrix} g_{N1}^{N1} & g_{N1-1}^{N1} & \cdots & \cdots \\ g_{N1+1}^{N1+1} & g_{N1}^{N1+1} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ g_{N2}^{N2} & g_{N2-1}^{N2} & \cdots & g_{N2-Nu+1}^{N2} \end{bmatrix} \tag{22}$$

The future control sequence is then obtained minimizing the criterion (23) (Clarke, Mohtadi and Tuffs, 1987):

$$\tilde{\mathbf{u}} = \mathbf{M}\left[\mathbf{w} - \mathbf{if}(q^{-1})\,y(t) - \mathbf{ih}(q^{-1})\,\Delta u(t-1)\right] \tag{23}$$

with:

$$\mathbf{M} = \mathbf{Q}\,\mathbf{G}'\,,\ N_u \times (N_2 - N_1 + 1)\,,\ \mathbf{Q} = \left[\mathbf{G}'\mathbf{G} + \lambda \mathbf{I}_{Nu}\right]^{-1},\ \mathbf{w} = \begin{bmatrix} w(t+N_1) & \cdots & w(t+N_2) \end{bmatrix} \tag{24}$$

4.5 RST Form Controller

The minimization of the previous cost function (Clarke, Mohtadi and Tuffs, 1987), results in the predictive controller derived in the RST form according to Fig. 5 and implemented through a differential equation in (25).

$$S(q^{-1})\Delta(q^{-1})u(t) = -R(q^{-1})y(t) + T(q)w(t) \tag{25}$$

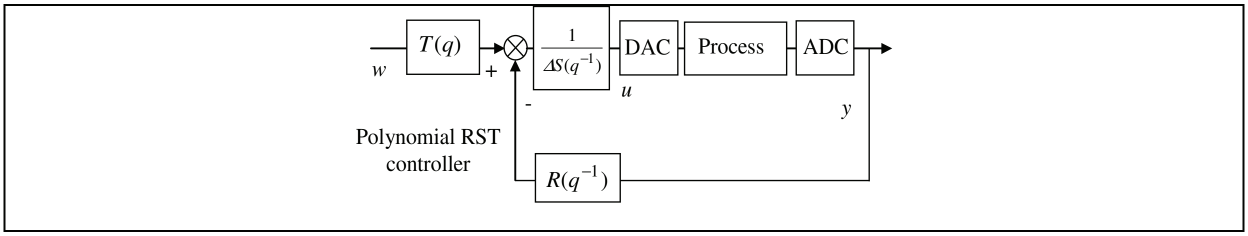


Fig. 5. GPC in a RST form.

The main feature of this RST controller is the non-causal form of the T polynomial, creating the anticipative effect of this control law.

4.6 Complete Model Implementation

The system's input is the desired trajectory x_r, From x_r and the objective is to calculate the actuator's length l_r for each sampled position. The actuator's, mechanism's and controller's dynamic effects are considered over the six legs, having as outputs the six legs δ_{ld} and the previous manipulator position x_{i-1} , it can be calculated the actual manipulator position x_o, where x_o output position is determined be the actuator's length l_o, then these values are compared with the target position in order to determinate the error δ_l between the reference position x_r and the manipulator's position x_o after all the dynamics effects have been considered (Fig. 6 and Fig. 7) (Hunt, 1978)(Jaramillo et al, 2006)(Ghobakhloo, Eghtesad and Azadi, 2006).

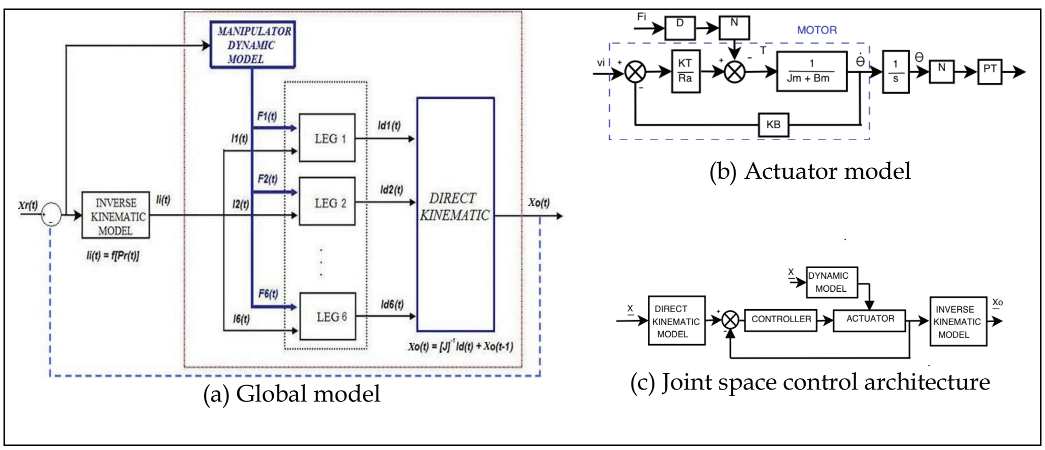


Fig. 6. Total system Model

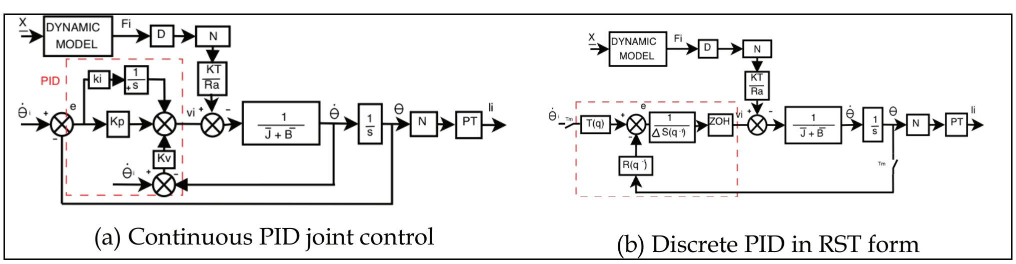


Fig. 7. Continuous and Discrete PID Controller.

$$\begin{aligned}
&\text{degree}\left(R(q^{-1})\right) = \text{degree}\left(R(q^{-1})\right)\\
&\text{degree}\left(S(q^{-1})\right) = \text{degree}\left(B(q^{-1})\right)\\
&\text{degree}\left(T(q)\right) = N_2
\end{aligned} \tag{44}$$

The GPC has shown to be an effective strategy in many fields of applications, with good time-domain and frequency properties (small overshoot, improved tracking accuracy and disturbance rejection ability, good stability and robustness margins), able to cope with important parameters variations.

