

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Identificación de parámetros dinámicos de un Robot 3-PRS por *Máxima Verosimilitud*



Grado en Ingeniería
en Tecnologías Industriales

Trabajo Fin de Grado

D. Julen Bacaicoa Diaz

Dr. Xabier Iriarte Goñi

Pamplona, junio 2017



Dime y lo olvido, enséñame y lo recuerdo, involúcrame y lo aprendo.

B. Franklin

RESUMEN

El presente proyecto se presenta como *Trabajo de Fin de Grado* realizado en la *Universidad Pública de Navarra*, en el Departamento de Ingeniería Mecánica, Energética y de Materiales, titulado "*Identificación de parámetros dinámicos de un Robot 3-PRS por Máxima Verosimilitud*" y trata sobre la estimación de los parámetros dinámicos que caracterizan en este caso un robot paralelo del tipo 3-PRS.

El robot empleado para la realización del proyecto, se encuentra localizado en la *Universidad Politécnica de Valencia*. Se trata de un robot manipulador paralelo de 3 GDL, y está constituido por una plataforma móvil, una plataforma fija y tres cadenas cinemáticas del tipo PRS.

La identificación de parámetros dinámicos se ha realizado mediante el método de estimación por *Máxima Verosimilitud (Maximum Likelihood Estimation)*, que consiste en calcular el valor de los parámetros que más probabilidad tienen de ser los que han provocado los datos medidos.

El proyecto comienza con el modelado del robot 3-PRS, para ello en primer lugar, se han desarrollado el modelo cinemático y el modelo dinámico del robot. En ellos, se han planteado la resolución de los problemas de posición, velocidad y aceleración, se han caracterizado los torsores considerados para su modelización, también por medio del *Principio de las Potencias Virtuales* se han obtenido las ecuaciones dinámicas del robot mediante el software *3d_mec* desarrollado por la *UpNa* y seguido, se ha expresado el modelo dinámico de manera lineal con respecto a los parámetros a identificar.

Posteriormente, se ha realizado la reducción del modelo dinámico a parámetros base a través de la supresión de las columnas nulas de la matriz de observación junto con sus parámetros dinámicos asociados y la aplicación del método simbólico de transferencias inerciales.

Una vez adentrados en la estimación de los parámetros, por medio del código desarrollado en *Matlab®* se ha aplicado el método de estimación por *Máxima Verosimilitud* y se han obtenido los parámetros dinámicos del robot 3-PRS.

A modo de conclusión, con los valores de los parámetros dinámicos estimados se ha analizado el ajuste entre los datos experimentales y aquellos proporcionados por el modelo, además de alguna propuesta de trabajo futuro entorno al robot 3-PRS vinculado con la estimación de los parámetros dinámicos por *Máxima Verosimilitud*.

Palabras clave: Identificación de parámetros, estimación por Máxima Verosimilitud, robot 3-PRS

ABSTRACT

This present project is presented as *End of Degree Work* carried out at the *Public University of Navarra*, in the Department of Mechanical Engineering, Energetic and Materials, titled "*Identification of dynamic parameters of a 3-PRS Robot by Maximum Likelihood*" it's about the estimation of the dynamic parameters that characterize in this case a parallel robot of the type 3-PRS.

The robot used for the completion of the project, is located at the *Polytechnic University of Valencia*. It is a parallel manipulator robot of 3 GDL, and is constituted by a mobile platform, a fixed platform and three kinematic chains of the type PRS.

The identification of dynamic parameters has been carried out by the method of *Maximum Likelihood Estimation*, which consists in calculating the value of the parameters that are most likely to be those that have caused the measured data.

The project begins with the modeling of the robot 3-PRS, for first, it has been developed the kinematic model and the dynamic model of the robot. In them, the resolution of the problems of position, velocity and acceleration have been considered, the torsors considered for their modeling have been characterized, also, through the *Principle of the Virtual Powers* the dynamic equations of the robot have been obtained through the software *3d_mec* developed by the *Public University of Navarra* and followed, the dynamic model has been expressed in a linear way with respect to the parameters to be identified.

Subsequently, the reduction of the dynamic model to base parameters was carried out through the suppression of the null columns of the observation matrix along with its associated dynamic parameters and the application of the symbolic method of inertial transfers.

After the parameters estimation, using the code developed in *Matlab*[®], the *Maximum Likelihood* estimation method was applied and the dynamic parameters of the 3-PRS robot were obtained.

As a conclusion, with the values of the dynamic parameters estimated, it has been analyzed the fit between the experimental data and those provided by de model, besides, some proposal of future work around the robot 3-PRS related to the area of estimation of the dynamic parameters by *Maximum Likelihood* are presented.

Key words: Parameters identification, Maximum likelihood estimation, robot 3-PRS

LABURPENA

Honako proiektu hau, *Nafarroako Unibertsitate Publikoan* egindako, Ingenieritza Mekanika, Energetikoa eta Materialen Departamentuan, *Gradu Amaierako Lan* bezala aurkezten da. Proiektu hau, izendatua “3-PRS Robot baten parametro dinamikoen identifikazioa Egiantz Handieneko metodoaren bidez” 3-PRS robot paralelo bat karakterizatzen dituzten parametro dinamikoen estimazioan datza.

Proiektu hontarako erabili den robota, *Valentziako Unibertsitate Politeknikoan* aurkitzen da. Robota, 3 askatasun maila dituen manipulatzailerako paraleloa da, eta plataforma mugikor, plataforma finko eta PRS motako 3 kate zinematikoz osatua dago.

Parametroen identifikazioa *Egiantz Handieneko* estimazio metodoaren bidez egin da, método honek neurtutako datuak eragin duten probalitate gehiago duten parametroen kalkuluan datza.

Proiektua 3-PRS robotaren modelaketarekin hasten da, hortarako lehen, modelo zinematikoa eta modelo dinamikoa garatu dira. Haietan, posizio, abiadura eta azelerazio arazoaren ebazpena planteatu dira, modelizazioan kontsideratu diren tortsoreak karakterizatu dira, baita ere, *Potentzia Virtualen Printzipioaren* bidez robotaren ekuazio dinamikoak lortu dira *Nafarroako Unibertsitate Publikoak* garatutako *3d_mec* software-arekin, eta jarraituta, modelo dinamikoa modu linealean adierazi egin da estimatzear parametro dinamikoekiko.

Geroago, modelo dinamikoaren erredukzioa egin da parametro baseetara. Horretarako, behaketa matrizearen zutabe baliogabeak ezabatu dira eta hauei erlazionatuta dauden parametro dinamikoekin batera, eta inertzia transferentzia bidezko metodo simbolikoa aplikatu da.

Behin parametroen estimazioan barreiatua, *Matlab*[®]-en garatutako kodearen bidez *Egiantz Handieneko* metodoa aplikatuz 3-PRS robotaren parametro dinamikoak lortu dira.

Konklusio gisa, estimatutako parametro dinamikoen balioekin egiaztatu da lortutako modeloa, neurtu egin diren datu esperimentalei estutzen dela. Gainera, etorkizuneko laneko proposamen batzuk egin dira 3-PRS robotaren inguruan, *Egiantz Handieneko* parametro dinamikoen estimazioaren arloan.

Hitz klabeak: Parametroen identifikazioa, Egiantz Handieneko estimazioa, robot 3-PRS

I-AGRADECIMIENTOS

En primer lugar, agradecer a mi director de *Trabajo Fin de Grado* el Dr. Xabier Iriarte, por todas esas mañanas y tardes en las que me guiabas, me ayudabas para que esto saliera adelante, y por todos esos buenos consejos. Todo el intercambio mutuo de correos electrónicos ha tenido su fruto, y al final, el trabajo que parecía que nunca iba terminar, aquí está. Mila esker Xabi.

Por otro lado, agradecer la colaboración de Aitor Plaza, aunque fue efímera, por todos esos intentos fallidos de puesta en marcha del manipulador *Hexaglide*. No me cabe la menor duda que tarde o temprano conseguiréis ponerlo en marcha y será objeto de algún otro proyecto.

También, dar las gracias a mis compañeros de clase, a ese grupo de *Euskalindustriales* que ha hecho que esta experiencia universitaria haya sido más liviana y agradable. Agradeceros toda la ayuda y colaboración que me habéis prestado en estos últimos años.

Y finalmente, y no menos importante, dar las gracias a aquellos que siempre me apoyaron desde el primer día que empecé el grado hasta el último día, mi familia y amigos. Por apoyarme y animarme, en este mundo “tan raro y complicado” como es la ingeniería.

Mila esker guztioi.

II-ÍNDICE GENERAL

I-AGRADECIMIENTOS	10
II-ÍNDICE GENERAL	12
III-ÍNDICE FIGURAS	16
IV-ÍNDICE DE TABLAS	16
V-NOTACIÓN	18
VI-Nomenclatura.....	18
VII-Operadores	21
VIII-Funciones	21
IX-Acrónimos	22
1. INTRODUCCIÓN	24
1.1. Conceptos generales.....	24
1.2. Alcance del trabajo.....	24
1.3. Antecedentes	24
1.4. Objetivos	25
1.5. Estado del arte	25
1.5.1. Modelado de sistemas mecánicos.....	26
1.5.2. Reducción de modelos dinámicos	27
1.5.3. Diseño de experimentos	29
1.5.4. Estimación.....	31
1.5.5. Validación	33
2. MODELADO DE SISTEMAS MECÁNICOS	34
2.1. Introducción	34
2.2. Modelado cinemático	35
2.2.1. Sistema de acotación	35
2.2.2. Ecuaciones cinemáticas.....	36
2.2.3. Resolución del problema de posición.....	36
2.2.4. Resolución del problema de velocidad.....	36
2.2.5. Resolución del problema de aceleración	37
2.3. Modelado dinámico.....	37
2.3.1. Parámetros dinámicos	37
2.3.2. Principio de las Potencias Virtuales	38
2.3.3. Elección de las velocidades virtuales	39
2.3.4. Caracterización de los Torsores	43

2.4.	Reducción del modelo dinámico	46
2.4.1.	Reducción del modelo a parámetros base	46
3.	MÉTODOS DE ESTIMACIÓN	48
3.1.	Introducción	48
3.2.	Fundamento estadístico	48
3.2.1.	Introducción	48
3.2.2.	Distribución Normal.....	49
3.3.	Estimación por mínimos cuadrados (Least Squares Estimation)	51
3.4.	Estimación por Máxima Verosimilitud (Maximum Likelihood Estimation)	53
4.	MODELADO DE UN ROBOT 3-PRS	56
4.1.	Introducción	56
4.1.1.	Características principales.....	56
4.1.2.	Robots paralelos	58
4.1.3.	Robot 3-PRS.....	59
4.2.	Modelo Cinemático de un robot 3-PRS	60
4.2.1.	Ecuaciones cinemáticas del robot 3-PRS	63
4.3.	Modelo Dinámico de un robot 3-PRS	63
4.3.1.	Torsores de inercia y gravedad.....	63
4.3.2.	Torsores de las fuerzas externas	63
4.3.3.	Torsores de enlace	64
4.3.4.	Torsores de fricción viscosa	64
4.3.5.	Torsores de fricción constante de Coulomb	65
4.3.6.	Elección de las velocidades virtuales del robot 3-PRS	66
4.4.	Reducción del modelo dinámico del robot 3-PRS	66
4.5.	Adquisición de datos	70
4.6.	Parámetros base del robot 3-PRS	71
5.	APLICACIÓN DEL MÉTODO DE ESTIMACIÓN POR MV A UN ROBOT 3-PRS Y RESULTADOS	74
5.1.	Introducción	74
5.2.	Parametrización de la trayectoria	74
5.3.	Desarrollo del método Máxima Verosimilitud.....	77
5.4.	Parámetros dinámicos estimados.....	78
5.5.	Validación del modelo dinámico.....	78
5.5.1.	Comparativa Trayectoria realizada vs. Trayectoria realizada estimada	79
5.5.2.	Comparativa Fuerza ejercida vs. Fuerza ejercida estimada.....	80
5.5.3.	Relación entre la velocidad y la fuerza en los actuadores	81

6. CONCLUSIONES	82
7. LÍNEAS FUTURAS DE TRABAJO	86
8. BIBLIOGRAFÍA	88
9. ANEXOS	92
9.1. Código para la obtención ecuaciones dinámicas robot 3-PRS, mediante el software 3d_Mec.....	92
9.1.1. Robot_3_prs.mec.....	92
9.2. Algoritmo para la obtención modelo dinámico y reducción del modelo del robot 3-PRS, mediante el software Matlab®	114
9.2.1. main_symbolic.m.....	114
9.2.2. parameters_definitions.m	119
9.2.3. parameters_vectors.m.....	120
9.2.4. kinematic_eq.m	120
9.2.5. dynamic_eq_origin.m.....	124
9.2.6. main_numeric.m.....	141
9.2.7. constants.m.....	143
9.2.8. get_trajec_parameters.m	144
9.2.9. Fourier_matrix.m.....	144
9.2.10. pos_prs.m	145
9.2.11. vel_prs.m.....	146
9.2.12. ace_prs.m.....	146
9.2.13. evalsign.m	146
9.2.14. evalomega.m	146
9.3. Algoritmo identificación parámetros dinámicos mediante el método de Máxima Verosimilitud.....	147
9.3.1. main_MLE.m	147
9.3.2. evalK.m	149
9.3.3. get_variances.m.....	151
9.3.4. trajectory_param_estim.m.....	151

III-ÍNDICE FIGURAS

Figura 1. Bloque identificador	48
Figura 2. Distribución de probabilidad alrededor de la media en una distribución $N(\mu, \sigma)$	50
Figura 3. Elementos estructurales de un robot industrial	56
Figura 4. Distintos tipos de articulación de un robot: a) lineal, b) rotacionales.....	57
Figura 5. Punto terminal de un manipulador.....	57
Figura 6. Ejemplo de robot paralelo de 6 GDL (hexápodo).....	58
Figura 7. Robot paralelo 3-PRS real	59
Figura 8. Representación gráfica del robot 3-PRS en 3d_mec.....	60
Figura 9. Representación CAD robot 3-PRS	61
Figura 10. Parámetros geométricos relativos de giros y traslaciones de las referencias.	61
Figura 11. Representación gráfica de los puntos de referencia del robot 3-PRS	62
Figura 12. Un periodo de la trayectoria realizada por los 3 actuadores del robot 3-PRS	75
Figura 13. Diferencia entre la trayectoria realizada y estimada por MV	79
Figura 14. Fuerza realizada y estimada por MC para cada uno de los actuadores.....	80
Figura 15. Fuerza realizada y estimada por MV para cada uno de los actuadores.....	80
Figura 16. Trayectoria realizada y velocidad del actuador 1	81

IV-ÍNDICE DE TABLAS

Tabla 1. Parámetros base simbólicos del robot 3-PRS	71
Tabla 2. Parámetros de la trayectoria estimados para los 3 actuadores del robot 3-PRS	76
Tabla 3. Parámetros dinámicos del robot 3-PRS obtenidos por MV.....	78

V-NOTACIÓN

VI-Nomenclatura

t	Tiempo
q	Vector de Coordenadas Generalizadas
\dot{q}	Vector de Velocidades Generalizadas
\ddot{q}	Vector de Aceleraciones Generalizadas
q_m	Vector de Coordenadas Generalizadas Medidas
q_{ind}	Vector de Coordenadas Generalizadas Independientes
q_{dep}	Vector de Coordenadas Generalizadas Dependientes
q_v	Vector de Coordenadas Generalizadas Virtuales
q_{ext}	Vector de Coordenadas Generalizadas y Virtuales
z	Vector de Coordenadas Independientes
\dot{z}	Vector de Velocidades Independientes
\ddot{z}	Vector de Aceleraciones Independientes
$V(P)$	Vector Velocidad Lineal del punto P
$a(P)$	Vector Aceleración Lineal del punto P
$\Omega(Sol)$	Vector Velocidad angular del Sólido Sol
$\alpha(Sol)$	Vector Aceleración Angular del Sólido Sol
V^*	Vector Velocidad Lineal Virtual
Ω^*	Vector Velocidad Angular Virtual
N_{Acc}	Número de Acciones Constitutivas
N_{Sol}	Número de solidos que componen el sistema
N_H	Número de harmónicos de la serie finita de Fourier
N_{iter}	Número de iteraciones del proceso iterativo
N_m	Número de mediciones realizadas
N_{per}	Número de periodos de la trayectoria completa
P_{Virt}	Potencia Virtual del sistema
F	Vector Fuerzas
\mathcal{F}	Vector Fuerza de Inercia de <i>D'Alembert</i>
M	Vector Momentos
\mathcal{M}	Vector Momento de Inercia de <i>D'Alembert</i>
Γ_k	Ecuación Dinámica k -ésima
RI	Referencia Inercial