5. Simulation

The modelling and simulation of the Parallel Manipulator leads to the design of a simulator adopting electric and mechanical libraries blocks using Simulink (Gosselin, Lavoie and Toutant, 1992).The main elements of the robotics joints are brushless DC motor drives, axis inertia, gears and control blocks. Other elements of the manipulator (including loads) are represented by three nonlinear models, one for each motor drive. The control system itself consists, essentially, in a cascade of control loops (for each axis). The inner speed and torque control loops are part of the drive model; only the position loop is explicitly modelled. In fact, the position control of the manipulator can be implemented through the control feedback of each isolated joint (Cappel, 1967), requiring the model of each joint.

The simulator also includes a path generation module, providing the joints with axis trajectories as reference signal for controlling the parts (Jaramillo et al, 2006). Finally, a graphic interface is developed, showing results of joint movement obtained through typical trajectories. The simulation software was implemented using Matlab. It is programmed with the equations of the Stewart Platform manipulator. This interface allows the input of the dynamic simulation parameters: mass and inertia of the mobile platform, actuator parameters and the gains of the PID controller. Fig. 8 shows a screen capture of the developed interface.

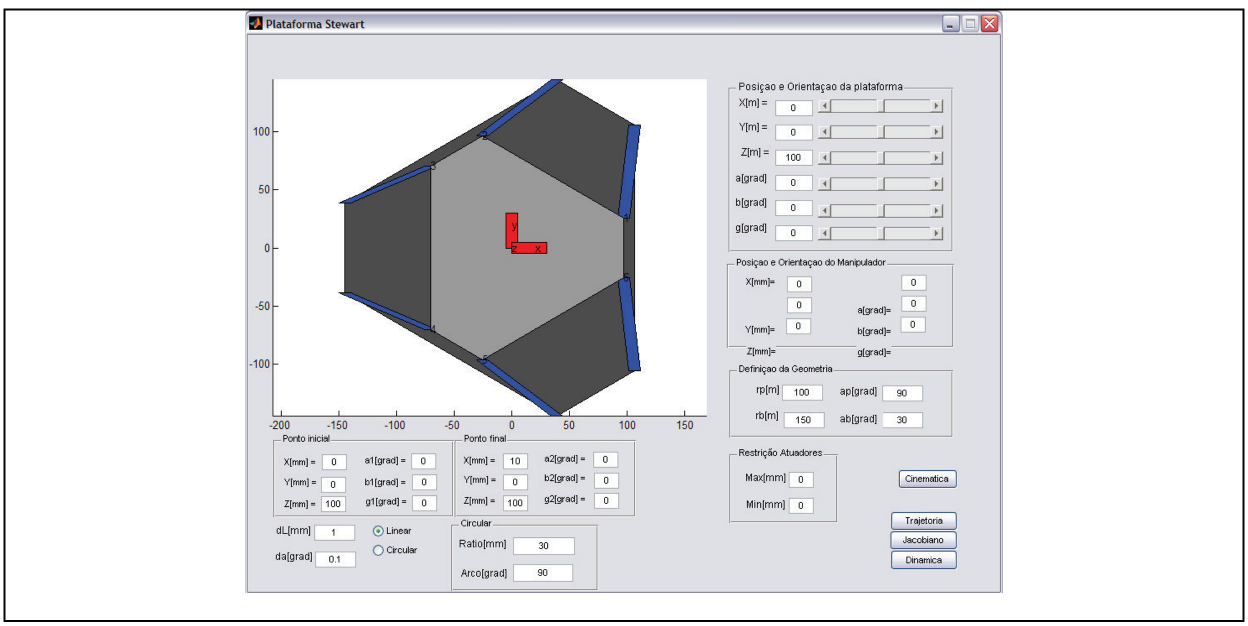


Fig. 8. Main screen of the simulation environment

In the Fig. 9 is presented the general block diagram of the total manipulator model dynamic and control Model (Fig. 3) implemented in simulink® of MATLAB.

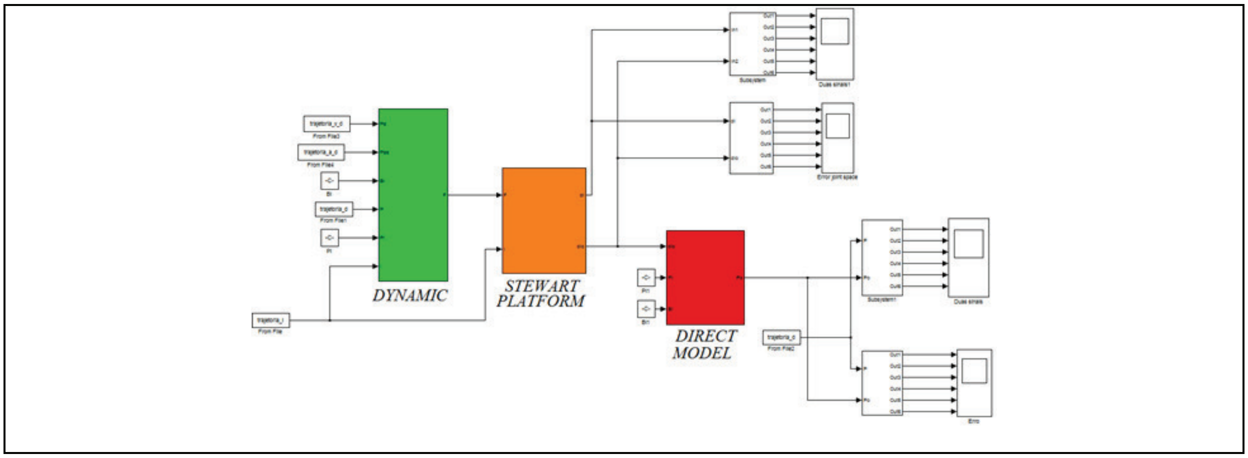


Fig.9. Simulink Dynamic and control Model

The considered system used for supervision and control implementation includes 3 DC motors, a 1:100 gear box (N), a ball screw transmission (for joint 1 only) and incremental encoders (Table 1).The joints controllers are designed independently, resulting in three RST parameters, considering the same axis motor but with different inertia on the motor side due to different geometrical features for each joint.

Jm - Inertia (kgm^2)	0.71 10^{-3}
Weight (kg)	8
Mechanical time constant (ms)	1.94
Voltage constant (V/rad/s)	0.807
Torque constant (Nm/A)	1.33
L - Inductance (mH)	14.7
R - Resistance (Ω)	1.44

Table 1. Motor Parameters.

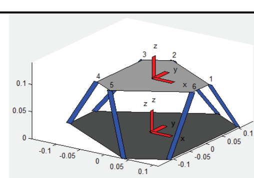
Four tuning parameters are required: N_1 the minimum prediction horizon, N_2 the maximum prediction horizon, N_u the control horizon and λ the control weighting factor. These are given in Table 2 have been chosen to provide good stability and robustness margins (Clarke, Mohtadi, and Tuffs, 1998).