Sol	Sólido
Sue	Suelo
g	Vector Aceleración de la Gravedad
m	Masa de un sólido
m_x	Primer Momento de Inercia en la Dirección x
I_{xx}	Segundo Momento de Inercia en la Dirección xx
I_{xy}	Producto de Inercia en la Dirección xy
$I_P(Sol)$	Tensor de Inercia del Sólido Sol definido en un punto P
\overline{PQ}	Vector que va desde el punto P al punto Q
Ψ	Vector de Ecuaciones de Restricción
K	Matriz de Observación para una sola configuración
K^m	Matriz de Observación de las Ecuaciones Dinámicas de Movimiento para una sola configuración
K^ε	Matriz de Observación de las Ecuaciones Dinámicas de Enlace para una sola configuración
W	Matriz de Observación del Sistema
W_b	Matriz de Observación del Sistema Base
r	Número de filas de la matriz W
c	Número de columnas de la matriz W
b	Número de parámetros base
ρ	Error de estimación
Σ_ρ	Matriz de Varianzas y Covarianzas del Error de Estimación de Fuerzas
σ_i	Valor Singular i -ésimo
μ	Media de la Distribución Normal
\dot{W}	Potencia Virtual de las Fuerzas Externas
$\dot{\mathcal{W}}$	Potencia Virtual de las Fuerzas de Inercia
ϕ	Vector de Parámetros Dinámicos
ϕ_j	j -ésimo Parámetro Dinámico
ϕ_b	Vector de Parámetros Base
ϕ_i	Vector de Parámetros Inerciales
ϕ_v	Vector de Parámetros de Fricción Viscosa
ϕ_C	Vector de Parámetros de Fricción de Coulomb

ϕ_{LS}	Vector de Parámetros Dinámicos estimados mediante <i>MC</i>
ϕ_{ML}	Vector de Parámetros Dinámicos estimados mediante <i>MV</i>
<i>TOL</i>	Tolerancia predefinida por <i>Matlab</i> [®]
τ	Vector de Fuerzas Externas
τ_ε	Vector de Fuerzas Enlace Generalizadas
τ_m	Vector de Fuerzas Externas Medidas
ε	Vector de Incógnitas de Enlace
λ	Vector de Parámetros Geométricos
ξ	Vector Multiplicadores de Lagrange
<i>A</i>	Matriz Jacobiana de las ecuaciones de Restricción
n_q	Ruido en la Medición de las Coordenadas <i>q</i>
n_τ	Ruido en la Medición de τ
δ	Vector de Parámetros de Trayectoria
δ_{LS}	Vector de Parámetros de Trayectoria estimados mediante <i>MC</i>
δ_{ML}	Vector de Parámetros de Trayectoria estimados mediante <i>MV</i>
ϑ	Vector de Parámetros de Trayectoria y Parámetros Dinámicos
ϑ_{ML}	Vector de Parámetros de Trayectoria y Parámetros Dinámicos estimados mediante <i>MV</i>
\mathcal{F}_C	Función de Criterio de Optimización <i>C</i>
ω	Pulsación Fundamental
T_f	Periodo Fundamental
a_{i0}, a_{ij}, b_{ij}	Coefficientes de la serie de Fourier, correspondientes a la <i>i</i> -ésima coordenada independiente y <i>j</i> -ésimo armónico

VII-Operadores

θ^T	Transpuesta del Vector θ
$\dot{\theta}$	Derivada Temporal de θ
$\partial\theta/\partial t$	Derivada Parcial de θ respecto de t
$\bar{\theta}$	Media de θ
σ_{θ}^2	Varianza de θ
σ_{θ}	Desviación típica estándar de θ
$\hat{\theta}$	Estimación del Vector θ
$\{\theta\}$	Coordenadas del Vector θ en un base dada
θ_i	Elemento i del Vector θ
β_{ij}	Elemento (i, j) de la matriz β
$\{\beta\}$	Coordenadas del tensor β en una base dada

VIII-Funciones

$\kappa(M)$	Número de Condición de la Matriz M
$\det(M)$	Determinante de la Matriz M
$\max(f)$	Máximo de la Función f
$\min(f)$	Mínimo de la Función f
$\arg(f)$	Argumento de la Función f
$ n $	Valor Absoluto de n
$\ \theta\ $	Norma del Vector θ
$E[\theta]$	Esperanza Matemática del Vector θ
$\text{rango}(M)$	Rango de la Matriz M
$\text{sign}(n)$	Signo de n

IX-Acrónimos

<i>CAD</i>	Computer Aided Desing
<i>CG</i>	Centro de Gravedad
<i>GDL</i>	Grados de Libertad
<i>DVS</i>	Descomposición por Valores Singulares
<i>LSE</i>	Least Squares Estimation
<i>MLE</i>	Maximum Likelihood Estimation
<i>MV</i>	Máxima Verosimilitud
<i>MC</i>	Mínimos Cuadrados
<i>NC</i>	Número de Condición
<i>PT</i>	Punto Terminal
<i>PPV</i>	Principio de las Potencias Virtuales
<i>RIA</i>	Robotic Industry Association
<i>TML</i>	Teorema del Momento Lineal
<i>TMA</i>	Teorema del Momento Angular
\mathcal{P}	Par Prismático
\mathcal{R}	Par de Revolución
\mathcal{S}	Par Esférico
<i>SD</i>	Steepest Decent method
<i>SVD</i>	Singular Values Descomposition

1. INTRODUCCIÓN

1.1. Conceptos generales

En el presente Trabajo de Fin de Grado se presentan los conceptos de la teoría de *Identificación de Sistemas Dinámicos*, siendo esta la disciplina encargada de la realización de modelos matemáticos de sistemas, partiendo del análisis de datos experimentales, mediciones y observaciones del sistema. Su base teórica se sustenta en parte por la teoría de sistemas dinámicos, como así también por métodos y algoritmos matemáticos para la correcta estimación de los parámetros involucrados. Cada rutina de identificación presentada se encuentra complementada con su correspondiente aplicación en *Matlab*[®].

Debido a que los sistemas dinámicos abundan en nuestro medio ambiente, las técnicas de identificación de sistemas han cobrado gran relevancia en diversas áreas del conocimiento (ingenierías, economía, biotecnología, etc.), donde se requiere un modelo preciso para fines de análisis, predicción, simulación, diseño y control.

En particular, las técnicas de control actuales requieren de modelos matemáticos cada vez más precisos en lo que se refiere a el análisis y el diseño. En muchos casos, tales modelos no pueden ser obtenidos de forma sencilla y económica a partir de las leyes que rigen cada proceso. Es aquí entonces donde juega un papel decisivo la *Identificación de Sistemas Dinámicos*, que es una herramienta capaz de proporcionar los métodos de aproximación necesarios para obtener de manera relativamente sencilla los modelos matemáticos buscados con un alto grado de exactitud.

1.2. Alcance del trabajo

En el trabajo se muestran y se emplean diferentes métodos para la estimación los parámetros dinámicos que caracterizan a cualquier mecanismo, en concreto, se desarrolla más detalladamente el método de *Estimación por Máxima Verosimilitud (MV)*, o (*MLE, Maximum Likelihood Estimation*).

En primer lugar, se presentará la metodología a seguir para la identificación de los parámetros dinámicos de cualquier robot paralelo, abordando temas como, la modelización del robot y reducción de este. En segundo lugar, se realizará el modelado dinámico de un robot 3-PRS explícitamente, a continuación, se aplicarán métodos de reducción de este que a través de las diferentes mediciones realizadas en el experimento se aplicará el método de estimación por *MV* y se obtendrán los parámetros dinámicos del mismo.

Finalmente, y a modo de conclusión, se realizan una serie de conclusiones que concluirán si el modelo dinámico propuesto se ajusta con la máxima precisión posible a la realidad y se propondrán algunas líneas de futuro trabajo.

1.3. Antecedentes

El interés por el estudio de los principios de teoría de máquinas y mecanismos en asignaturas como *Mecánica* (4° semestre del plan de estudios) y *Teoría de Máquinas* (6° semestre del plan de estudios) y la afición a la programación mediante programas informáticos como *Matlab*[®] y *3d_mec* son los principales impulsores de este trabajo. Además, el hecho de la concesión de la

Beca de Colaboración otorgada por la *Universidad Pública de Navarra* es otro gran impulsor para la realización de este trabajo.

1.4. Objetivos

El principal objetivo del presente trabajo es presentar el desarrollo de una estrategia metodológica para la determinación del modelo dinámico adecuado para la identificación de los parámetros dinámicos de un robot 3-PRS y posteriormente, mediante el método de estimación por *Máxima Verosimilitud* estimar los parámetros del mismo. El desarrollo del modelo dinámico se basará en las ecuaciones dinámicas del movimiento de sólidos rígidos utilizando la formulación regida por el *Principio de Potencias Virtuales*. Entre los objetivos específicos cabe destacar:

- ◆ Desarrollar la formulación del problema cinemático y dinámico de un robot 3-PRS.
- ◆ Realizar la reducción del modelo dinámico para así obtener una nueva matriz de observación de rango máximo.
- ◆ Proponer una parametrización mediante series finitas de Fourier de la trayectoria a recorrer por cada uno de los actuadores de un robot 3-PRS.
- ◆ Determinar mediante el método de estimación por *MV* los parámetros dinámicos y, además, determinar si el método es adecuado y eficiente para la identificación de los parámetros.
- ◆ Finalmente, validar el modelo dinámico propuesto inicialmente, tras el análisis de los resultados obtenidos.

1.5. Estado del arte

La identificación de sistemas consiste en construir modelos matemáticos de sistemas dinámicos basados en las entradas y salidas observadas, a través de métodos estadísticos. Aunque gran parte del desarrollo de las técnicas está ligado a la *Comunidad de Control*, está básicamente construida a partir de técnicas estadísticas, en particular en los métodos de regresión lineal y no-lineal.

Construir modelos para sistemas desconocidos es un propósito importante de la Ingeniería de control. Estos modelos necesitan simular el comportamiento real en los casos en que existe un conocimiento previo limitado de la estructura del sistema.

Si se considera el motor de un coche, por ejemplo. Es importante simular el comportamiento de un motor para la detección de fallos y su posterior diagnóstico. Algunos factores importantes que afectan este proceso son las dificultades asociadas con el desarrollo de un modelo aceptable con un orden de complejidad mínimo y con un número de medidas mínimo. Estos factores hacen que el proceso de modelado de un motor de coche sea bastante difícil. Esto puede ser generalizado a una amplia clase de sistemas en la industria.

La identificación de sistemas no-lineales se considera un problema de gran dificultad. La razón reside en que identificar un sistema no-lineal conlleva dos grandes etapas: En primer lugar,

la selección de la estructura del modelo con un cierto número de parámetros y, en segundo lugar, la selección de un algoritmo eficiente que estime dichos parámetros.

Aunque muchos sistemas no-lineales pueden ser linealizados representándolos por medio de una ecuación diferencial o en diferencias, modelar sistemas no lineales con modelos lineales implica muchas aproximaciones. Estas aproximaciones no reflejan suficientemente el comportamiento real del sistema no-lineal. Por tanto, el obtener un buen modelo, con una estructura que refleje la información real del sistema, exige un incremento del coste. Este coste es debido a la necesidad de algoritmos más avanzados que puedan manejar modelos con estructuras más complejas.

Las fases de las que consta el procedimiento de identificación de un sistema robótico serían las siguientes:

- ◆ **Modelado dinámico** del robot.
- ◆ **Reducción del modelo dinámico** a parámetros base. A poder ser construyendo un modelo lineal respecto a los parámetros dinámicos, sin realizar simplificaciones.
- ◆ **Diseño de experimentos.** Optimizando en la mayor medida la manera en la que se realizarán los experimentos para que el modelo identificado tenga la máxima calidad y precisión posible.
- ◆ **Estimación de parámetros.** Obteniendo el valor estimado de los parámetros dinámicos y la incertidumbre con la que se han obtenido estos.
- ◆ **Validación del modelo.** Consiste en determinar si el modelo obtenido satisface el grado de exactitud requerido para la aplicación en cuestión. Si se llega a la conclusión de que el modelo no es válido, se deben revisar los siguientes aspectos como posibles causas:
 - El conjunto de datos de entrada-salida no proporciona suficiente información sobre la dinámica del sistema.
 - La estructura escogida no es capaz de proporcionar una buena descripción del modelo.
 - El criterio de ajuste de parámetros seleccionado no es el más adecuado.

En lo que al estado del arte se refiere, en los subapartados posteriores se verá una breve síntesis de cuál es el estado del arte de las diferentes fases de identificación de robots.

1.5.1. Modelado de sistemas mecánicos

El modelado de sistemas mecánicos es un campo que se ha desarrollado desde los años 80, y hoy en día está totalmente consolidada. Existen una amplia selección de libros en los que se describen tanto el modelado cinemático como el modela dinámico de sistemas mecánicos, como las Referencias [5,24], entro otras.

En lo que a la identificación de robots se refiere, a través de los primeros trabajos de identificación se consiguió construir modelos dinámicos de robots que pudieran ser expresados de forma lineal con respecto a los parámetros dinámicos. *Kanade* y *Khosla* en el año 1985 [26] escribieron las ecuaciones para robots de cadena abierta en su forma lineal respecto a los

parámetros dinámicos empleado la formulación de *Newton-Euler*. Más adelante, *Sheu et al.* en 1991 [41] y *Grotjahn et al.* en 2004 [20] escribieron modelos lineales en los parámetros dinámicos mediante la formulación de *Multiplicadores de Lagrange y Principio de las Potencias Virtuales*.

La mayor parte del trabajo realizado entorno a la modelización de robot se ha dedicado a la de métodos computacionales eficientes, tanto en lo que a la cinemática y dinámica se refiere. Sin embargo, la obtención de modelos eficientes no forma parte de este trabajo.

Cabe mencionar el trabajo de *Raucent et al.* en 1992 [35] en el que se construye un modelo reducido, del cual los parámetros inerciales se estiman utilizando mediciones de fuerzas y momentos de enlace entre el primer elemento móvil del robot, y la base fija al suelo. Este hace que la identificación de parámetros inerciales se pueda realizar sin que en las ecuaciones aparezcan términos relacionados con los modelos de fricción.

Construyendo un modelo dinámico para el sistema robótico, y evitando el uso de modelos de fricción no lineales en los parámetros a estimar, se tiene el siguiente sistema de ecuaciones, que resulta ser lineal en los parámetros dinámicos:

$$K(q, \dot{q}, \ddot{q})\phi = \tau \quad (1)$$

donde K representa la matriz de observación del sistema robótico que contiene la información del movimiento, (q, \dot{q}, \ddot{q}) corresponden con las coordenadas generalizadas con las que se acota el mecanismo (y sus dos primeras derivadas temporales) y τ representa el vector de fuerzas y momentos externos que actúan sobre el sistema. El vector ϕ , incluye los parámetros dinámicos del sistema.

La matriz que se construye evaluando las ecuaciones dinámicas del movimiento en cada estado diferente, viene denotado por W y se le confiere el nombre de matriz de observación.

$$W = \begin{bmatrix} K(q_1, \dot{q}_1, \ddot{q}_1) \\ K(q_2, \dot{q}_2, \ddot{q}_2) \\ \vdots \\ K(q_n, \dot{q}_n, \ddot{q}_n) \end{bmatrix} \quad (2)$$

donde n representa el número de estados diferentes en los que se evalúa la matriz K .

1.5.2. Reducción de modelos dinámicos

Una vez elaborado el modelo dinámico, el paso siguiente es realizar los experimentos para obtener las mediciones necesarias para la evaluación de la matriz de observación y el vector de fuerzas y momentos externos. El sistema lineal antes planteado en la Ecuación 1 podría resolverse para obtener una estimación de los parámetros dinámicos. Sin embargo, esto no siempre es posible, sólo será posible cuando el rango de la matriz de observación W sea máximo,

cosa que no suele suceder en los modelos de robots. Que el rango no sea máximo puede deberse a dos motivos. El primero, que algunos de los parámetros inerciales no aparezcan en las ecuaciones, y por esta razón, las columnas de la matriz de observación asociadas a dichos parámetros serán nulas. Esto suele suceder cuando los sólidos que están unidos al suelo mediante un par cinemático tienen restringido su movimiento, y por tanto, algunos parámetros inerciales no tienen influencia sobre las fuerzas y momentos que habría que aplicar para mover el sólido en alguna de sus direcciones permitidas. La segunda razón es que algunos parámetros inerciales se agrupan en las ecuaciones dinámicas, de manera que sólo la suma o combinación lineal entre ellos tiene influencia sobre la dinámica del sistema.