Joint	N_1	N_2	N_u	λ
1	1	8	1	92
2	1	8	1	107.3
3	1	8	1	126

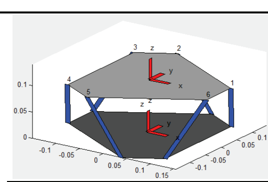
Table 2. GPC tuning parameters for each joint.

5.1 Manipulator Geometry variation: Case Study

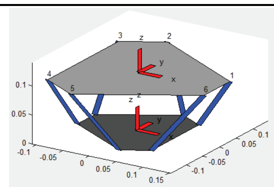
The purpose of changing the manipulator's geometry is to simulate different configurations, as can be seen from Fig. 10.



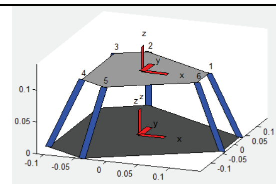
Case	rp	rb	ap	ab
1	0.1	0.15	90°	30°



Case	rp	rb	ap	ab
2	0.15	0.15	90°	30°



Case	rp	rb	ap	ab
3	0.15	0.15	90°	30°



Case	rp	rb	ap	ab
4	0.1	0.15	90°	90°

Fig. 10. Various geometry Case Studies

5.3 Motion's Simulation Based on the Path Generator

In the following example is presented a circular path trajectory on *xy* plane, the initial point is presented in Fig. 11.

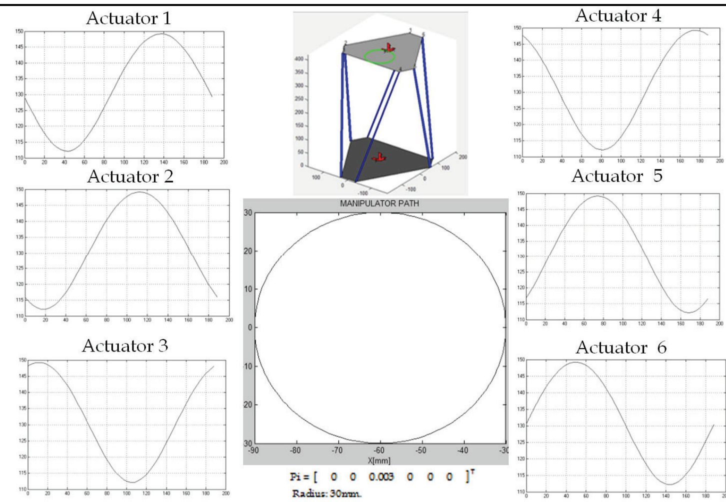


Fig. 11. Path Generator Results

The maximum velocity for this workspace trajectory is 2mm/s and the maximum acceleration is 0.1 mm/s² (Fig. 12).

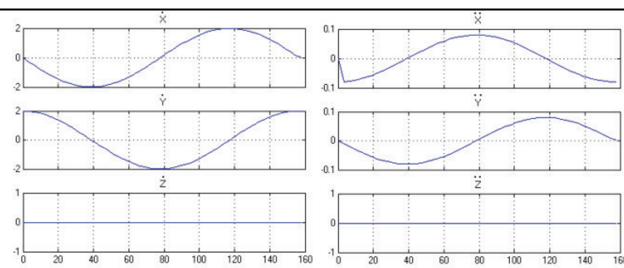


Fig. 12. a. workspace velocity b. workspace acceleration

The joint space trajectory, correspondent to the workspace trajectory imposed, for each actuator *li(t)*, is found apply the inverse kinematic model (Fig. 13).

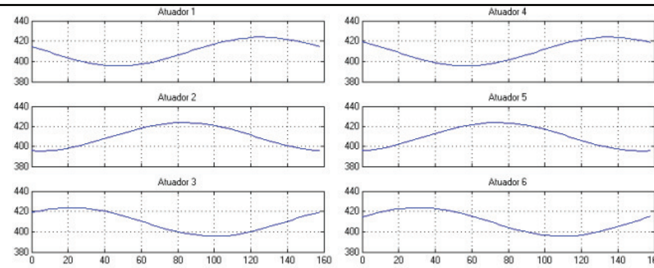


Fig. 13. Joint space trajectory

The singular configurations with various condition numbers along the trajectory are analyzed (Fig. 14). In this case, the variation of the singular number respect to Initial condition is 2.52%; with this small variation it is avoid singular configurations. In otherwise the variation of condition number would be the double or higher.

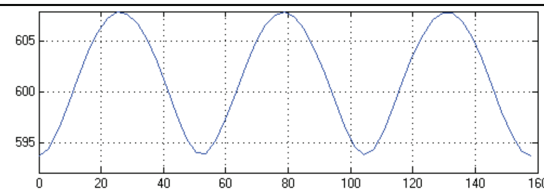


Fig. 14. Singular Analyses.

The constant workspace volume of the manipulator is also evaluated (Fig. 15). This useful characteristic helps to plan new workspace trajectories with constant orientation.

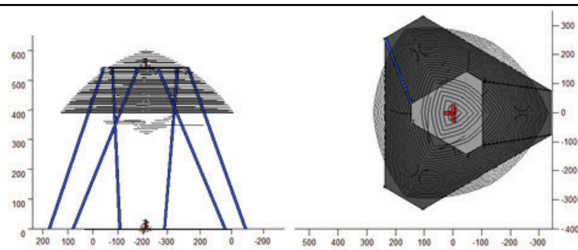


Fig. 15. Work space volume

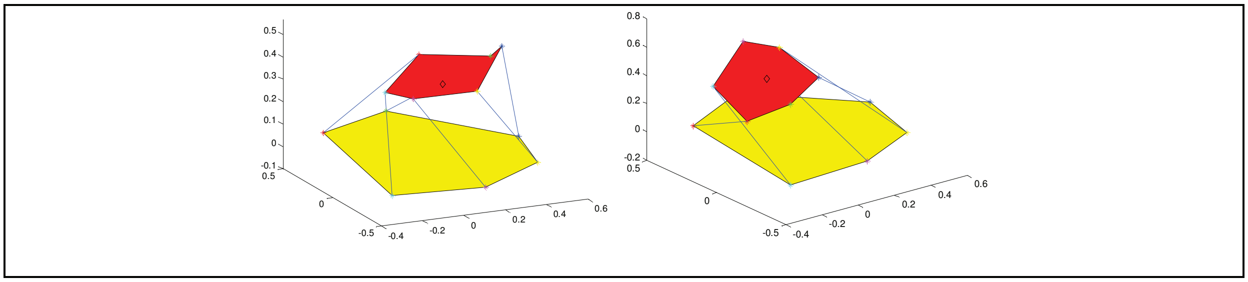
5.2 Dynamics Study

In order to perform the dynamic study and analyze the associated effect of various forces over the platform, the entire system has been modelled in a Simulink environment. The obtained model is composed of the equations block and four integrator blocks for calculating velocity and position from the known acceleration. In addition, to simplify the implementation of the simulation, platform and initial conditions are declared through an initialization button, and finally the graph button allows the visualization of the results after simulation.