Diversos autores han estudiado la manera en la que suplir esta deficiencia de rango, desde dos puntos de vista: simbólico y numérico.

En lo que a métodos de reducción numérico se refiere, los primeros trabajos que hacen referencia a la manera de agrupar parámetros de forma numérica son de *Atkenson, An y Hollerbach* [8]. En estos trabajos propusieron la clasificación de los parámetros en *identificables, identificables en combinaciones lineales y no identificables*. Se basan fundamentalmente en la *Descomposición por Valores Singulares (SVD)* para obtener una solución para el sistema $W\phi = \tau$, y de esta forma es posible eliminar los parámetros no identificables.

Algo más tarde, en 1991, *Gautier* [16] desarrolló dos métodos para la determinación de las expresiones numéricas de los parámetros, basados en las descomposiciones *SVD* y *QR*. Estos métodos, además, son capaces de deducir combinaciones lineales entre parámetros de fricción.

También, en 1991, *Ghodoussi y Nakamura* [18] son capaces de definir el concepto de *Parámetro Base Principal* y razonan que las expresiones de los parámetros base son independientes de los pares en que se actúa el mecanismo.

Se puede decir que estos métodos son generales y pueden aplicarse tanto a robots de cadena abierta como a robots de cadena cerrada.

Por otro lado, a lo que métodos de reducción simbólicos se refiere, *Gautier y Khalil* desarrollaron un método basado en las Ecuaciones de Lagrange para robots seriales [17]. Se resume en la aplicación de una serie de reglas para la agrupación de parámetros dependiendo si el par cinemático entre dos sólidos es de revolución o prismático.

Mayeda et al. [27] en 1990 desarrollaron un método parecido para la determinación de las agrupaciones de parámetros en robots serie. Éstos, dieron un número de parámetros base identificables en función del número de sólidos y pares de tipo prismático y de revolución, proponiendo así el término *Parámetro Base*.

En el año 1995, *Khalil y Bennis* [23] desarrollaron otro método mediante el cual se podrían obtener las expresiones simbólicas de los parámetros base de robots de cadena cerrada.

Años más tarde, en 2002, *Chen et al.* [13] desarrollaron un método simbólico nuevo para la determinación de las expresiones de los parámetros base de mecanismos planos, sustentado en el empleo del concepto de transferencia de masa e inercia. Ese mismo año, *Chen y Beale* [12]

intentaron de transformar el método para aplicarlo a mecanismos espaciales, pero no consiguieron generalizar el concepto de transferencia de masa e inercia.

1.5.3. Diseño de experimentos

Una vez realizadas las fases de construcción del modelo dinámico y reducción de éste a parámetros base, se realiza la estimación de los parámetros base en función de los datos medidos en los experimentos. El siguiente paso es la evaluación de la Ecuación 1 para un conjunto de estados del sistema. Para ello, será necesario que el sistema se mueva siguiendo una trayectoria, y medir, en cada instante, la posición, velocidad, y aceleración de sus coordenadas independientes, y los esfuerzos externos realizados para mover el sistema según la trayectoria comandada. La fase de diseño de los experimentos es fundamental para la precisa estimación de los parámetros del sistema.

Swevers et al. [43,44] trata sobre el diseño de los experimentos los aspectos relacionados con la elección del tiempo de duración del experimento y la frecuencia de muestreo, entre otros.

Es de gran importancia a la hora de obtener resultados precisos, la trayectoria que realiza el robot en el momento de realizar el experimento. Cuando la *Trayectoria* impuesta al robot se dice que es pobre, pues el movimiento es lento y/o no se aprovecha toda la capacidad de movimiento del robot, al evaluar la matriz de observación del sistema de ecuaciones a resolver (Ecuación (1)), con medidas obtenidas en el experimento, la matriz de observación resulta estar mal acondicionada, y por tanto los resultados que se obtengan serán erróneos o carecerán de precisión. Es por esto que, una de las tareas a realizar a la hora de diseñar los experimentos, es la de diseñar trayectorias que sean óptimas, de manera que los modelos que se deriven de los parámetros identificados sean lo más precisos posibles. Este trabajo se conoce como *Optimización de Trayectorias*, y a continuación se describe el estado del arte.

El primer trabajo acerca de la optimización de trayectorias lo realizó *Armstrong* en 1898 [7] y utilizó por primera vez el término *Permanentemente Excitantes* para hacer referencia a trayectorias con las que se pueden estimar los parámetros dinámicos con cierta precisión. Además, *Armstrong* realiza la optimización de las trayectorias utilizando el criterio de optimización de *Mínimo Valor Singular* de la matriz de observación, el cual se desea maximizar para minimizar el error de estimación de la salida del modelo.

Más adelante, *Gautier* y *Khalil* desarrollaron un nuevo criterio de optimización, basado en el Número de Condición (NC) de la matriz de observación ($\kappa(W)$). El criterio trata en la ponderación del número de condición, y un factor relacionado con el tamaño de los elementos de la matriz de observación, es decir, la expresión del criterio de optimización queda:

$$\mathcal{F}_{\kappa}(W) = \lambda_1 \kappa(W) + \lambda_2 \frac{|W_{ij}|_{\max}}{|W_{ij}|_{\min}} \quad (3)$$

donde λ_1 y λ_2 resultan ser los dos coeficientes de ponderación, y $|W_{ij}|_{m\acute{a}x}$ y $|W_{ij}|_{m\acute{i}n}$ corresponden con los elementos de la matriz W con mayor y menor valores absolutos, respectivamente.

Unos años más tarde, *Presse y Gautier* [34] en 1993 propusieron un nuevo criterio de optimización para las trayectorias. El criterio, se construía como una suma ponderada del *Número de Condición* y el inverso del *Mínimo Valor Singular* de la matriz de observación. Por una parte, minimiza la transmisión de los errores de medición en fuerzas a la estimación de los parámetros base, y por otra parte minimiza la estimación de la desviación típica con la que se estimarán los parámetros. Por tanto, el criterio puede escribirse de la manera:

$$\mathcal{F}_{\kappa\sigma}(W) = \lambda_1 \kappa(W) + \lambda_2 \frac{1}{\sigma_b(W)} \quad (4)$$

donde $\sigma_b(W)$ es el menor valor singular (distinto de cero) de la matriz W .

Hasta ese año, la definición de la trayectoria se realizaba mediante puntos por que debiera pasar el elemento terminal del robot. En entonces, en 1996 cuando *Swevers et al.* [43] definieron trayectorias continuas mediante un conjunto finito de parámetros. La función continua escogida fue una *Serie Finita de Fourier*, por eso, las fusiones resultantes son directamente periódicas e infinitamente derivables. Son muchas las ventajas que supone el uso de series finitas de Fourier para la definición de las trayectorias a optimizar. El criterio que *Swevers* emplea en este trabajo se fundamenta en el número de condición de la matriz de observación.

En su siguiente trabajo, *Swevers et al.* [44] realizaron un estudio tanto de la optimización de trayectorias como de la estimación óptima. Introdujeron el criterio de *Máxima Verosimilitud* en el proceso de estimación de parámetros, el cual tiene buenas propiedades tanto en la estimación como en la optimización de trayectorias. El nuevo criterio que introdujeron, está relacionado con la *Matriz de Información de Fisher* [22] que no es sino la inversa de la matriz de covarianzas de las estimaciones de los parámetros dinámicos (Σ_ρ). Así, mediante el criterio de máxima verosimilitud, se obtuvieron trayectorias que daban estimaciones de los parámetros con menor desviación típica. Por tanto, el criterio puede escribirse de la siguiente manera:

$$\mathcal{F}_{MV}(W) = -\log(\det(W^T \Sigma_\rho^{-1} W)) \quad (5)$$

Tras el trabajo de *Swevers et al.*, no ha habido grandes avances en el área de optimización de trayectorias. *Calafiore et al.* [11] emplearon en 2001 una parametrización semejante a la que utiliza series finitas de Fourier, per utilizando únicamente los términos seno y dejando libres las frecuencias de los armónicos implicados en las sumas finitas.

Otra modificación de la parametrización de trayectorias fue la propuesta por *Park* [32] en 2006. En ella, proponía la modificación de la parametrización por series finitas de Fourier, a una

en la que a las series se les sumase un polinomio de grado 5. De este modo, se posibilita la restricción de las condiciones iniciales y finales de posición, velocidad y aceleración.

Hasta el año 2005, la mayoría de los trabajos de optimización de trayectorias se habían realizado sobre robots serie, en los que el elemento terminal tiene gran movilidad. Así, en 2005 [2,3] y en 2009 [1], *Abdellatif et al.* aplicaron la optimización de trayectorias para la identificación del robot paralelo de 6 grados de libertad *PaLiDa*, y utilizaron el criterio de máxima verosimilitud, tanto para la optimización de trayectorias como para la estimación de los parámetros dinámicos.

Por último, destacar que en 2008, *Sun y Hollerbach* [42] realizaron un estudio de los diferentes criterios de optimización. El estudio estaba orientado a la elección de un criterio de óptimo para la *Calibración de Robots*. Así los autores de este trabajo concluyeron que el criterio de *Máxima Verosimilitud* es el que menor desviación típica provocará en los parámetros estimados, y que el criterio de *Maximización del Menor Valor Singular* es el que proporcionará modelos con los menores errores de predicción. De este modo, en el presente trabajo se estimarán los parámetros dinámicos haciendo uso del criterio de *Máxima Verosimilitud*.

1.5.4. Estimación

Una vez se ha construido el modelo, realizada la reducción de éste a parámetros base, y optimizadas las trayectorias para la correcta estimación de los parámetros, se realiza el experimento y se miden los valores de las coordenadas y fuerzas actuantes. Posteriormente, se estimarán los parámetros inerciales y de fricción de los que depende el modelo.

La estimación del vector de parámetros en función de los parámetros experimentales, puede realizarse en línea o fuera de línea. En la estimación en línea (*on-line*) se realiza una estimación de los parámetros dinámicos en función de los datos experimentales tomados hasta un cierto momento dado, y se va modificando la estimación según se van recibiendo nuevos datos experimentales. Sin embargo, exige la realización de los cálculos en menos tiempo del que resta para recibir los nuevos datos experimentales, por lo que está condicionado a poder realizar los cálculos en tiempo real. Por otra parte, la estimación fuera de línea (*off-line*) no posee restricciones de tiempo para su cálculo, y resulta sencilla de implementar, ya que desvincula la realización del experimento con el proceso de estimación. Este último proceso es el más utilizado en la estimación de parámetros dinámicos de robots.

Por otro lado, a la hora de estimar el valor de los parámetros dinámicos del sistema, éste puede ser calculado mediante diferentes estimadores. Uno de los más empleados es el que minimiza el cuadrado del error de estimación, es decir:

$$\hat{\phi} = \arg \min \{ \|\tau - \tau_m\|_2 \} \quad (6)$$

donde τ corresponde las fuerzas externas estimadas del modelo, y τ_m representa el vector de fuerzas externas medidas en el experimento. Este método de estimación se denomina *Estimación por Mínimos Cuadrados*.

Cuando la estimación del vector τ se realiza mediante un modelo que es lineal en los parámetros, la resolución del problema de minimización expresado en la Ecuación (6), puede realizarse mediante la *Matriz Pseudoinversa*, de modo que la estimación por mínimos cuadrados para modelos lineales en los parámetros queda de la siguiente manera:

$$\hat{\phi} = (W^T W)^{-1} W^T \tau_m \quad (7)$$

Sin embargo, cuando la estimación del vector de fuerzas, τ , se realiza mediante un modelo *no lineal*, deberán utilizarse métodos de minimización de funciones no lineales para la obtención de la estimación por mínimos cuadrados.

En 1997, Swevers et al. [44] aplicaron por primera vez el *Principio de Máxima Verosimilitud* para la estimación de los parámetros dinámicos de robots seriales. El método de estimación por máxima verosimilitud, selecciona el vector ϕ que tiene la propiedad de maximizar la probabilidad la muestra aleatoria de medidas realizada. Este estimador es insesgado, por lo que aún en presencia de ruido en las mediciones de las coordenadas, la esperanza del estimador es el valor real del parámetro.

Cuando el ruido en la medición de las coordenadas es despreciable, el estimador de máxima verosimilitud puede expresarse a través de la Ecuación 8:

$$\hat{\phi} = (W^T \Sigma_\rho^{-1} W)^{-1} W^T \Sigma_\rho^{-1} \tau_m \quad (8)$$

donde Σ_ρ^{-1} representa la matriz de varianzas y covarianzas del error de estimación ρ , definido mediante el operador *Esperanza Matemática* $E[\cdot]$:

$$\Sigma_\rho = E[\rho \rho^T] \quad (9)$$

El estimador de la Ecuación 8 se interpreta como el estimador por *Mínimos Cuadrados* utilizando la matriz de ponderación que da diferente peso a unas ecuaciones y otras. Por esto, este estimador se le denomina *Estimador por Mínimos Cuadrados Ponderados*.

Los resultados obtenidos por Abdellatif et al. [2], Swevers et al. [44] y Olsen et al. [31], muestran que los métodos de estimación por *Mínimos Cuadrados Ponderados* y por *Máxima Verosimilitud*, proporcionan resultados prácticamente iguales en condiciones experimentales en las que los errores en la medición de las coordenadas de posición son despreciables frente a los errores en la medición de las fuerzas externas. Es por esto, que en el presente trabajo se utilizará la estimación por *Máxima Verosimilitud* como método para la estimación de los parámetros dinámicos.

1.5.5. Validación

La validación del modelo consiste en comprobar que el modelo construido mediante datos experimentales, realmente representa el sistema mecánico a estudio. Para realizarlo es necesario comprobar que el modelo, no sólo se ajusta a los datos experimentales con los que ha sido obtenido, sino que realmente representa al sistema cuando los *inputs* que se introduzcan al modelo sean diferentes a los utilizados en el experimento de identificación. Así, utilizando los datos de un experimento aislado del realizado para la identificación, pueden compararse los *outputs* medidos y estimados por el modelo, y analizar si el modelo representa con la suficiente precisión al sistema.

2. MODELADO DE SISTEMAS MECÁNICOS

2.1. Introducción

En este apartado, se describe la metodología a seguir para el desarrollo del modelo dinámico empleado en la identificación de parámetros de robots paralelos. Al observar el funcionamiento de un sistema mecánico se puede comprobar que se trata de un conjunto mecánico de miembros, generalmente idealizados como sólidos rígidos, que reciben energía de alguna forma y la emplean para conseguir un fin determinado, como puede ser, transmitir potencia o generar movimiento: Por ejemplo, el mecanismo biela-manivela de un motor de explosión transmite potencia al cigüeñal y este al resto de órganos del automóvil, pero la función principal del sistema de distribución es general movimiento.

Luego para el estudio de un sistema mecánico, se hace necesario el estudio de las fuerzas que han de aplicarse para conseguir la finalidad propuesta y el estudio del movimiento de las partes que constituyen dicha máquina. De esta forma, se deberá estudiar las trayectorias recorridas por determinados puntos que pertenecen a diferentes órganos de la máquina, y la forma en que se recorren dichas trayectorias: velocidades y aceleraciones; en definitiva, si el estudio se centra en el movimiento de una máquina (o de un elemento) prescindiendo de su causa se hablará de cinemática de las máquinas. Por el contrario, si lo que interesa es el estudio de las fuerzas, ligado al efecto producido (el movimiento), se abarca la disciplina de la dinámica de máquinas.

Como se ha mencionado, es la dinámica la que incluye el modelado de los sólidos rígidos y las fricciones en los pares o juntas que constituyen el robot. El objetivo es reescribir la ecuación del movimiento de manera que el sistema quede de forma lineal respecto a los parámetros a identificar.

Previo al planteamiento del modelo dinámico, se describe inicialmente el procedimiento para la resolución del problema cinemático de robots paralelos. Tras definir la cinemática, se aborda el problema dinámico. Se modela el robot paralelo considerándolo como un sistema mecánico constituido por varias cadenas abiertas, que se obtienen de separar el robot en uno o varios de sus pares. De esta forma se obtiene el modelo dinámico que incluye la dinámica de las cadenas abiertas, obtenidas por la separación, conjuntamente con las ecuaciones de restricción debido a las separaciones realizadas.

Una vez desarrollas las ecuaciones de dinámica de solido rígido, se presentan los modelos de fricción que han sido empleados para la identificación de parámetros. Los modelos de fricción empleados no se limitan exclusivamente a modelos lineales, sino que además se presentan modelos no lineales de fricción. Finalmente, resaltar que en este trabajo no se considera el modelado de los elementos activos, como pueden ser sistemas de actuación, tornillos de potencia y motores.