The simulation tests were performed using values defined in subsection 5.1, and also the initial position of the centre of gravity of the following platform:

$$T_o = [\ 0.1 \quad 0 \quad 0.395\ ] \text{ m}$$
$$\theta_o = [\ 0.1 \quad 0 \quad -0.2\ ] \text{ rad}$$

The initial position of the 3D platform is given in Fig. 16a. A constant force of 50N applied on each arm for 0.5 s results in the final position of the platform shown in Fig. 16b, during this movement the change of position, linear and angular velocities of the center of gravity is calculated and presented in Fig.17 versus time (in sec.), it also shows the variation with time (in sec.) Lengths and sliding speeds of each arm.



(a) Initial Position (b) 50N load position

Fig. 16. Stewart Platform Positions

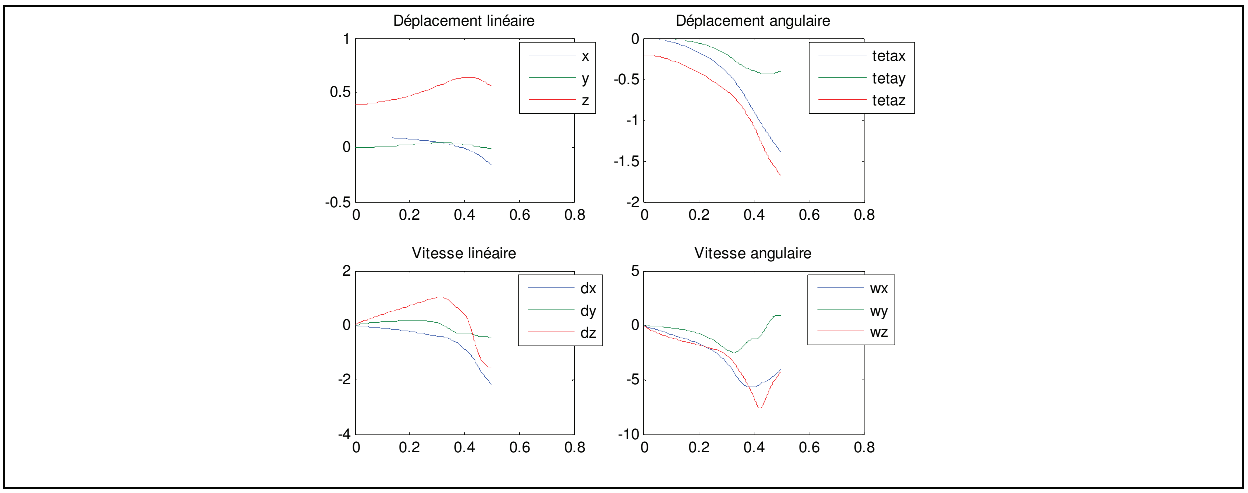


Fig. 17. Linear and angular displacement of of the center of gravity of the Stewart Platform.

5.3. Control Analysis

For joint space position control is tuning the PID controller with the following gains:
Table 1. Controller gains: K_p=100, K_i=1 and k_d=1. The input *li(t)* and the output *ld(t)* joint space trajectory are presented in the following Fig. 18.

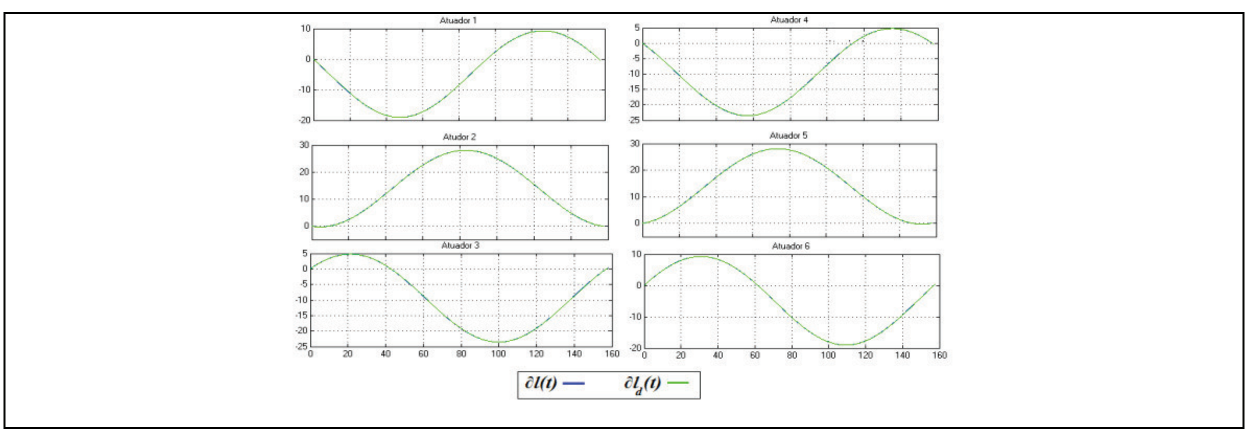


Fig. 18. Input and output joint space trajectory.

The maximum joint space error is 0.1 mm in actuator all the actuators for a maximum motion of 30 mm or 0.333%.

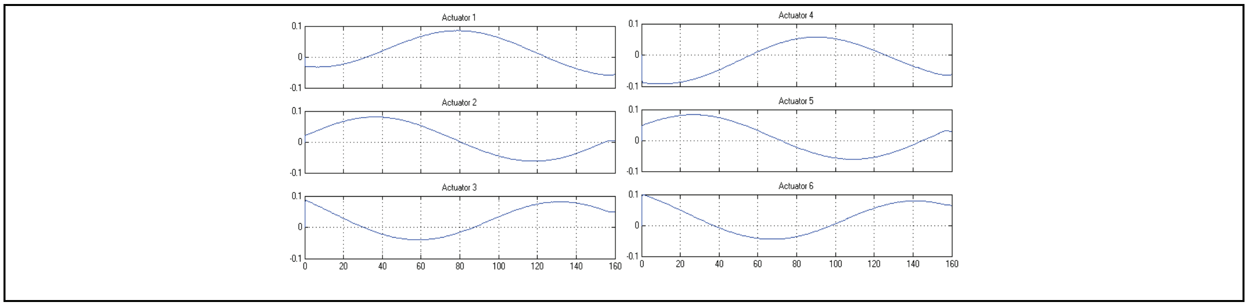


Fig. 19. Joint Space error

Appling the direct cinematic model is found the workspace output x_o and compared with the workspace input x_r.