Es de destacar que cuando el modelo dinámico es lineal, con respecto a los parámetros a identificar, no todos los parámetros influyen en el comportamiento dinámico del sistema robótico y algunos de ellos se puede agrupar hasta formar combinaciones lineales. Es por esto

que al final del apartado se presenta la obtención del modelo dinámico en parámetros mínimos o “*Parámetros Base*”.

2.2. Modelado cinemático

El análisis cinemático tiene como objetivo el determinar el movimiento (posición, velocidad, y aceleración) de cada uno de los elementos que conforman el sistema mecánico, en función de las coordenadas generalizadas empleadas para el modelado del sistema. Es por esto, que es necesario definir las coordenadas generalizadas que se considerarán en la modelización del robot. Los sólidos rígidos que componen el sistema mecánico en el espacio tienen un movimiento general de 6 grados de libertad, por tanto, su posición y orientación se podrán fijar mediante 6 coordenadas generalizadas. Sin embargo, la presencia de pares cinemáticos entre los sólidos rígidos, reduce la movilidad del sistema, lo que hace necesario del uso de las ecuaciones de enlace cinemático que describen las relaciones entre coordenadas generalizadas.

2.2.1. Sistema de acotación

Un método de acotación muy empleado en el área de la robótica es la denominada *Notación de Denavit-Hartenberg*. Según esta notación, cada sólido se acota respecto del sólido anterior en la cadena cinemática mediante de 4 parámetros. Cada conjunto de parámetros de *Denavit-Hartenberg* representa un par cinemático de revolución (\mathcal{R}) o prismático (\mathcal{P}), dependiendo de qué parámetro se tome como *Coordenada Generalizada* y cuales sean sus *Parámetros Geométricos*. Cuando dos sólidos están unidos por un par cinemático que no es \mathcal{R} ni \mathcal{P} , se puede seguir utilizando esta notación introduciendo sólidos intermedios sin masa y describiendo el movimiento relativo entre los sólidos en una combinación equivalente de pares \mathcal{R} y \mathcal{P} .

En el campo de la identificación de robots y la obtención de los parámetros base, la gran mayoría de los autores utilizan la notación de *Denavit-Hartenberg Modificada* para la construcción de modelos geométricos. Pero en este trabajo, se va emplear una acotación de estilo libre de los sólidos, sin hacer uso de una notación específica en concreto.

Como se ha mencionado, la acotación que se va realizar es de estilo libre, por eso, se realizará mediante vectores de posición y cambios de base. La acotación de un sólido en el espacio requiere de la determinación de 6 coordenadas que represente su posición y orientación. Es por esto que, la acotación de un sólido se puede realizar mediante el vector posición de uno de sus puntos (r) y a través de tres giros respecto de una orientación conocida $(\gamma_x, \gamma_y, \gamma_z)$.

De este modo, mediante 6 coordenadas $(r_x, r_y, r_z, \gamma_x, \gamma_y, \gamma_z)$ se puede determinar la posición y orientación de cada sólido respecto de otro, y así, acotar cada uno de los sólidos que componen el sistema mecánico.

Una vez se han acotado todos los sólidos del sistema, existe la posibilidad de dividir las coordenadas $(r_x, r_y, r_z, \gamma_x, \gamma_y, \gamma_z)$ en dos grupos, según sus características. Las coordenadas que represente una variable, y por tanto, una capacidad de movimiento relativo entre dos sólidos se le denomina *Coordenada Generalizada* del mecanismo, y se incluye en el vector con el mismo nombre denotado por q . Por el contrario, las coordenadas que representan una longitud o una

orientación fija de un sólido respecto de otro, no representarán coordenadas sino *Parámetros Geométricos*. Así pues, estos parámetros se agrupan en el vector λ .

2.2.2. Ecuaciones cinemáticas

La presencia de enlaces cinemáticos entre los sólidos del sistema mecánico puede implicar la existencia de ecuaciones cinemáticas entre las coordenadas generalizadas con las que se ha acotado el mecanismo. Si se considera un mecanismo actuado con n coordenadas generalizadas, de las cuales f son independientes que se tomarán como grados de libertad, existirán $m = n - f$ ecuaciones cinemáticas independientes entre las coordenadas. Así, todas las coordenadas quedarán determinadas en función de las f coordenadas independientes.

Al conjunto de m ecuaciones que relacionan las coordenadas generalizadas entre sí, se les denomina *Ecuaciones de Restricción* y en este trabajo se las denotará mediante la ecuación vectorial siguiente:

$$\Psi(t, q) = 0 \quad (10)$$

2.2.3. Resolución del problema de posición

El problema de posición de un sistema mecánico consiste en determinar el valor de las coordenadas dependientes, q_{dep} , en función del valor de las coordenadas independientes, q_{ind} . El vector de coordenadas generalizadas incluye las dos anteriores, de modo que $q = (q_{dep}^T, q_{ind}^T)^T$. La Ecuación 10, suele no ser lineal respecto a las coordenadas generalizadas, y puede resolverse mediante la utilización del método iterativo de *Newton-Raphson* que se expresa a continuación (véase Sección 9.2.10):

$$A_{dep}^{(i)} (q_{dep}^{(i)} - q_{dep}^{(i+1)}) = \Psi(t^{(i)}, q^{(i)}) \quad (11)$$

donde $A_{dep} = \partial\Psi(t, q)/\partial q_{dep}$ es la matriz jacobiana de las ecuaciones de restricción respecto de las coordenadas generalizadas dependientes, y el subíndice i hace referencia a la evaluación de la iteración i -ésima.

2.2.4. Resolución del problema de velocidad

Una vez abordado y resuelto el problema de posición y cuando todas las coordenadas generalizadas tienen valores que hacen cumplir la Ecuación 10, se trata de resolver el problema de velocidad para obtener las velocidades generalizadas dependientes, \dot{q}_{dep} , a partir del valor de las velocidades generalizadas independientes, \dot{q}_{ind} . Este problema resulta ser lineal, a diferencia del anterior, y consiste en resolver la derivada temporal de las ecuaciones de posición (Ecuación 11). Por tanto, el problema de velocidad a resolver es (véase Sección 9.2.11):

$$A_{dep}\dot{q}_{dep} = -A_{ind}\dot{q}_{ind} \quad (12)$$

donde en este caso, $A_{ind} = \partial\Psi(q)/\partial q_{ind}$ es la matriz jacobiana de las ecuaciones de restricción respecto de las coordenadas generalizadas independientes.

2.2.5. Resolución del problema de aceleración

Y finalmente, una vez resueltos los problemas de posición y velocidad se plantea el problema de aceleraciones para obtener las aceleraciones de las coordenadas generalizadas dependientes, \ddot{q}_{dep} , en función de las aceleraciones generalizadas independientes, \ddot{q}_{ind} . Este problema se aborda obteniendo la resolución de la derivada temporal de las ecuaciones de velocidad (Ecuación 12), y al igual que el problema de velocidad resulta ser lineal. Por tanto, el problema de aceleraciones resulta ser (véase Sección 9.2.12):

$$A_{dep}\ddot{q}_{dep} = -\dot{A}_{dep}\dot{q}_{dep} - \dot{A}_{ind}\dot{q}_{ind} - A_{ind}\ddot{q}_{ind} \quad (13)$$

2.3. Modelado dinámico

El modelo dinámico está constituido por los aspectos de un sistema relacionados con el tiempo y con los cambios en los objetos y sus relaciones a lo largo del tiempo. Frecuentemente los problemas de dinámica de mecanismos se resuelven mediante el método de *Newton-Euler*. Sin embargo, en este trabajo de fin de grado se hace uso del *Principio de las Potencias Virtuales Sistematizado* [39] como método de construcción de las ecuaciones dinámicas del sistema.

Se ha elegido el uso de este método debido a la facilidad para obtener las ecuaciones dinámicas, y además porque se obtienen un menor número de ecuaciones-incógnitas que empleando el método ordinario de *Newton-Euler*.

El *Principio de las Potencias Virtuales* nos permite obtener las ecuaciones dinámicas del sistema como un sumatorio de productos de torsores por velocidades virtuales. Más adelante se caracterizarán todos estos torsores y velocidades virtuales que darán lugar a las ecuaciones dinámicas. Y se analizará de qué manera hay que definir los torsores para que las ecuaciones dinámicas sean lo más simple posibles y lineales respecto de los parámetros dinámicos.

Además, se mostrará la forma de obtención de las *Ecuaciones de Movimiento* y las *Ecuaciones de Enlace* que permitirán el cálculo de las aceleraciones generalizadas y las fuerzas de enlace, respectivamente.

2.3.1. Parámetros dinámicos

El propósito principal del modelo dinámico es el de identificar el propio sistema que representa. De este modo, una vez obtenida la estructura del modelo dinámico, se intentará estimar una serie de parámetros, denominados parámetros dinámicos, que serán desconocidos *a priori*, de manera que el modelo identificado sea capaz de predecir, con la mayor precisión posible, el comportamiento dinámico del sistema estudiado.

Los parámetros que deberemos estimar serán aquellos relacionados con las inercias de los sólidos que compongan el sistema (*Parámetros Inerciales*) y los referidos a cualquier parámetro de fricción que se emplee (*Parámetros de Fricción*).

Los parámetros inerciales comprenden simplemente la masa de cada sólido, la posición de su CG en el espacio y las 6 componentes diferentes del tensor de inercia. A la hora de escribir las ecuaciones dinámicas, se tendrá en cuenta que, para obtener un modelo lineal respecto a los parámetros inerciales, éstos deberán cumplir 2 condiciones:

- ◆ La información de la posición del centro de gravedad habrá que estimarla a partir de los 3 primeros momentos de inercia del sólido, dados por el producto de la masa por las tres coordenadas de la posición del centro de gravedad. Una vez estimada la masa, podrá calcularse la posición del centro de gravedad dividiendo estos parámetros por la masa estimada.
- ◆ La referencia en la que se defina el tensor de inercia del sólido, deberá definirse en un punto cinemáticamente conocido del sólido. Por tanto, el único punto en el que no podremos definir la referencia será el centro de gravedad, dado que su posición es uno de los parámetros a estimar, y es desconocido a priori.

De modo que, los parámetros inerciales de cada sólido que habrán de estimarse serán los siguientes:

$$(m, m_x, m_y, m_z, I_{xx}, I_{xy}, I_{xz}, I_{yy}, I_{yz}, I_{zz}) \quad (14)$$

donde m corresponde con la masa de cada sólido, (m_x, m_y, m_z) representa el producto de la masa por la posición del centro de gravedad en la referencia solidaria a cada sólido, e I_{ij} definen los momentos de inercia definidos en esa misma referencia. Al vector que agrupa todos los parámetros inerciales lo llamaremos *Vector de Parámetros Inerciales*, y se denotará por ϕ_i .

Se considerarán los modelos de fricción del tipo *Fricción Viscosa* y *Fricción Constante de Coulomb* y se denotarán por ϕ_v y ϕ_c , respectivamente.

El vector que incluya tanto los parámetros inerciales de todos los sólidos como todos los parámetros de fricción, se denominará *Vector de Parámetros Dinámicos* y se denotará por ϕ .

2.3.2. Principio de las Potencias Virtuales

El método que se emplea para la obtención de las ecuaciones dinámicas del robot 3-PRS es el *Principio de las Potencias Virtuales Sistematizado (PPV)*. Este, permite plantear el análisis dinámico de un mecanismo de manera selectiva, es decir, sin tener que plantear un número elevado de ecuaciones dinámicas para aislar posteriormente lo que interesa.

Las ecuaciones dinámicas de *Newton-Euler*, a través del *Principio de Potencias Virtuales* se pueden expresar de la siguiente manera:

$$\Gamma_k = \sum_{i=1}^{N_{Sol}} (\mathcal{F}_i \cdot V_{ik}^* + \mathcal{M}_i \cdot \Omega_{ik}^*) + \sum_{i=1}^{N_{Sol}} \sum_{j=1}^{N_{Acc}} (F_{ij} \cdot V_{ik}^* + M_{ij} \cdot \Omega_{ik}^*) = 0 \quad (15)$$

donde F_{ij} y M_{ij} representan las fuerzas y momentos exteriores (acciones constitutivas) j -ésimos aplicados sobre el sólido i , N_{Sol} corresponde con el número de sólidos del sistema mecánico y N_{Acc} el número de acciones que se ejercen sobre cada sólido. Por otro lado, los vectores V_{ik}^* y Ω_{ik}^* , representan los vectores por los que se multiplican las fuerzas ejercidas sobre el sólido i que darán la ecuación dinámica k -ésima, Γ_k (escalar).

Si a los vectores V^* y Ω^* se les denomina *Velocidades Virtuales*, multiplicados por desplazamientos infinitesimales dan lugar a *Potencias Virtuales*, de ahí el nombre del *Teorema o Principio*.

A la hora de generalizar este principio para cualquier sistema de varios sólidos, resulta de gran interés hacer uso del concepto de tórsor. El *Tórsor* es la pareja de *Fuerza-Momento* actuantes sobre un punto concreto de un sólido rígido. Los tórsores son diferentes dependiendo del punto en el que se caracterice el mismo. A su vez, a la pareja V^* y Ω^* se denomina *Rotor*, y dan lugar a que el *Principio de Potencias Virtuales* quede expresado de esta manera reducida:

$$\sum_{i=1}^{N_{Sol}} \dot{\mathcal{W}}_i + \sum_{i=1}^{N_{Sol}} \sum_{j=1}^{N_{Acc}} \dot{W}_{ij} = 0 \quad (16)$$

donde $\dot{\mathcal{W}}_i$ es la potencia virtual del sólido i debida a su tórsor de inercia, y \dot{W}_{ij} es la potencia virtual del sólido i debida al tórsor de fuerzas j aplicado sobre el sólido i .

Los componentes de la Ecuación 16 resultan ser los siguientes:

$$\dot{\mathcal{W}}_i = \mathcal{F}_i \cdot V^* + \mathcal{M}_i \cdot \Omega^* \quad (17)$$

$$\dot{W}_{ij} = F_{ij} \cdot V^* + M_{ij} \cdot \Omega^* \quad (18)$$

2.3.3. Elección de las velocidades virtuales

El concepto de velocidad virtual, además de ser vectores arbitrarios por que se multiplican las ecuaciones dinámicas, pueden interpretarse como vectores velocidad de los puntos de los sólidos en los que se definen los tórsores, y vectores velocidad angular de los propios sólidos. Si se construyesen tantos vectores de velocidad virtual independientes como grados de libertad tiene el mecanismo, de forma que el movimiento de los sólidos pudiera realizarse sin romper ningún enlace, aplicando estos vectores a las ecuaciones dinámicas, se obtendrían las denominadas *Ecuaciones Dinámicas de Movimiento*. En estas ecuaciones, no aparecería ninguna fuerza de enlace que aparecen en los pares cinemáticos. Sin embargo, si las velocidades virtuales definen algún movimiento de los sólidos que es imposible de realizar por el mecanismo, si no fuese rompiendo alguno de sus enlaces, entonces en las ecuaciones resultantes aparecerá el

término de la potencia virtual correspondiente al par cinemático que se ha roto. Este conjunto de ecuaciones se les denominará *Ecuaciones Dinámicas de Enlace*.

A continuación, se verá la obtención de las *Ecuaciones Dinámicas de Movimiento* en función de las coordenadas independiente y dependientes y las *Ecuaciones Dinámicas de Enlace*. Mencionar que la obtención de las *Ecuaciones Dinámicas de Movimiento* en función de las coordinas independientes no se aplicará en este trabajo.

❖ VELOCIDADES VIRTUALES PARA LA OBTENCIÓN DE LAS ECUACIONES DINÁMICAS DE MOVIMIENTO EN COORDENADAS INDEPENDIENTES

Si se supone que las velocidades angulares de las referencias (Ω_i) y las velocidades de los orígenes de éstas (V_{O_i}) se pueden escribir en términos del conjunto de velocidades independientes (\dot{z}), las velocidades virtuales del sólido i a emplear para la ecuación dinámica j -ésima del movimiento ser:

$$V_{ik}^* = \frac{\partial V_{O_i}(\dot{z})}{\partial \dot{z}_k} \quad ; \quad \Omega_{ik}^* = \frac{\partial \Omega_i(\dot{z})}{\partial \dot{z}_k} \quad (19)$$

❖ ECUACIONES DINÁMICAS DE MOVIMIENTO EN COORDENADAS DEPENDIENTES

De lo contrario, las velocidades de los puntos y referencias en los que se han definido los torsos no pueden escribirse de forma explícita en función de las coordenadas independientes, por tanto, es necesario algunas modificaciones en la formulación de las ecuaciones dinámicas del movimiento.

Una de las formas de resolver el problema es el plantear las llamadas *Ecuaciones de Restricción Cinemática*. Si se diesen valores arbitrarios a los componentes del vector de velocidades generalizadas \dot{q} , algunos de los enlaces del mecanismo se romperían. Es por esto, que se definen las fuerzas de enlace para que esto no suceda. Es decir, si se define τ_ε como el vector de *Fuerzas Generalizadas* de enlace, estos pueden escribirse como:

$$\tau_\varepsilon = A^T \cdot \xi \quad (20)$$

donde ξ representan los *Multiplicadores de Lagrange* y A es la matriz jacobiana de las ecuaciones de restricción. El vector τ_ε , además, puede escribirse directamente en términos de las incógnitas de enlace ε . De esta manera, se podrían escribir las fuerzas generalizadas de enlace τ_ε como:

$$\tau_\varepsilon = A^T \cdot \varepsilon \quad (21)$$

Por lo tanto, las ecuaciones dinámicas del movimiento, escribiendo τ_ε en términos de la matriz jacobiana de las ecuaciones de restricción y de los *Multiplicadores de Lagrange*, se tiene la Ecuación 22:

$$\Gamma = \sum_{i=1}^{N_{Sol}} \left(\left(\frac{\partial V_{O_i}(\dot{q})}{\partial \dot{q}} \right)^T \cdot \mathcal{F}_i + \left(\frac{\partial \Omega_i(\dot{q})}{\partial \dot{q}} \right)^T \cdot \mathcal{M}_i \right) + \sum_{i=1}^{N_{Sol}} \sum_{j=1}^{N_{Acc}} \left(\left(\frac{\partial V_{O_i}(\dot{q})}{\partial \dot{q}} \right)^T \cdot F_j + \left(\frac{\partial \Omega_i(\dot{q})}{\partial \dot{q}} \right)^T \cdot M_j \right) + A^T \cdot \varepsilon = 0 \quad (22)$$

Derivando las ecuaciones parcialmente respecto del vector parámetros dinámicos ϕ , y definiendo la traspuesta del jacobiano de las ecuaciones de restricción como $V = A^T$, se obtiene:

$$K^m \cdot \phi + V \cdot \varepsilon = \tau \quad (23)$$

donde $\phi^T = (\phi_i^T, \phi_v^T, \phi_c^T)$ corresponde con el vector de agrupa los parámetros dinámicos inerciales, viscosos y de Coulomb, y τ es el vector de fuerzas y momentos externos. La matriz K^m se calcula derivando parcialmente las ecuaciones respecto de los parámetros dinámicos:

$$K^m = \frac{\partial \Gamma}{\partial \phi} \quad (24)$$

La Ecuación 23 es una ecuación vectorial. Cada una de las ecuaciones escalares de las que está compuesta, se calcula utilizando un conjunto de velocidades virtuales diferentes. Así, para obtener la ecuación k -ésima, se derivan las velocidades (reales) de cada sólido respecto de la velocidad generalizada k -ésima, \dot{q}_k . Del mismo modo, la columna k de la traspuesta de la matriz jacobiana A , es decir, la matriz V , se obtiene derivando cada ecuación de restricción respecto de la coordenada generalizada k -ésima.

Se puede ver como cada ecuación escalar de Γ están relacionadas con una coordenada generalizada. Por tanto, si se dividen las ecuaciones en dos grupos, uno con las ecuaciones relacionadas con las coordenadas generalizadas que se toman como independientes, q_{ind} , (tantas como *GDL*) y el otro con las ecuaciones relacionadas con las coordenadas generalizadas que se toman como dependientes, q_{dep} . Entonces, si los pares acotados mediante las

coordenadas dependientes son pasivos, el vector τ_{dep} , será nulo y podrá eliminarse del sistema de ecuaciones. De manera que se puede describir la Ecuación 23 de la manera siguiente:

$$K_{ind}^m \cdot \phi + V_{ind} \cdot \varepsilon = \tau_{ind} \quad (25)$$

$$K_{dep}^m \cdot \phi + V_{dep} \cdot \varepsilon = \tau_{dep} \quad (26)$$

Despejando ε de la Ecuación 26(26):

$$\varepsilon = -V_{dep}^{-1} \cdot (\tau_{dep} - K_{dep}^m \cdot \phi) \quad (27)$$

y sustituyendo en la Ecuación 25, se obtiene:

$$(K_{ind}^m + X^T \cdot K_{dep}^m) \cdot \phi = \tau_{ind} + X^T \cdot \tau_{dep} \quad (28)$$

donde $X^T = -V_{ind}V_{dep}^{-1}$. Si los pares acotados mediante las coordenadas dependientes son pasivos, el vector τ_{dep} será nulo, y puede eliminarse de la ecuación anterior. Simplificando la notación llamando simplemente τ al vector de fuerzas externas, la Ecuación 29 nos proporcionará las ecuaciones dinámicas de movimiento de un sistema mecánico mediante coordenadas dependientes y ecuaciones de restricción cinemáticas:

$$(K_{ind}^m + X^T \cdot K_{dep}^m) \cdot \phi = \tau \quad (29)$$

Se puede observar como la Ecuación 29 es lineal en ϕ . Sin embargo, como las matrices K_{ind}^m y K_{dep}^m , dependen de las incógnitas de enlace (que a su vez dependen de los parámetros dinámicos) la estimación de los parámetros dinámicos será un problema de naturaleza no lineal.

❖ ECUACIONES DE ENLACE

El uso del método de las potencias virtuales en gran medida se ha empleado pues permite construir las ecuaciones necesarias para calcular cualquier fuerza de enlace, y únicamente esa. De manera que eligiendo de forma correcta las velocidades virtuales de los sólidos, es posible obtener una Ecuación de Enlace para cada fuerza de enlace que se quiera estimar.

Haciendo uso del Principio de las Potencias Virtuales como mecanismo para la obtención de las ecuaciones dinámicas de enlace, sólo será necesario plantear un conjunto de velocidades virtuales que sean incompatibles con los enlaces cuyas fuerzas se desean estimar. Como al romper el enlace la velocidad virtual del punto de aplicación como perteneciente a cada uno de

los sólidos será diferente, la potencia virtual generada por la acción de enlace será no nula, y las incógnitas de enlace aparecerán de forma explícita en la ecuación.

Al plantear movimientos virtuales que rompan los enlaces para obtener las fuerzas de enlace, generalmente las coordenadas generalizadas empleadas para la acotación no serán suficientes para determinar un movimiento arbitrario de los sólidos, y será necesario añadir coordenadas adicionales para acotar esos movimientos virtuales. Estas coordenadas se les denominará como *Coordenadas Generalizadas Virtuales*, y se denotarán por q_v . Así, construyendo un nuevo vector que agrupe las coordenadas generalizadas y las virtuales $q_{ext} = (q^T, q_v^T)^T$ las ecuaciones de enlace podrán obtenerse mediante la siguientes velocidades virtuales:

$$V_{ij}^* = \frac{\partial V_{O_i}(\dot{q}_{ext})}{\partial \dot{q}_{ext j}} \quad ; \quad \Omega_{ij}^* = \frac{\partial \Omega_i(\dot{q}_{ext})}{\partial \dot{q}_{ext j}} \quad (30)$$

donde V_{ij}^* y Ω_{ij}^* son las velocidades lineal y angular del sólido i cuando se plantea una velocidad virtual que rompe un enlace según la coordenada $\dot{q}_{ext j}$.

Por tanto, las *Ecuaciones Dinámicas de Enlace* como resultado de velocidades virtuales incompatibles con los enlaces, teniendo en cuenta que las componentes del vector τ no aparecerán en las ecuaciones dinámicas de enlace, pueden escribirse de la siguiente manera:

$$K^\varepsilon \cdot \phi = U^\varepsilon \cdot \varepsilon \quad (31)$$

2.3.4. Caracterización de los Torsores

Haciendo uso del *Principio las de Potencias Virtuales* para la obtención del modelo dinámico, es necesario escribir todos y cada uno de los torsores que actúan sobre cada solido que conforma el mecanismo, así como las velocidades virtuales que se emplearán en cada caso para obtener cada ecuación dinámica.

❖ TORSORES DE INERCIA

La expresión del torsor de inercia del solido i , definido en un punto P_i solidario a este se define como:

$$\mathcal{F}_i = -m_i \cdot a(P_i) \quad (32)$$

$$\mathcal{M}_i^{P_i} = -I_{P_i}(Sol_i) \cdot \alpha(Sol_i) - m_i \cdot \overline{P_i G_i} \times a(P_i) \quad (33)$$

donde m_i es la masa del sólido i . El término $a(P_i)$ es la aceleración lineal del punto P_i respecto de la referencia inercial (RI) y $\alpha(Sol_i)$ es la aceleración angular del sólido i respecto de la referencia inercial, también. Por otro lado, $P_i G_i$ se corresponde con el vector posición desde el punto P_i al centro de gravedad del sólido i , denotado por G_i . Y finalmente, $I_{P_i}(Sol_i)$ hace referencia al tensor de inercia del sólido i definido en una referencia solidaria a este que tiene como origen el punto P_i . Por consiguiente, sus componentes resultan ser:

$$\{I_P\} = \{I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}\}^T \quad (34)$$

❖ TORSORES DE LAS FUERZAS DE LA GRAVEDAD

La expresión del torsor de fuerza de gravedad sobre el sólido i , definido en un punto P_i solidario a éste, se define como:

$$F_{g,i} = m_i \cdot g \quad (35)$$

$$M_{g,i}^{P_i} = m_i \cdot \overline{P_i G_i} \times g \quad (36)$$

donde g es la aceleración de la gravedad. Fijémonos que si el momento $M_{g,i}^{P_i}$ se caracteriza en el punto centro de gravedad del sólido i , G_i , éste sería nulo.

❖ TORSORES DE LAS FUERZAS Y MOMENTOS EXTERNOS

La expresión del torsor de fuerzas y momentos externos, con la fuerza aplicada en un punto Q_i perteneciente al sólido i , y definido el torsor en un punto P_i solidario al sólido puede escribirse como:

$$F_{e,i} = F_e \quad (37)$$

$$M_{e,i}^{P_i} = \overline{P_i Q_i} \times F_e + M_e \quad (38)$$

donde F_e y M_e son las fuerzas y momentos externos, y $\overline{P_i Q_i}$ es el vector posición que va desde el punto P_i hasta el punto Q_i .

❖ TORSORES DE LAS FUERZAS DE FRICCIÓN VISCOSA

El modelo de fricción viscosa considera un par de fuerzas (o un par de momentos) del tipo acción-reacción entre dos sólidos (i y j) unidos mediante un par cinemático, siendo esta fuerza

de fricción proporcional a la derivada temporal de las coordenadas generalizadas que acotan el movimiento relativo entre los dos sólidos.

La expresión del torsor de fuerza de fricción viscosa definido en un punto P_i del sólido i y aplicado en el punto Q_i se define como:

$$F_{v,i} = -\mu_{k,v}^F \cdot \dot{q}_{k,s} \cdot u_k \quad (39)$$

$$M_{v,i}^{P_i} = -\overline{P_i Q_i} \times F_{v,i} - \mu_{k,v}^M \cdot \dot{q}_{k,\theta} \cdot \omega_k \quad (40)$$

donde $\mu_{k,v}^F$ y $\mu_{k,v}^M$ son los coeficientes de fricción viscosa del par cinemático k asociados a la Fuerza y Momento, respectivamente. Los vectores u_k y ω_k resultan ser vectores unitarios que suelen coincidir con la dirección del movimiento relativo entre los sólidos i y j . Sin embargo, dependiendo de la geometría del enlace, estas direcciones pueden ser diferentes a u_k y ω_k .

Si se considerase la expresión para el otro sólido j , que tiene el común el par cinemático k , esta fuerza sería igual, pero de signo contrario, y se aplicaría en el punto P_j . Los puntos P_i y P_j resultan ser el mismo punto, pero P_i se considera perteneciente al sólido i , y P_j perteneciente al sólido j . Mediante estas dos expresiones se corrobora el teorema de *Acción-Reacción*, pues ambas dos son del mismo módulo y sentido opuesto.

❖ TORSORES DE LAS FUERZAS DE FRICCIÓN CONSTANTE DE COULOMB

Un modelo válido junto con el modelo de fricción viscosa es el modelo de fricción constante de Coulomb que es muy empleado en la modelización de robots. Este modelo es similar al modelo de fricción cinética solo que éste considera una fuerza de enlace constante cualquiera, por ello que se pueda tomar como una fuerza de fricción constante independiente de las fuerzas de enlace que puedan aparecer en el par cinemático.

Por lo tanto, la expresión del torsor de fuerza de fricción constante de Coulomb definido en un punto P_i y aplicado en un punto Q_i puede escribirse de la siguiente manera:

$$F_{C,i} = -\mu_{k,C}^F \cdot F_C \cdot \text{sign}(\dot{q}_{k,s}) \cdot u_k \quad (41)$$

$$M_{C,i}^{P_i} = -\overline{P_i Q_i} \times F_{C,i} - \mu_{k,C}^M \cdot M_C \cdot \text{sign}(\dot{q}_{k,\theta}) \cdot \omega_k \quad (42)$$

donde, F_C y M_C son magnitudes constantes y desconocidas, por lo tanto, los parámetros dinámicos que se estimarán serán los productos $\mu_{k,C}^F \cdot F_C$ y $\mu_{k,C}^M \cdot M_C$. Tanto la función $\text{sign}(\dot{q}_{k,\theta})$, como la función $\text{sign}(\dot{q}_{k,s})$, corresponden con el signo de la derivada temporal de la coordenada generalizada k -ésima, \dot{q}_k , que acota el movimiento relativo entre los dos sólidos unidos por el par cinemático.

2.4. Reducción del modelo dinámico

Una vez se ha escrito el modelo dinámico en forma de un sistema de ecuaciones algebraicas, lineal respecto de los parámetros inerciales, la estimación de estos conlleva la evaluación de la matriz W y el vector τ con datos experimentales.

No obstante, el sistema resulta tener ciertas propiedades matemáticas que hacen que su resolución no sea del todo trivial. De estas propiedades se deduce que la dinámica del sistema es independiente del valor que tomen algunos para metros inerciales, y también, que una estimación del valor de estos para metros es imposible de obtener mediante la información dinámica o medidas experimentales tomadas del sistema.

Es por esto, que se han adoptado varias maneras de manejar estas dependencias de parámetros, y todas ellas conducen a modelos dinámicos reducidos, es decir, de menor tamaño que el original, que proporcionan ventajas computacionales y permiten ver la relación ente los parámetros dinámicos y la dinámica del sistema.

2.4.1. Reducción del modelo a parámetros base

De apartados anteriores se sabe que las ecuaciones dinámicas de un sistema mecánico pueden escribirse como un sistema de ecuaciones lineales, en el que el vector de incógnitas ϕ agrupa el conjunto de parámetros dinámicos, la matriz de observación del sistema $W(q, \dot{q}, q)$ es la matriz del sistema, y el vector de fuerzas externas τ es el término independiente.

$$W_{r \times c} \phi_{c \times 1} = \tau_{r \times 1} \quad (43)$$

Si el rango de la matriz W no es completa, es decir, ($\text{rango}(W) = b < c$), el modelo dinámico se reducirá al sistema de ecuaciones siguiente:

$$W_{r \times b} \phi_{b \times 1} = \tau_{r \times 1} \quad (44)$$

Para realizar esta reducción del modelo dinámico, se puede proceder de diversas maneras. En esta ocasión, en primer lugar, se eliminarán las columnas nulas de la matriz W y sus parámetros asociados del vector ϕ , y a continuación, se aplicará el método simbólico de transferencia de inercias para obtener así, Matriz de Observación del Sistema Base, denotada por W_b .

3. MÉTODOS DE ESTIMACIÓN

3.1. Introducción

La identificación de sistemas dinámicos consiste en la obtención de un modelo aproximado para describir un sistema dinámico a partir de los datos obtenidos de experimentos “entrada-salida” realizados sobre el sistema mecánico. En la Figura 1 se ilustra esquemáticamente un bloque identificador que toma información de la entrada y la salida de un sistema mecánico para obtener los parámetros de un modelo dinámico propuesto.

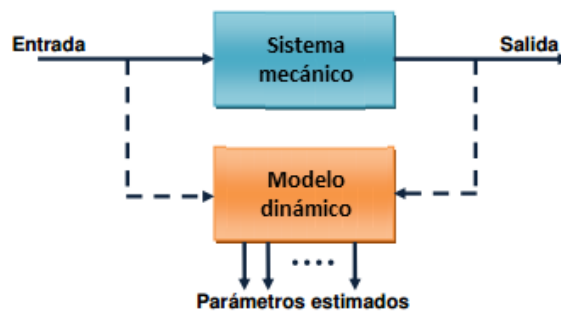


Figura 1. Bloque identificador

Las técnicas de identificación aplicadas a procesos con una entrada y una salida tienen su origen a principios de los años 70, sin embargo, sólo hasta finales de la década de los 90 empiezan a aplicarse a procesos industriales.