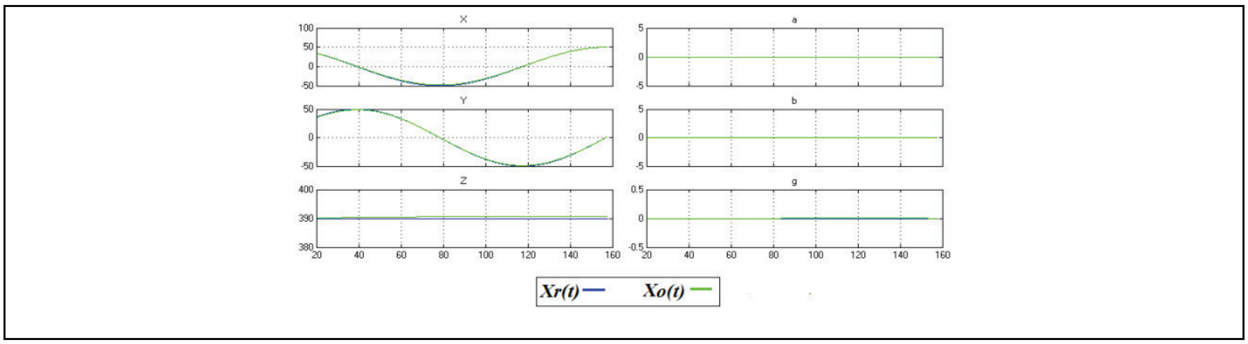


Fig. 20. Output and input workspace trajectory

The maximum work space error is 0.5 mm for a maximum linear motion of 390 mm in z axis (Fig. 21).

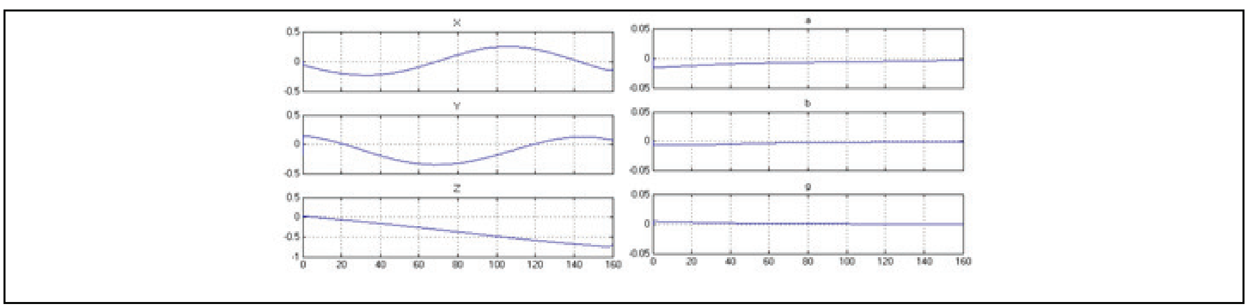
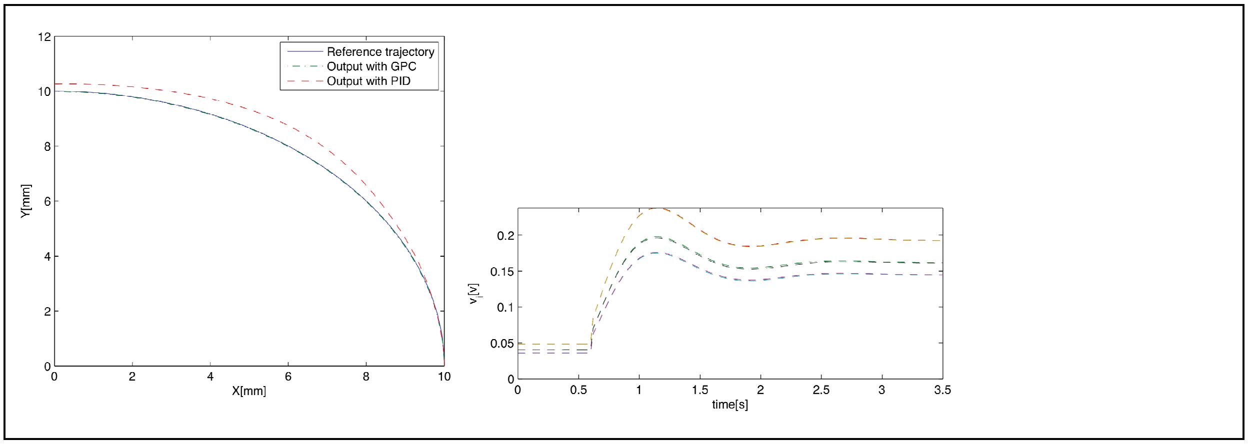


Fig. 21. Work space error



a) Tracking error b) Disturbance reaction, PID

Fig. 22. Time domain simulation results, tracking error and disturbance reaction

6. Supervision and Control Architecture

The purpose of implementing supervisory system over the platform is to permit an easy, fast adaptation and expansion of the system due to technological trends, this results in better portability and scalability of the system. Through the structure division in functional blocks, with very specific dedicated interfaces, the project implementation becomes more efficient. The rapid prototyping tools allow designing integrated environments for modelling, simulating, and testing algorithm development, with components that, simulates the dynamic models of the mechatronic systems; performs complex simulation of the overall system and environment; generates code for embedded robot control, and communicates with the platform and controls it locally or remotely (McCallion, 1977).

The proposed control architecture is a set of hardware and software modules, implemented emphasizing on rapid prototyping systems integrated to support the development of the platform tasks..

6.1 Control Levels

In the supervisory control level, the supervision of a generic platform task can be achieved through the execution of global control strategies. This level also allows correcting the task execution according to the data information obtained through the sensors. The embedded control level is dedicated to the required software for executing control strategies that allow decision making locally, with occasional corrections from the supervisory control level. The local control is restricted to local strategies associated with the sensors and actuators data. The strategies in this level can be implemented under a rapid prototyping framework, through FPGA, as described in Fig. 23.