La identificación de sistemas mecánicos trata con el problema de construir modelos de un sistema mecánico a partir de observaciones (mediciones de datos) sobre el sistema mecánico. Podría decirse que en esencia la identificación es una modelización experimental, sin embargo, en la práctica la identificación de sistemas no solamente utiliza información de observaciones sobre el propio sistema, sino que también hace uso de información basadas en leyes físicas que gobiernan el sistema mecánico.

Son varios los métodos de estimación que se emplean en la actualidad para la identificación de parámetros dinámicos de sistemas dinámicos. En los apartados posteriores se presentará la *Estimación por Mínimos Cuadrados* y *Estimación por Máxima Verosimilitud* y especialmente se hará hincapié en este último que es el que empleará para la estimación principal de los parámetros dinámicos del robot 3-PRS.

3.2. Fundamento estadístico

3.2.1. Introducción

La estadística es una disciplina que utiliza recursos matemáticos para organizar y resumir una gran cantidad de datos obtenidos de la realidad, y obtener conclusiones respecto de ellos. La estadística puede aplicarse a cualquier ámbito de la realidad, y por ello se emplea en materias como física, química, biología, ingeniería, medicina, astronomía, psicología, sociología, lingüística y demografía, entre otros muchos.

Al iniciar el análisis estadístico de una serie de datos, y después de la etapa de detección y corrección de errores, un primer paso consiste en describir la distribución de las variables estudiadas y, en particular, de los datos numéricos tomados.

Una de las distribuciones teóricas mejor estudiadas en los textos de bioestadística y más utilizada en la práctica es la distribución normal, también conocida como distribución gaussiana. Su importancia se debe fundamentalmente a la frecuencia con la que distintas variables asociadas a fenómenos naturales y cotidianos siguen, aproximadamente, esta distribución. No obstante, y aunque algunos autores han señalado que el comportamiento de muchos parámetros en el campo de la salud puede ser descrito mediante una distribución normal, puede resultar incluso poco frecuente encontrar variables que se ajusten a este tipo de comportamiento.

El uso extendido de la distribución normal en las aplicaciones estadísticas puede explicarse, además, por otras razones. Muchos de los procedimientos estadísticos habitualmente utilizados asumen la normalidad de los datos observados. Aunque muchas de estas técnicas no son demasiado sensibles a desviaciones de la normal y, en general, esta hipótesis puede obviarse cuando se dispone de un número suficiente de datos, es por este motivo que resulta recomendable contrastar siempre si se puede asumir o no una distribución normal.

No obstante, existen otras medidas, gráficos de normalidad y contrastes de hipótesis que pueden ayudarnos a decidir, de un modo más riguroso, si la muestra de la que se dispone procede o no de una distribución normal. Cuando los datos no sean normales, podremos o bien transformarlos o emplear otros métodos estadísticos que no exijan este tipo de restricciones (los llamados métodos no paramétricos).

A continuación, se describirá la distribución normal, su ecuación matemática y sus propiedades más relevantes.

3.2.2. Distribución Normal

La distribución normal fue reconocida por primera vez por el francés *Abraham de Moivre* (1667-1754). Posteriormente, *Carl Friedrich Gauss* (1777-1855) elaboró desarrollos más profundos y formuló la ecuación de la curva, de ahí que también se la conozca, más comúnmente, como la "*campana de Gauss*".

La distribución de una variable normal está completamente determinada por dos parámetros, su media y su desviación estándar, denotadas generalmente por μ y σ . Con esta notación, la densidad de la normal viene dada por la Ecuación 45:

$$f(X) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{\left\{\frac{-(X-\mu)^2}{2\sigma^2}\right\}} ; \quad (-\infty < X < +\infty) \quad (45)$$

Así, se dice que una característica X sigue una distribución normal de media μ y varianza σ^2 , y se denota como $X \sim \mathcal{N}(\mu, \sigma)$, y su función de densidad viene dada por la Ecuación 45.

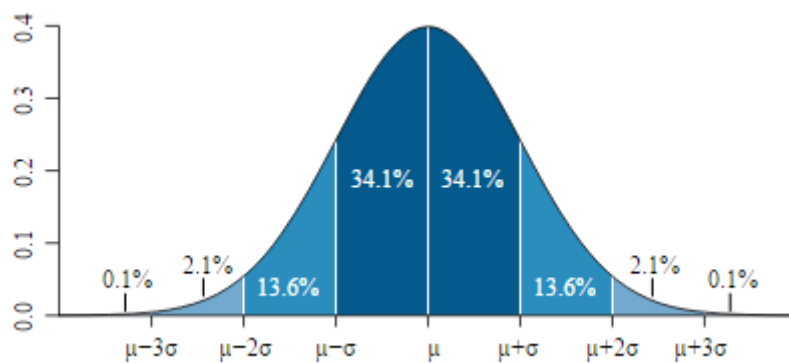


Figura 2. Distribución de probabilidad alrededor de la media en una distribución $N(\mu, \sigma)$.

Como se puede observar en la **Figura 2**, el área bajo la curva delimitada por esas líneas, indica la probabilidad de que la variable de interés, X , tome un valor cualquiera en ese intervalo. La curva alcanza su mayor altura en torno a la media μ , mientras que sus ramas se extienden asintóticamente hacia los ejes, es por esto, que cuando una variable sigue una distribución normal será mucho más probable observar un dato cercano al valor medio μ que uno que se encuentre muy alejado de éste.

La distribución normal posee ciertas propiedades importantes que conviene destacar, pues algunas de ellas serán útiles en el desarrollo del este trabajo:

- ◆ Tiene una única moda, que coincide con su media y su mediana.
- ◆ La curva normal es asintótica al eje de abscisas. Por ello, cualquier valor entre $-\infty$ y $+\infty$ es teóricamente posible. Además, el área total bajo la curva es, por tanto, igual a 1.
- ◆ Es simétrica con respecto a su media μ . Según esto, para este tipo de variables existe una probabilidad de un 50% de observar un dato mayor que la media, y un 50% de observar un dato menor.
- ◆ La forma de la *campana de Gauss* depende de los parámetros μ y σ . La media μ indica la posición de la campana, de modo que para diferentes valores de este la campana se desplaza a lo largo del eje horizontal. Por otra parte, la desviación estándar σ determina el grado de apuntamiento de la curva. Cuanto mayor sea el valor de la desviación estándar, más se dispersarán los datos en torno a la media y la curva será más plana. Un valor pequeño de este parámetro indica, por tanto, una gran probabilidad de obtener datos cercanos al valor medio de la distribución.

Una vez vistas las propiedades más representativas de la distribución normal, es interesante saber que no existe una única distribución normal, sino una familia de distribuciones con una forma común, diferenciadas por los valores de su media y su varianza. De entre todas ellas, la más utilizada es la distribución normal estándar, que corresponde a una distribución de media $\mu = 0$ y varianza $\sigma^2 = 1$. Así, la expresión que define su densidad se puede obtener de la **Ecuación 45**, resultando:

$$f(Z) = \frac{1}{\sqrt{2\pi}} \cdot e^{\left\{\frac{-Z}{2}\right\}} ; \quad (-\infty < Z < +\infty) \quad (46)$$

También es importante conocer que, a partir de cualquier variable X que siga una distribución $\mathcal{N}(\mu, \sigma)$, se puede obtener otra característica Z con una distribución normal estándar, sin más que efectuar la transformación:

$$Z = \frac{X - \mu}{\sigma} \quad (47)$$

Esta propiedad resulta especialmente interesante en la práctica, ya que para una distribución $\mathcal{N}(0,1)$ existen tablas publicadas a partir de las que se puede obtener de modo sencillo la probabilidad de observar un dato menor o igual a un cierto valor Z , y que permitirán resolver preguntas de probabilidad acerca del comportamiento de variables de las que se sabe o se asume que siguen una distribución aproximadamente normal.

3.3. Estimación por mínimos cuadrados (Least Squares Estimation)

El método de estimación por *Mínimos Cuadrados*, *MC*, o (*LSE*, *Least Squares Estimation*), es un método de estimación para aproximar sistemas de ecuaciones sobredeterminados, es decir, conjuntos de ecuaciones en las que hay más ecuaciones que incógnitas. El nombre del método viene dado pues la solución (o estimación) es aquella que minimiza la suma de los cuadrados de los errores cometidos al resolver cada ecuación del sistema de ecuaciones.

Para realizar una breve introducción sobre el método de *MC*, se va plantear el caso de regresión lineal simple y, a continuación, se generalizará para el caso de más incógnitas que será el empleado en este trabajo, a través del caso de regresión lineal múltiple.

El método de estimación *MC* para el caso de regresión lineal simple, resulta ser un procedimiento empleado para determinar la mejor línea de ajuste a unos datos determinados y utiliza el cálculo simple y el álgebra lineal para su ajuste. El problema trata de encontrar la mejor línea recta, de la forma $y = \beta_1 \cdot x + \beta_0$, dados los pares (x_n, y_n) con $n \in \{1, \dots, N\}$.

En la estimación por mínimos cuadrados, los valores de los parámetros desconocidos β_1, β_0, \dots en la función de regresión, $f(\vec{x}, \vec{\beta})$, se calculan buscando valores numéricos para los parámetros que minimizan la suma de las desviaciones cuadradas entre las respuestas observadas y la parte funcional del modelo. Matemáticamente, el criterio de mínimos cuadrados que es minimizado para obtener los parámetros es:

$$Q = \sum_{i=1}^n [y_i - f(\vec{x}_i, \vec{\beta})]^2 \quad (48)$$

Los parámetros β_1, β_0, \dots se tratan como variables de optimización y las variables x_1, x_2, \dots se tratan como coeficientes. Para hacer hincapié en el hecho de que las estimaciones de los valores de los parámetros no son los mismos que los parámetros reales, las estimaciones se denotan por $\widehat{\beta}_1, \widehat{\beta}_0, \dots$. En el caso de modelos lineales se suele resolver analíticamente mediante el cálculo matemático, sin embargo, para el caso de modelos no lineales la minimización se resuelve con algoritmos numéricos iterativos.

Si se considera el modelo de una línea recta, suponiendo una distorsión en la medición, denotada por φ ,

$$y = \beta_0 + \beta_1 \cdot x + \varphi \quad (49)$$

La estimación por mínimos cuadrados de los parámetros a identificar se calcula minimizando la función Q siguiente:

$$Q = \sum_{i=1}^n [y_i - \widehat{\beta}_0 + \widehat{\beta}_1 \cdot x_i]^2 \quad (50)$$

Para su resolución se debe realizar:

- ◆ Calcular las derivadas parciales de Q respecto a $\widehat{\beta}_0$ y $\widehat{\beta}_1$.
- ◆ Igualando cada derivada parcial a cero.
- ◆ Y resolver es sistema resultante, de dos ecuaciones y dos incógnitas. Obteniendo así los valores estimados por mínimos cuadrados:

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (51)$$

$$\widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \cdot \bar{x} \quad (52)$$

Estas aproximaciones muestran que las estimaciones de los parámetros son función tanto de la variables de entrada (x , en este caso) como de las variables de respuesta (y , en este caso) y que los estimadores no son independientes entre sí, a menos que $\bar{x} = 0$.

Lo visto hasta ahora referente al método de estimación por mínimos cuadrados está relacionado con la regresión lineal simple y no con la regresión lineal múltiple como es el caso de la identificación de los parámetros dinámicos del robot 3-PRS que se estudia en este trabajo. Es por esto que a continuación se va a proponer la Ecuación 53 que ha de resolverse para la aplicación del método de estimación por mínimos cuadrados.

$$\widehat{\delta}_{LS} = (W^T W)^{-1} W^T b \quad (53)$$

donde la matriz W se corresponde con la matriz de observación del sistema mecánico, que es función de las posiciones, velocidades y aceleraciones generalizadas del mismo, y el vector b contiene las fuerzas externas ejercidas por el sistema mecánico que se han medido. El valor de N_m , hace referencia al número de medidas realizadas.

$$W = \begin{bmatrix} W(q_m(1), \dot{q}_m(1), \ddot{q}_m(1)) \\ \vdots \\ W(q_m(N_m), \dot{q}_m(N_m), \ddot{q}_m(N_m)) \end{bmatrix}; \quad b = \begin{bmatrix} \tau_m(1) \\ \vdots \\ \tau_m(N_m) \end{bmatrix} \quad (54)$$

En este trabajo se aplicará este método para la realización de la primera aproximación de los parámetros dinámicos del robot 3-PRS, para posteriormente poder utilizarla como vector inicial para el proceso iterativo presente en el método de estimación por máxima verosimilitud.

3.4. Estimación por Máxima Verosimilitud (Maximum Likelihood Estimation)

El método empleado en este trabajo para la estimación de los parámetros dinámicos del robot 3-PRS se conoce como estimación por *Máxima Verosimilitud*, *MV*, o (*MLE*, *Maximum Likelihood Estimation*), y consiste en calcular el valor de los parámetros que más probabilidad tienen de ser los que han provocado los datos medidos.

El modelo dinámico de un robot rígido de n grados de libertad es lineal en los coeficientes de fricción y los parámetros de la distribución de las masas si se combinan en los llamados parámetros baricéntricos o inerciales. Después de la reducción del modelo, el modelo de robot dinámico puede escribirse como un conjunto mínimo de ecuaciones lineales de la forma

$$W(q, \dot{q}, \ddot{q})\phi = \tau \quad (55)$$

donde por un lado, q es el vector de dimensión n de los ángulos de articulación y τ es el vector de dimensión n también, de los pares realizados por los actuadores. Por otro lado, $W(q, \dot{q}, \ddot{q})$ es la matriz de observación de dimensiones $n \times r$, que depende de los ángulos de las articulaciones y velocidades y aceleraciones de los mismos. r corresponde con el número de parámetros

independientes del robot. Finalmente, ϕ es vector de dimensión r que contiene los parámetros inerciales desconocidos y coeficientes de fricción que se desean calcular en este trabajo.

La identificación del robot trata de estimar los parámetros del modelo ϕ (Ecuación (55)) a partir de los datos medidos durante un experimento de excitación del robot. En la mayoría de los casos, los datos son secuencias de ángulos de uniones y corrientes del motor que controla los actuadores, a partir de las cuales se calculan una secuencia de velocidades de las uniones, aceleraciones de los mismos y pares ejercidos por el motor. Se supone que los pares que generan los actuadores son proporcionales a la corriente que circula por ellos.

En la identificación de sistemas basada en un marco estadístico, como es el caso, es común suponer que los ángulos de las articulaciones medidos $q_m(k)$ y los pares de los actuadores $\tau_m(k)$, desde $k = 1$ a N , siendo este el número de mediciones realizadas sobre el robot, están corrompidos por ruido Gaussiano independiente de media cero ($\mu = 0$), que se denotará por $n_q(k), n_\tau(k)$, es decir,

$$q_m(k) = q(k) + n_q(k) \quad (56)$$

$$\tau_m(k) = \tau(k) + n_\tau(k) \quad (57)$$

La estimación por máxima verosimilitud ϕ_{ml} del vector de parámetros ϕ viene dada por valor de ϕ que maximiza la probabilidad de medición $q_m(k)$ y $\tau_m(k)$ para $k = 1$ a N . Dado que el ruido en las diferentes mediciones es independiente y gaussiano, esto corresponde a minimizar la siguiente función cuadrática denotada por K :

$$K(q_m, \tau_m | \phi) = \frac{1}{2} \sum_{k=1}^N \sum_{i=1}^n \left(\frac{n_{qi}^2(k)}{\sigma_{qi}^2} + \frac{n_{\tau i}^2(k)}{\sigma_{\tau i}^2} \right) \quad (58)$$

donde $n_{qi}(k)$ se corresponde con el ruido medido en la articulación $q_i(k)$ y $n_{\tau i}(k)$ se corresponde con el ruido medido en el par del actuador $\tau_i(k)$. Además, $\sigma_{qi}(k)$ y $\sigma_{\tau i}(k)$ son sus correspondientes varianzas que resultan ser conocidas y constantes y los índices i y k se corresponden con el número de articulaciones del sistema mecánico y medidas realizadas, respectivamente.