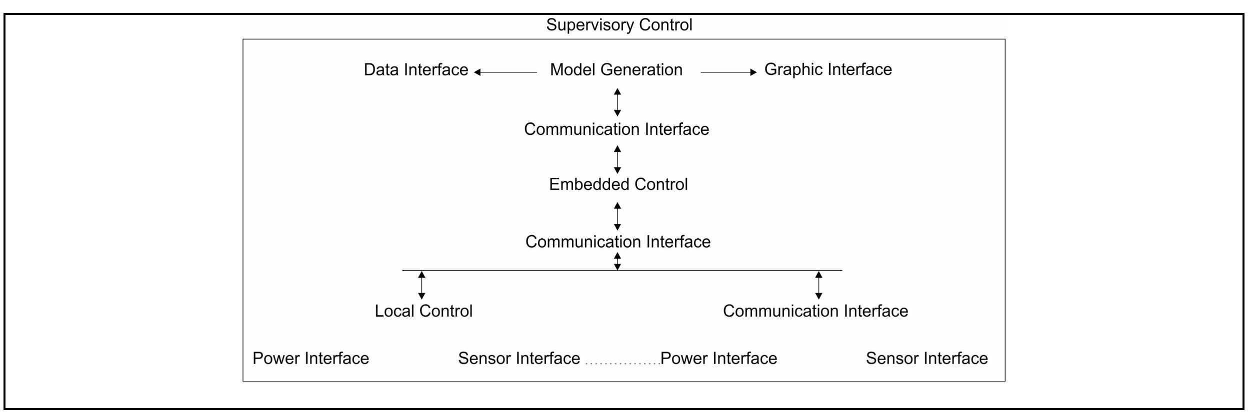


Fig. 23: Stewart-Gough Platform – Control Architecture.

6.2 Embedded Level

At the embedded control level two main tasks are implemented: the command decoder and the logic control. The first task decodes commands received by the embedded communication interface (from supervisory control), allowing

different actions to be executed according to the received data. The second task generates control signals to actuators' interfaces and receives signals from sensors' interfaces, both located at local control level. Control strategies are implemented in the logic control block.
The prototype uses a FPGA from Altera (Stratix II EP2S60) (Altera, 2008). The configware blocks were implemented in VHDL or Graphic language in the Altera's development platform Quartus II. The embedded control strategies in the logic control block were development using C++ language, in a system-on-a-programmable-chip (SOPC) environment or through the use of blocks implemented in reconfigurable hardware.

6.3 Position Control using FPGA

The proposed controller has for objective the control of a platform with linear actuators. It is able to process the digital signals from an encoder coupled to each linear actuator and the digital signals of a trajectory. For example, a PID digital controller written in a RST form can be implemented in PLD, with the gain parameters fitted through external programming. The controller's output is a digital signal for the PWM power block. Different implementations of the digital PID (Proportional-Integrative-Derivative) controllers are done, and, consequently different tuning parameters of the PID are necessary for fulfulling the different performance requirements, or to endure different levels of operating noise. A typical implementation of a PID controller can be done using a set of difference equations, as follows:

$$U[n] = P[n] + I[n] + D[n]\,,\; P[n] = Kp \cdot e[n]\,,\; I[n] = I[n-1] + \frac{Kp \cdot Ts}{2 * Ti}\left(e[n] - e[n-1]\right) \tag{45}$$

$$D[n] = \frac{(pTs-2)}{(pTs+2)} \cdot D[n-1] + \frac{2 \cdot Kp \cdot Td}{Ts \cdot (pTs+2)} \cdot \left(e[n] - e[n-1]\right) \tag{48}$$

where: U[n] is the current control signal resultant, P[n] the current proportional control signal, I[n] the current integral control signal, D[n] the current derivative signal, Kp, Ts and Ti the proportional, integral and derivative gain parameters , respectively. Also, e[n] the current error sample, and finally, e[n-1] the previous error sample. A register error block stores values of e[n] and e[n-1], and makes shift operations (e[n-1] = e[n] and u[n-1] = u[n]). An output register block stores u[n] and u[n-1].

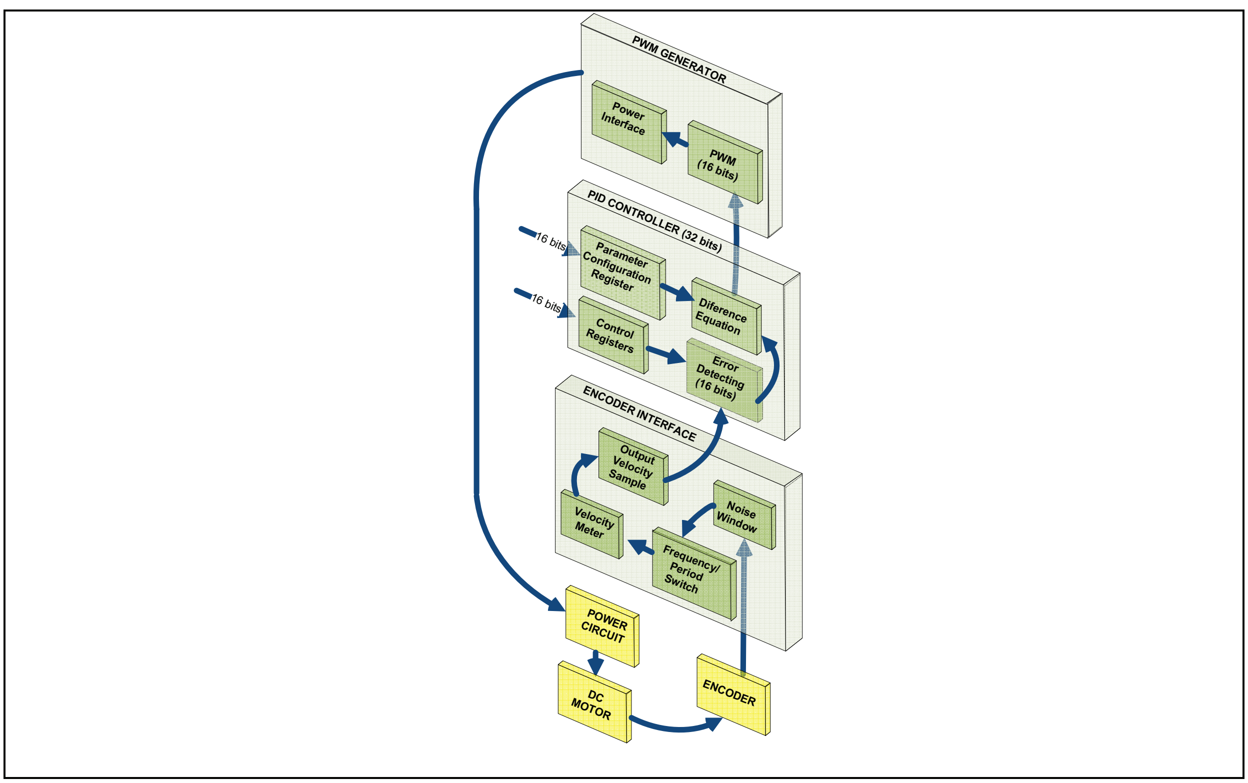


Fig. 24. Embedded DC motor control blocks.