La minimización del anterior criterio (Ecuación 58), teniendo en cuenta la Ecuación 55, es un problema de optimización de mínimos cuadrados no lineal. Su implementación práctica requiere que se calculen $n_q(k)$ y $n_\tau(k)$ para cada estimación de ϕ dados los datos medidos $q_m(k)$ y $\tau_m(k)$. Esto no es posible con la presente formulación, puesto que los ángulos de articulación $q(k)$ y los pares de los actuadores $\tau(k)$ no se pueden calcularse con el conocimiento solo de ϕ . El vector de parámetros ϕ , es decir, los grados de libertad del problema de optimización de máxima probabilidad, debe extenderse con una parametrización de trayectoria δ : $q(\delta, k)$, $\dot{q}(\delta, k)$ y $\ddot{q}(\delta, k)$.

Esta parametrización está relacionada con la optimización de la excitación del robot y permite reformular el criterio de máxima verosimilitud (Ecuación 58) de la siguiente manera:

$$K(q_m, \tau_m | \vartheta) = \frac{1}{2} \sum_{k=1}^N \sum_{i=1}^n \left(\frac{(e_q(\delta, k))^2}{\sigma_{qi}^2} + \frac{(e_\tau(\vartheta, k))^2}{\sigma_{\tau i}^2} \right) \quad (59)$$

donde:

$$e_q(\delta, k) = q_{mi}(k) - q_i(\delta, k) \quad (60)$$

$$e_\tau(\vartheta, k) = \tau_{mi}(k) - \tau_i(\vartheta, k) \quad (61)$$

$$\vartheta = [\phi^T \ \delta^T]^T \quad (62)$$

Y $\tau_i(\vartheta, k)$ corresponde con el i -ésimo elemento de $\tau(\vartheta, k) = \Psi(q(\delta, k), \dot{q}(\delta, k), \ddot{q}(\delta, k))\phi$.

Este criterio cuadrático debe minimizarse respecto de ϑ para una secuencia dada de ángulos de articulación medidos $q_m(k)$ y pares de los actuadores $\tau_m(k)$. Esta minimización se puede realizar de diversas maneras, como pueden ser, el método de *Gauss-Newton* y el método de *Levenberg-Marquardt*, entre otros. En este trabajo se empleará la rutina *fminunc* del software *Matlab*[®], en el que está implementado el método de *Newton* con reflexión anterior.

4. MODELADO DE UN ROBOT 3-PRS

4.1. Introducción

La robótica industrial nace de la exigencia de prácticas de producción, es un elemento importante del término conocido como automatización flexible, está encaminada a la reducción de coste de producción y se considera actualmente un área de la tecnología avanzada manejadora de problemas de alta complejidad. Sus productos se constituyen en aplicaciones de las áreas de control, programación, inteligencia artificial, percepción e instrumentación, y sirven de base para el avance en diversos campos de la industria, aportando soluciones tecnológicas innovadoras orientadas al desarrollo de mejores robots y a la ampliación del abanico de aplicaciones disponibles.

En los subapartados posteriores correspondientes a esta sección, se hará una breve descripción de las características generales de los robots y se hará especial hincapié en los robots del tipo paralelo, concretamente en el robot 3-PRS, y finalmente, se desarrollará el modelo dinámico realizado de acerca del robot 3-PRS.

4.1.1. Características principales

Entre los robots considerados de más utilidad en la actualidad se encuentran los robots industriales o manipuladores. Existen ciertas dificultades a la hora de establecer una definición formal de lo que es un robot industrial.

La definición más comúnmente aceptada es la de la *Asociación de Industrias de Robótica (RIA, Robotic Industry Association)*, según la cual [45]:

"Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas"

Un manipulador robótico consta de una secuencia de elementos estructurales rígidos, denominados enlaces o eslabones, conectados entre sí mediante juntas o articulaciones, que permiten el movimiento relativo de cada dos eslabones consecutivos.

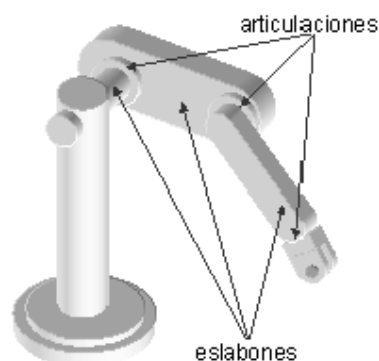


Figura 3. Elementos estructurales de un robot industrial

Existen diferentes tipos de articulaciones dependiendo del tipo de movimiento que permitan:

- ◆ Lineal (deslizante, traslacional o prismática), si un eslabón desliza sobre un eje solidario al eslabón anterior.
- ◆ Rotacional, en caso de que un eslabón gire en torno a un eje solidario al eslabón anterior.

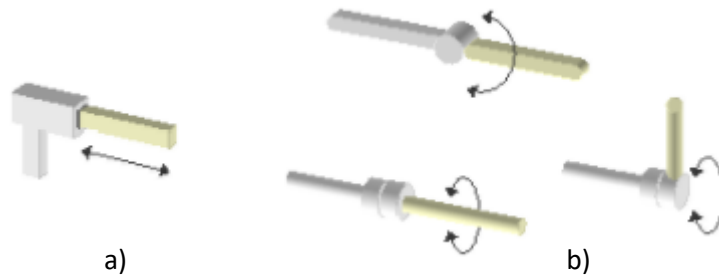


Figura 4. Distintos tipos de articulación de un robot: a) lineal, b) rotacionales

El conjunto de eslabones y articulaciones se denomina cadena cinemática. Se dice que una cadena cinemática es abierta si cada eslabón se conecta mediante articulaciones exclusivamente al anterior y al siguiente, exceptuando el primero que suele ir fijado a una base o soporte, y el último cuyo extremo final queda libre. A éste último se puede conectar un elemento terminal o actuador final: habitualmente suele ser una herramienta especial que permite al robot de uso general realizar una aplicación en particular, que debe diseñarse específicamente para dicha aplicación: una herramienta de sujeción, de soldadura, de pintura, etc. El punto más significativo del elemento terminal se denomina punto terminal (*PT*).

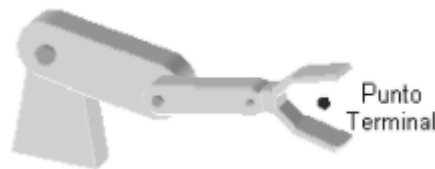


Figura 5. Punto terminal de un manipulador

Los elementos terminales pueden clasificarse en dos categorías:

- ◆ **Pinzas (gripper):** Las pinzas se utilizan para tomar un objeto, normalmente la pieza de trabajo, y sujetarlo durante el ciclo de trabajo del robot. Existen una diversidad de métodos de sujeción que pueden utilizarse, además de los métodos mecánicos más simples de agarre de la pieza entre dos o más “dedos”. Estos métodos suplementarios se sustentan en el uso de casquillos de sujeción, imanes, ganchos y cucharas.
- ◆ **Herramientas:** Una herramienta se utiliza como punto terminal en aplicaciones en donde se exija al robot realizar alguna operación sobre la pieza final. Estas aplicaciones incluyen la soldadura por puntos, la soldadura por arco, la pintura por pulverización y las operaciones de taladro, entre otras.

Se denomina grado de libertad (*GDL*) a cada uno de las coordenadas independientes que son necesarias para describir el estado del sistema mecánico del robot (posición y orientación en el espacio de sus elementos).

Para describir y controlar el estado de un robot es preciso determinar:

- ◆ La posición del punto terminal (o cualquier otro punto) respecto de un sistema de coordenadas externo y fijo, comúnmente suele ser la base fija o el suelo (*Sue*).
- ◆ El movimiento del robot cuando los elementos actuadores aplican sus fuerzas y momentos.

El análisis desde el punto de vista mecánico de un robot se puede efectuar atendiendo exclusivamente a sus movimientos (estudio cinemático) o atendiendo, además, a las fuerzas y momentos que actúan sobre sus pares (estudio dinámico) debidas a los elementos actuadores y la carga transportada por el elemento terminal.

4.1.2. Robots paralelos

Los robots o manipuladores paralelos son una concepción de robot completamente diferente a la que poseen los robots más utilizados en el ámbito de la industria, los robots serie. La definición más formal de este tipo de robots podría ser [45]:

“Se denomina manipulador paralelo general a aquel que posee un elemento terminal unido a un elemento base o fijo por medio de varias cadenas cinemáticas independientes.”

Estos robots están constituidos por cadenas cinemáticas cerradas, cuyo órgano terminal o plataforma móvil está conectado a la base mediante varias cadenas cinemáticas seriales independientes. Cada una de las cadenas cinemáticas que la componen están compuestas por diferentes tipos de pares cinemáticos, como rotacionales, prismáticas, universales, esféricas, etc. Y generalmente, solo una de las articulaciones es actuadora y el resto son pasivas.

Esta clase de robots, se suelen clasificar según el número de grados de libertad (*GDL*) que posee la plataforma móvil, que puede ser entre 2 y 6 *GDL*. En el actual trabajo se ha optado por el estudio de un robot paralelo de 3-PRS, y en los apartados posteriores se verá como trabajar con él de manera correcta y eficiente para obtener, en este caso, los parámetros dinámicos del mismo.



Figura 6. Ejemplo de robot paralelo de 6 *GDL* (hexápodo)

Los robots paralelos son utilizados principalmente en aquellas aplicaciones donde sus características los hacen ideales para resolver aquellos problemas que se presentan con los robots serie. Las características principales desde el punto de vista de las ventajas que aportan se enumeran a continuación:

- ◆ La relación carga/potencia es alta ya que los accionamientos de potencia conectan directamente la base del robot al punto terminal, sirviendo de elementos estructurales que actúan de manera simultánea, permitiéndoles manipular cargas superiores a su propio peso.
- ◆ Presentan una alta rigidez, lo cual se traduce en mayores precisiones respecto a un robot tipo serie.
- ◆ Su arquitectura les permite alcanzar altas velocidades y aceleraciones, lo cual les permite realizar tareas industriales de manera más eficiente.

Igualmente, este tipo de robots presentan ciertas desventajas, como puede ser:

- ◆ La cinemática, la definición de las posiciones singulares y la construcción del modelo dinámico son más complicados que para el caso de robots paralelos serie. La mayoría de las veces esto se resuelve de manera particular para cada configuración del robot, ya que no existen ecuaciones generales aplicables a todos ellos.

4.1.3. Robot 3-PRS

El robot empleado para la realización del trabajo se encuentra localizado en la *Universidad Politécnica de Valencia* y adquiere el nombre completo de robot (o manipulador) paralelo 3-PRS. Este, está constituido de una plataforma base y una plataforma móvil. A la plataforma base están unidos tres actuadores lineales, es decir, tres pares prismáticos \mathcal{P} , que constituyen las patas del robot y son modeladas como dos barras. Una barra inferior, constituida por el tubo exterior del actuador unida a la plataforma base fijamente. La segunda barra la conforma la parte interna del actuador y va conectada a otra barra mediante un par de revolución \mathcal{R} . Esta última barra a su vez, está unida a la plataforma móvil mediante un par esférico \mathcal{S} . De esta forma, cada pata puede verse como una cadena cinemática de pares $\mathcal{P} - \mathcal{R} - \mathcal{S}$, que dan nombre al robot 3-PRS.

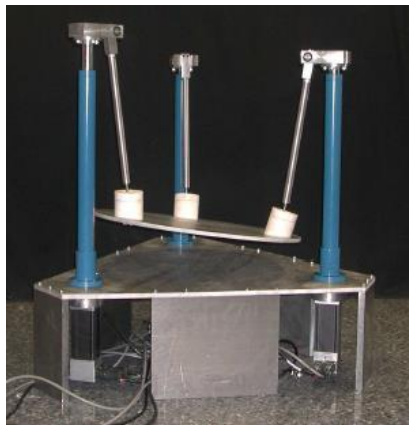


Figura 7. Robot paralelo 3-PRS real

4.2. Modelo Cinemático de un robot 3-PRS

Hasta ahora se ha realizado una descripción general de las técnicas y del procedimiento a seguir para la estimación de los parámetros dinámicos. Desde el presente apartado en adelante, se aplicarán estas técnicas al modelado del robot paralelo 3-PRS y se analizarán detalladamente.

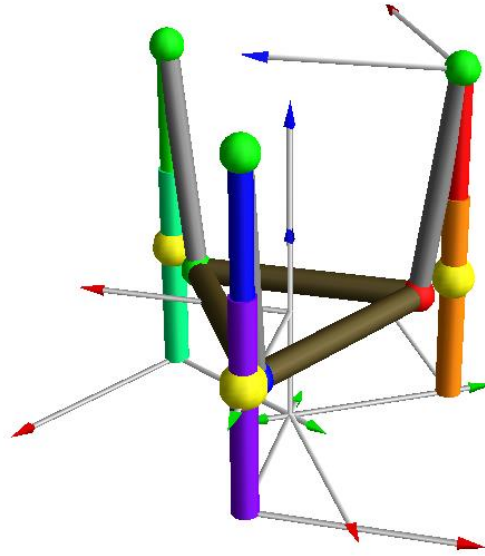


Figura 8. Representación gráfica del robot 3-PRS en *3d_mec*

Mediante el modelo cinemático del robot 3-PRS, se definirá el movimiento de cada sólido respecto de la referencia inercial (*RI*) o respecto de otros sólidos. La modelización se realizará mediante sucesivas transformaciones, por giros y traslaciones de la referencia inercial, hasta llegar a cada una de las referencias solidarias a cada sólido del sistema. Estos giros y traslaciones se contienen en los vectores de *Parámetros Geométricos* y *Coordenadas Generalizadas*.

El robot 3-PRS, como se ha detallado en la Sección 4.1.3, es un robot paralelo de 3 grados de libertad en el que mediante 3 cadenas cinemáticas del tipo $\mathcal{P} - \mathcal{R} - \mathcal{S}$ se unen la plataforma base (o fija) con la plataforma móvil como puede apreciarse en la Figura 7. Por otro lado, en la Figura 9 se puede observar una representación CAD del robot 3-PRS, y se puede apreciar como las cadenas cinemáticas están compuestas de un par prismático, una barra unida a este anterior mediante un par de revolución y finalmente, un par esférico que hace de unión entre la barra y la plataforma móvil.

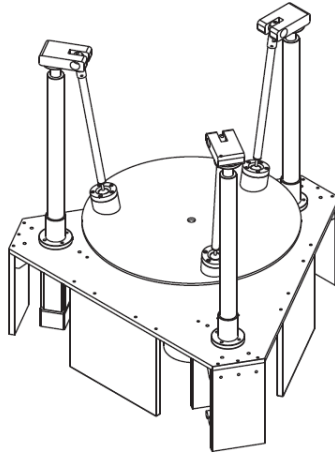


Figura 9. Representación CAD robot 3-PRS

Para definir el movimiento relativo entre la referencia inercial (*RI*) y las barras actuadoras, ha sido necesario aplicar una traslación y un giro a la referencia inercial. La traslación se realiza desplazando el origen de la referencia inercial al vértice del triángulo inferior (plataforma base), como se muestra en la Figura 10, y se denotará por r_i para el caso del actuador i . El giro se realiza respecto al eje Z de la referencia inercial y es aquel que deje el nuevo eje Y orientado hacia el centro de la plataforma base, así, el giro se denotará por α_i para el caso del actuador i .

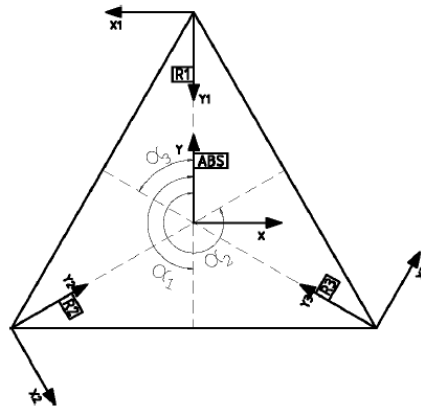


Figura 10. Parámetros geométricos relativos de giros y traslaciones de las referencias.

La traslación r_i y el giro α_i nos trasladan de la referencia inercial a la referencia a la que son solidarios los cilindros de los actuadores. Como estos parámetros permanecen constantes y no cambian con el movimiento del mecanismo, forman parte del vector de los *Parámetros Geométricos* del robot.

La única forma para la obtención de las tres traslaciones es mediante los ángulos $\alpha_1, \alpha_2, \alpha_3$ y la longitud del lado de la plataforma base L_B . Es por esto, que estos cuatro parámetros formarán parte del vector de *Parámetros Geométricos*.

Las variables que acotan la distancia entre el punto inferior del actuador y el punto superior (par cinemático de revolución) serán *Coordenadas Generalizadas* del mecanismo, y se denotarán por (s_1, s_2, s_3) . De manera similar, las variables que acotan el giro relativo entre los actuadores y las barras también serán *Coordenadas Generalizadas* y estas se denotarán por $(\theta_1, \theta_2, \theta_3)$. Además, para el posicionamiento de los puntos T_i (Figura 11 **Figura 11**), que resultan ser los puntos ubicados en las extremidades inferiores de las barras, se ha hecho uso de el parámetro geométrico L_R que corresponde con la longitud de dichas barras.