Some blocks are described as follows: **Error Detecting** block is used for the comparison of the reference input velocity and output velocity signals, allowing the generation of a proportional binary word to the error among the periods of the signs. The current output of this block is U[n]. **Difference Equation** implements the PID digital controller, using the gain parameters (K_p, T_s and T_i) contained in the control input registers. **Control Register** implements the control registers, responsible for the programming of several operational parameters, including the gain parameters. **PWM and Power Interface** converts the binary word supplied by PID controller in a pattern of digital signs to control the PWM potency block.

The reconfiguration considered in the design of interfaces and logical blocks greatly facilitates testing, implementation and future updates. Indeed, development systems based on Reconfigurable Computing present well-suited features to this sort of project.

The system's synchronized control of whole actuators can be easily achieved through the same PLD. Four main blocks are observed: Error Detecting Block for comparing ENCODER and TRAJETORY signals. PID Controller Block using the gain parameters incorporated in the control registers. Control Register Block is responsible for the parameters programming in PLD. Power Interface Block converts the binary word supplied by PID controller digital signals for further use by PWM power block.

6.3 Prototyping Environment

A simulation tool was developed for the 6 DOF parallel manipulator, including motor drives, gear boxes, kinematic and dynamic models, and design of the control system for three axes. Simulations described below consider trajectories issued from the path generation module. The model was tested first in Matlab-Simulink language and the final control hardware implementation was performed in visual programming using LabVIEW™ software (Fig. 25). This last one is used for communication purposes between the program and the control hardware of the prototype.

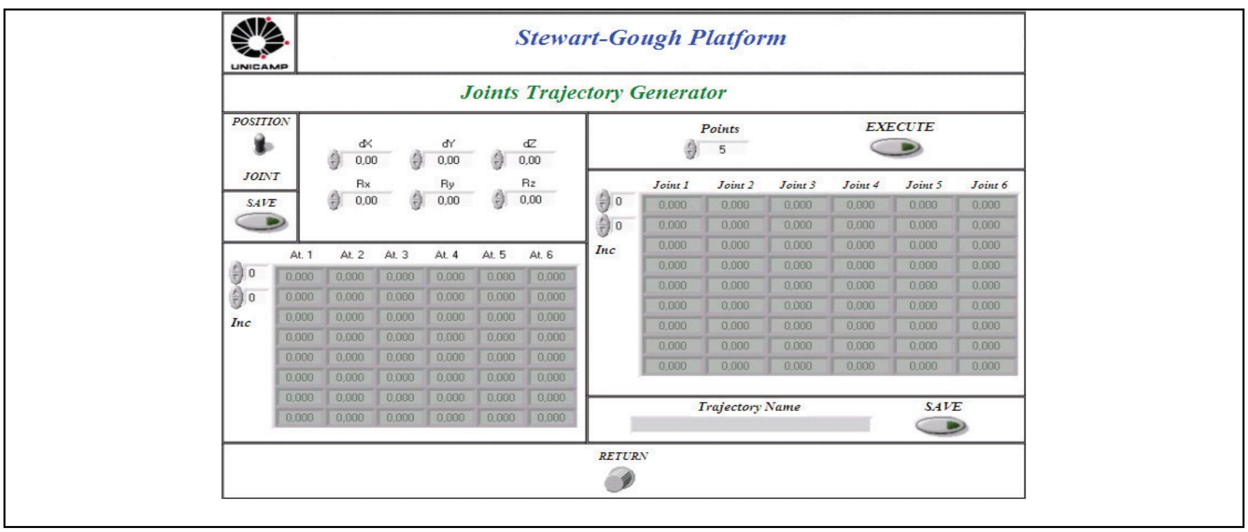
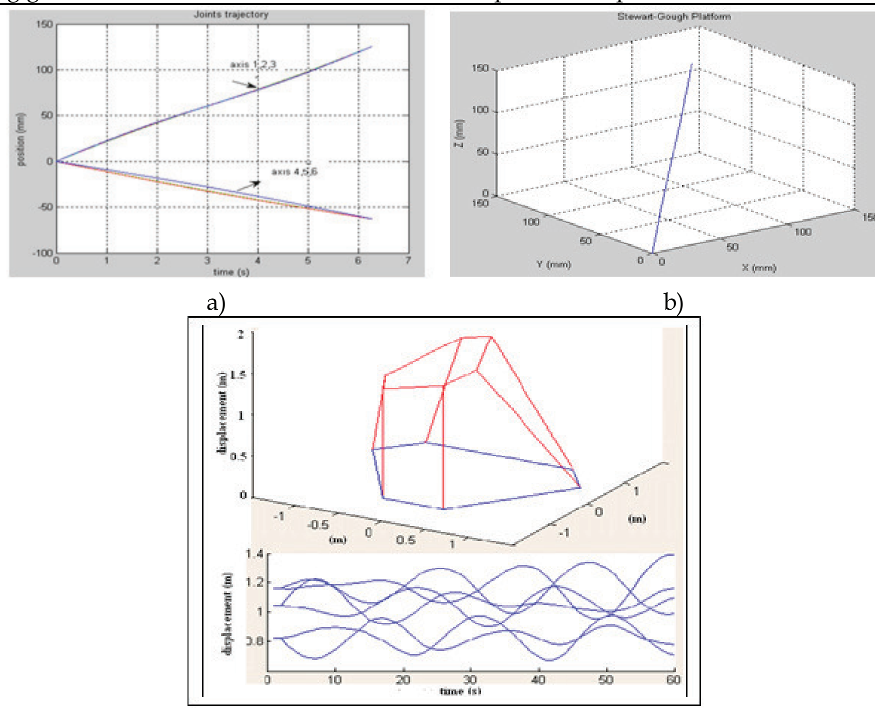


Fig. 25. Model implemented in LabVIEW™.

6.4 Experimental Results

The development of a numerical algorithm [8], which aims at finding the linear positions for a task defined with respect of the platform center in the Cartesian Space, contains the solution of the inverse kinematics through the use of recursive numerical methods based on the calculation of the kinematics model and of the inverse Jacobian matrix of the manipulator. This algorithm has been validated through different simulations, assessing the behavior of the trajectory (joint coordinate). For this purpose the kinematics model of the platform was used, with six linear joints. Fig. 261a shows the joints movements of each linear actuator and the translation displacement (45 degrees, approximately) of one point of the upper base of this platform obtained through the inverse kinematics model

(Fig. 26b). Fig. 26c shows results of the proposed simulation, obtained with PID axis controllers implemented through FPGA, considering general sea movements and LABVIEW™ experimental platform.



a) Joints evolutions. b) Trajectory description. c) Joint motion

Fig. 26. Kinematics model - Simulation results.

6. Conclusions

This chapter presents the study of kinematics, dynamics and supervision and control of a Stewart-Gough platform, under a reconfigurable architecture concept, considering the division of the system in small functional blocks. This implementation consisted in merging knowledge acquired in multiple areas, and appears as a very promising design strategy for a better reconfiguration capability and portability.

This platform also becomes a powerful benchmark for many research activities, such as the validation of controllers and supervision strategies, model generation and data transmission protocols, among others. For example, the implementation of predictive controllers on this prototype may enable the test of this advanced control strategy under severe conditions of use.

To simplify tests, implementation and future modifications, the use of rapid prototyping functions in the implementation of the interfaces and other logical blocks is emphasized in the proposed prototype. The control block, for example, can benefit of the characteristics of low consumption, high-speed operations, integration capacity, flexibility and simple programming. Some promising aspects of this architecture are:

- Flexibility, as there is a large variety of possible configurations in the implementation of solutions for several problems,
- It is a powerful tool for prototype design, allowing simple solution to control the several sensors and actuators usually present in this kind of projects,
- Possibility of modification of control strategies during operation of the platform,
- The open architecture of this platform enables the use for educational and researches activities.

8. References

Altera Corporation. http://www.altera.com, 2008.

Bessala, J.; Philippe, B. & Ben Ouezdou, F. (1996). Analytical Study of Stewart Platforms Workspaces. *Proceedings of the 1996 IEEE International Conference on Robotics and Automation.* pp. 3179-3184.

Cappel, K. (1967). Motion simulator. Patent No. 3,295,224 , US Patent No. 3,295,224.

Clarke D. W., Mohtadi C., Tuffs P.S. (1987) Generalized Predictive Control. Part I. The Basic Algorithm. Part II. *Extensions and Interpretation, Automatica*. Vol.23(2). pp. 137-160.

Dasgupta, B. & Mruthyunjaya, T. (1998). Newton-Euler formulation for the inverse dynamics of the Stewart platform manipulator. *Mechanism and Machine Theory* Vol. 33(8). pp. 135-1152.

Fasse, E. & Gosselin, C. (1998). On the spatial impedance control of Gough-Stewart platforms. *IEEE International Conference on Robotics and Automation.* pp. 1749-1754.

Ghobakhloo, A.; Eghtesad, M. & Azadi, M. (2006). Position Control of a Stewart-Gough Platform using Inverse dynamics Method with full dynamics. *International Workshop on Advanced Motion Control, AMC*. Vol. 1. pp. 50-55.

Gosselin C.; Lavoie, E. & Toutant, P. (1992). An efficient algorithm for the graphical representation of the threedimensional workspace of parallel manipulators. *Proceeding of the 22nd ASME Mechanisms Conference*. pp. 323-328.

Gosselin, C. M.; Perreault, L. & Vaillancourt, C. (1999). Simulation and Computer-Aided Kinematic Desing of Tree-Degree-of-Freedom Spherical Parallel Manipulators. *Jornal of Robotic Systems.* Vol.12(12). pp. 857-869.

Gough, V.E. e Whitehall, S. (1962). Universal tyre test machine. *Proceedings of the FISITA Ninth International Technical Congress.* pp. 117-137.

Guo, H. B. & Li, H. R. (2006). Dynamic analysis and simulation of a six degree of freedom Stewart platform manipulator. *Proceedings of the Institution of Mechanical Engineers. Part C, Journal of Mechanical Engineering Science*. Vol. 220(1). pp. 61-72.

Jaramillo-Botero, A.; Matta-Gomez, A.; Correa-Caicedo, J. F. & Perea-Castro, W. (2006). Robotics Modeling and Simulation Platform. *Robotics & Automation Magazine.* IEEE Vol. 13. pp. 62-73.

Karger, A. (2003) Architecture singular planar parallel manipulators. *Mechanism and Machine Theory*, Vol. 38, pp. 1149-1164.

Kim, D. I.; Chungt, W. K. & Youmt, Y. (1997). Geometrical Approach for the Workspace of 6-DOF Parallel Manipulators. *Proceedings of the 1997 IEEE lnternational Conference on Robotics and Automation.* pp. 2986-299 1.

Li, D. & Salcudean, S. E. (1997). Modeling, Simulation, and Control of a Hydraulic Stewart Platform. *Proceedings of the 1997 JEEE International Conference on Robotics and Automation*. Vol. 4. pp. 3360-3366.

Liu, M.; Li, C. & Li, C. (2000). Dynamics Analysis of the Gough–Stewart Platform Manipulator. *IEEE Transactions on Robotics and Automation*. Vol.16(1). pp. 94-98.

MacCallion, H. e. D. P. (1979). The analisys of six degrees of freedom work and station for mechanized assembly. *5th Congress on theory of machines and mechanisms*. pp. 616.

Nadimi, E. S.; Bak, T. & Izadi-Zamanabadi, R. (2006). Model Predictive Controller Combined with LQG Controller and Velocity Feedback to Control the Stewart Platform. *Advanced Motion Control, 2006. 9th IEEE International Workshop* Vol. 1(0.1109/AMC.2006.1631630). pp. 44-49.

Ollero, S. Boverie, R. Goodal. Mechatronics. (2005). *Robotics and Components for Automation and Control, Annual Reviews in Control IFAC Journal*. Vol. 25. pp. 203–228.

Remillard, V. & Boukas, E. (2007). Gough-Stewart Platforrn Control: A Fuzzy control approach. *Annual meeting of the North American Fuzzy Information Processing Society.* Vol. 1. pp. 108 - 113.

Stewart, D. (1965). A platform with six degrees of freedom. *Proceedings of the IMechE*. 180(15). pp. 371-385.

Su, Y. X.; Duan, B. Y.; Zheng, C. H.; Zhang, Y. F.; Chen, G. D. & Mi, J. W. (2004), 'Disturbance-Rejection High-Precision Motion Control of a Stewart Platform'(3)'IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY'.

Sugahara, Y.; Ohta, A.; Hashimoto, K.; Sunazuka, H.; Kawase, M.; Tanaka, C.; Lim, H. & Takanishi, A. (2005), 'Walking Up and Down Stairs Carrying a Human by a Biped Locomotor with Parallel Mechanism', Intelligent Robots and Systems International Conference on Volume, 1489 - 1494.

Wang, J.; Gosselin, C. & Cheng, L. (2002). Modeling and simulation of robotic systems with closed kinematic chains using the virtual spring approach. *Multibody System Dynamics*. Vol. 7(2). pp. 145-170.

Wendlandt, J. M. & Sastry, S. S. (1994). Design and Control of a Simplified Stewart Platform for Endoscopy. *Proceedings of the 33rd conferenceon Decision and Control*. Vol.1. pp. 357-362.

Zhang, Z. & Chen, T. (2007). Modeling and Movement Simulation of a Manipulator of 6-DOF Based on Stewart Platform with Pro/E. *10th IEEE International Conference on Computer-Aided Design and Computer Graphics.* pp. 533 - 536.