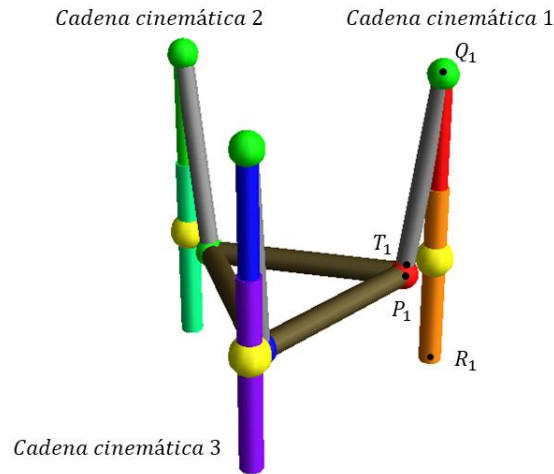


Figura 11. Representación gráfica de los puntos de referencia del robot 3-PRS

Finalmente, la plataforma móvil será acotada directamente desde la referencia inercial, el suelo (*Sue*) en este caso. Se posicionará el centro de la plataforma, denotado por O_{PM} , mediante tres coordenadas cartesianas (x, y, z) y tres ángulos (ψ_1, ψ_2, ψ_3) que corresponden con giros consecutivos del primer, segundo y tercer eje de la referencia. Estas seis coordenadas también formarán parte del vector de *Coordenadas Generalizadas*. Y en este caso para el posicionamiento de los puntos P_i que forman los vértices del triángulo de la plataforma móvil, será necesario definir tres vectores de posición, que tan solo dependerán de la longitud de este triángulo, L_N . Este parámetro se considerará un parámetro geométrico del mecanismo y, por tanto, se incluirá en el vector de *Parámetros Geométricos*.

Es por esto, que queda definido el vector de *Coordenadas Generalizadas* como:

$$q = (s_1, s_2, s_3, \theta_1, \theta_2, \theta_3, x, y, z, \psi_1, \psi_2, \psi_3)^T \quad (63)$$

Y el vector de *Parámetros Geométricos*:

$$\lambda = (L_B, L_R, L_N, \alpha_1, \alpha_2, \alpha_3)^T \quad (64)$$

4.2.1. Ecuaciones cinemáticas del robot 3-PRS

Para la realización del modelo cinemático en el software *3d_mec*, en una primera instancia se han supuesto rotas las cadenas cinemáticas en el punto del par esférico, posteriormente, mediante las 12 coordenadas generalizadas definidas en la Ecuación 63 y teniendo en cuenta que se trata de un robot de 3 grados de libertad, se necesitarán un total de 9 ecuaciones geométricas para completar el modelo cinemático. Las ecuaciones empleadas son las siguientes:

$$\overline{OP_i} - \overline{OT_i} = 0 \text{ para } i = 1,2,3. \Rightarrow 9 \text{ Ecuaciones} \quad (65)$$

donde los puntos T_i representan las extremidades inferiores de las barras de cada una de las cadenas cinemáticas y los puntos P_i se corresponden con los vértices de la plataforma móvil (véase Figura 11).

4.3. Modelo Dinámico de un robot 3-PRS

Atendiendo al planteamiento propuesto en la Sección 2.3, para la construcción del modelo dinámico se requiere de la definición de todos los torsores actuantes sobre sólidos, su punto de aplicación y la velocidad virtual de los puntos de aplicación que lleven asociada. Esto se realizará mediante la aplicación del *Principio de las Potencias Virtuales* y se obtendrán las ecuaciones dinámicas y de enlace del robot 3-PRS.

Mencionar que algunos de los vectores de fuerzas, momentos y velocidades se han expresado mediante sus coordenadas en las bases solidarias a los actuadores.

4.3.1. Torsos de inercia y gravedad

Los puntos de aplicación de los torsores de gravedad serán los propios centros de gravedad de los sólidos. Sin embargo, para los torsores de inercia, como se vio en apartados anteriores, se deben caracterizar en un punto del sólido conocido, que no sea el centro de gravedad, para evitar que el modelo dinámico quede de forma lineal respecto a los parámetros dinámicos.

Por tanto, los torsores de inercia se definen en los siguientes puntos:

- ◆ $(\mathcal{F}_{PM}, \mathcal{M}_{PM}^{O_{PM}})$: Es el torsor de inercia de la plataforma móvil (PM) aplicado en el punto O_{PM} que resulta ser el centro de la misma, y que es perteneciente al sólido PM .
- ◆ $(\mathcal{F}_{Barra,i}, \mathcal{M}_{Bar,i}^{Q_i})$: Es el torsor de inercia de la barra i , aplicado en el punto Q_i que es perteneciente al propio sólido. Donde el subíndice $i = 1,2,3$ hace referencia a cada una de las cadenas cinemáticas del robot 3-PRS.

4.3.2. Torsos de las fuerzas externas

La única fuerza externa que se considera en el robot es aquella que provoca el movimiento del mismo. Esta fuerza se aplica en los actuadores, siendo rigurosos, en los husillos que los componen. Por lo tanto, el torsor sobre cada actuador será de la siguiente manera:

$$\{F_e\} = \{0,0,\tau\}^T; \quad \{M_e^{Act}\} = \{0,0,0\}^T \quad (66)$$

Las reacciones se aplican sobre los respectivos actuadores con signo contrario al de la expresión anterior.

4.3.3. Torsores de enlace

La totalidad de los torsores se definirán en las bases en las que se hayan definido los propios sólidos.

❖ PARES ESFÉRICOS

Debido a que el par esférico permite la rotación entre las barras y la plataforma móvil (*PM*) en las tres direcciones del espacio, su torsor de enlace queda definido de la siguiente manera:

$$\{F_{\varepsilon,S}\} = \{F_{Sx}, F_{Sy}, F_{Sz}\}^T; \quad \{M_{\varepsilon,S}^{PM}\} = \{0,0,0\}^T \quad (67)$$

En este caso la acción del torsor de define actuante sobre la plataforma móvil.

4.3.4. Torsores de fricción viscosa

En este trabajo se han considerado fricciones en los tres pares cinemáticos que componen el robot, pares prismáticos, pares de revolución y pares esféricos. Estos se definen de la siguiente forma:

❖ PARES PRIMÁTICOS

$$\{F_{v,P}\} = -\mu_{v,P} \cdot \{0,0,\dot{s}\}^T; \quad \{M_{v,P}^{Act}\} = \{0,0,0\}^T \quad (68)$$

La acción se define actuante sobre el actuador.

❖ PARES DE REVOLUCIÓN

$$\{F_{v,R}\} = \{0,0,0\}^T; \quad \{M_{v,R}^{Bar}\} = -\mu_{v,R} \cdot \{\dot{\theta}, 0,0\}^T \quad (69)$$

La acción se define actuante sobre la barra.

❖ PARES ESFÉRICOS

En el caso de los pares esféricos, el momento de fricción irá en la dirección de la velocidad angular relativa de la barra respecto del de la plataforma móvil, $(\Omega_{Bar-PM}(Bar))$. Por tanto, el tórsor de fricción viscosa se define como:

$$\{F_{v,S}\} = \{0,0,0\}^T; \quad \{M_{v,S}^{PM}\} = -\mu_{v,S} \cdot (\Omega_{Bar-PM}(Bar)) \quad (70)$$

La acción del tórsor, en este caso, se define actuante sobre la plataforma móvil (PM).

4.3.5. Torsos de fricción constante de Coulomb

Los torsos de fricción constante de Coulomb, al igual que los de fricción viscosa, también tendrán la dirección del movimiento permitido por los enlaces en los que están definidos. En este caso la fuerza de fricción constante de Coulomb es proporcional a la velocidad relativa entre sólidos. Esta proporcionalidad se denotará por F_C^{eq} (*Fuerza Equivalente*) y M_C^{eq} (*Momento Equivalente*). Estas dos últimas variables resultan ser constantes y *a priori* distintas, por ello se diferencian. Por consiguiente, los torsos para los pares prismáticos, de revolución y esféricos se definen como:

❖ PARES PRISMÁTICOS

$$\{F_{C,P}\} = -F_C^{eq} \cdot \{0,0,sign(\dot{s})\}^T; \quad \{M_{C,P}^{Act}\} = \{0,0,0\}^T \quad (71)$$

La acción se define actuante sobre el actuador.

❖ PARES DE REVOLUCIÓN

$$\{F_{C,R}\} = \{0,0,0\}^T; \quad \{M_{C,R}^{Bar}\} = M_C^{eq} \cdot \{sign(\dot{\theta}),0,0\}^T \quad (72)$$

La acción se define actuante sobre la barra.

❖ PARES ESFÉRICOS

$$\{F_{C,S}\} = \{0,0,0\}^T; \quad \{M_{C,S}^{PM}\} = -M_C^{eq} \cdot \frac{(\Omega_{Bar-PM}(Bar))}{\|(\Omega_{Bar-PM}(Bar))\|} \quad (73)$$

En el caso del torsor de fricción de Coulomb en los pares esféricos, la acción se define actuante sobre la plataforma móvil.

4.3.6. Elección de las velocidades virtuales del robot 3-PRS

Una vez definidos todos los torses del modelo, lo único restante para la obtención del conjunto de ecuaciones dinámicas de movimiento y enlace, es la determinación de las velocidades virtuales a emplear para la aplicación del *Principio de Potencias Virtuales Sistematizado*.

❖ ECUACIONES DINÁMICAS DE MOVIMIENTO

Las ecuaciones dinámicas de movimiento han sido obtenidas mediante la aplicación del principio de potencias virtuales al conjunto de coordenadas generalizadas según la Ecuación 22, utilizando los *Multiplicadores de Lagrange*. La eliminación de los multiplicadores de Lagrange se realiza mediante el método conocido como *Partición de Coordenadas* descrito en las Ecuaciones 23-29.

❖ ECUACIONES DINÁMICAS DE ENLACE

Por otra parte, para poder obtener las ecuaciones dinámicas en las que aparezcan las fuerzas de enlace, se plantearán velocidades virtuales que rompan algún enlace. Los movimientos virtuales planteados son tales que tratan de romper un mínimo número de enlaces. Sin embargo, cada movimiento virtual planteado rompe uno o más enlaces, por lo que aparecerán más de una incógnita de enlace.

Los movimientos virtuales planteados para el robot 3-PRS, para la determinación de las incógnitas de enlace es la siguiente:

- ◆ Traslación y rotación de la plataforma móvil (*PM*), respecto de su centro geométrico (*OP*). Se tratan de 6 ecuaciones de enlace, tres correspondientes a la traslación y las otras tres correspondientes a la rotación. Y las incógnitas de enlace involucradas resultan ser $F_{Sxi}, F_{Syi}, F_{Szi}$, donde $i = 1,2,3$ es el subíndice que identifica cada cadena cinemática.

4.4.Reducción del modelo dinámico del robot 3-PRS

En la presente sección se realizará la reducción del modelo del robot 3-PRS para posteriormente obtener los parámetros dinámicos. En primer lugar, se analizará qué parámetros inerciales por las características del movimiento del sólido al que pertenecen, no aparecen en la expresión de las ecuaciones dinámicas. Y a continuación, se realizará una transferencia de propiedades inerciales entre los sólidos móviles, es decir, se agruparán los parámetros inerciales de uno y otro sólido, de manera que, eligiendo la propiedad inercial transferida, se reduce el modelo dinámico para que dependa de algún parámetro inercial independiente menos.

❖ ELIMINACIÓN DE PARÁMETROS:

- Debido a que los actuadores se mantienen fijos al suelo y tan solo existe una traslación en el eje *Z*, los parámetros m_{xAct_i}, m_{yAct_i} y m_{zAct_i} no aparecerán

en las ecuaciones dinámicas, con $i = 1,2,3$. Es por esto, que el modelo dinámico constará de 9 parámetros dinámicos menos.

- Asimismo, debido a que los actuadores no participan en la dinámica del robot 3-PRS pues tan solo se trasladan en el eje Z , como se ha mencionado anteriormente, los componentes de su tensor de inercia no aparecerán en las ecuaciones dinámicas. Así, en el modelo dinámico aparecerán 18 parámetros inerciales menos (6 parámetros del tensor de inercia por cada actuador).
- Puesto que las barras tan sólo tienen permitido realizar el giro respecto al eje X , debido a que están unidas por un par cinemático de revolución a los actuadores, los parámetros $m_{x_{B_i}}$ tampoco aparecerán en las ecuaciones dinámicas, donde el subíndice $i = 1,2,3$ hace referencia a cada una de las barras que componen el robot 3-PRS. Por tanto, el modelo dinámico constará de 3 parámetros dinámicos menos.
- También, debido a que las barras sólo pueden girar en el eje X , los componentes $I_{xy,B_i}, I_{xz,B_i}, I_{yy,B_i}, I_{yz,B_i}, I_{zz,B_i}$ de su tensor de inercia no aparecerán, con $i = 1,2,3$. Así pues, el modelo constará de 15 parámetros dinámicos menos.

❖ AGRUPACIÓN DE PARÁMETROS:

- Se transfiere la totalidad de la masa de cada uno de los actuadores (m_{Act_i}) a cada una de las barras (m_{B_i}), donde el subíndice $i = 1,2,3$ corresponde con la identificación de cada cadena cinemática, como se puede ver en la Figura 11. Así, la nueva masa de las barras $m_{B_i}^*$ será:

$$m_{B_i}^* = m_{B_i} + m_{Act_i} \quad (74)$$

Por tanto, de ahora en adelante la nueva masa de las barras ($m_{B_i}^*$) considerará la masa propia de la barra y del actuador de la cadena cinemática i .

- Se transfiere parte de la masa de la plataforma móvil (m_{B_iPM}) a cada una de las barras (m_{B_i}), donde el subíndice $i = 1,2,3$ corresponde con la identificación de cada cadena cinemática. Por tanto, los nuevos parámetros inerciales para cada una de las barras que se ven modificados serán:

$$m'_{B_i} = m_{B_i}^* + m_{B_iPM} = m_{B_i} + m_{Act_i} + m_{B_iPM} \quad (75)$$

$$m'_{y,B_i} = m_{y,B_i} + L_R \cdot m_{B_iPM} \quad (76)$$

$$I'_{xx,B_i} = I_{xx,B_i} + L_R^2 \cdot m_{B_iPM} \quad (77)$$

Debido a esta transferencia de masa desde la plataforma móvil (PM) a cada una de las tres barras del robot 3-PRS, la plataforma móvil sufrirá una decrecimiento en su masa, y se modificarán la posición del CG y componentes del tensor de inercia. La transferencia de masa ha de realizarse tres veces, es decir, tantas como barras tiene el robot, con el fin de simplificar la notación se denotará con el superíndice ($'$) cada vez que se considere la transferencia de masa a una barra. Considérese un vector genérico que defina la posición de los puntos P_i desde el centro geométrico O_{PM} de la plataforma móvil, de la forma:

$$\overline{O_{PM} - P_i} = \{L_x, L_y, 0\}^T \quad (78)$$

donde los componentes L_x y L_y corresponden con la distancia entre el punto O_{PM} y los puntos P_i en la dirección de los ejes X e Y , respectivamente. Así, el vector posición de los puntos P_i para cada una de las cadenas cinemáticas en función de la longitud de la arista del triángulo de la plataforma móvil, L_N , será:

$$\overline{O_{PM} - P_1} = \left\{ \frac{L_N}{2}, \frac{L_N}{2\sqrt{3}}, 0 \right\}^T \quad (79)$$

$$\overline{O_{PM} - P_2} = \left\{ -\frac{L_N}{2}, \frac{L_N}{2\sqrt{3}}, 0 \right\}^T \quad (80)$$

$$\overline{O_{PM} - P_3} = \left\{ 0, -\frac{L_N}{\sqrt{3}}, 0 \right\}^T \quad (81)$$

Por tanto, los nuevos parámetros inerciales para la plataforma móvil que se verán modificados. Inicialmente, suponiendo la transferencia de masa a una de las barras sólo, los nuevos parámetros de la plataforma móvil se definen de la siguiente manera:

$$m'_{PM} = m_{PM} - m_{B_1PM} \quad (82)$$

$$m'_{x,PM} = m_{x,PM} - L_x \cdot m_{B_1PM} \quad (83)$$

$$m'_{y,PM} = m_{y,PM} - L_y \cdot m_{B_1PM} \quad (84)$$

$$I'_{O_{PM}}(PM) = I_{O_{PM}}(PM) + (-m_{B_1PM}) \cdot \begin{bmatrix} L_y^2 & -L_x \cdot L_y & 0 \\ -L_x \cdot L_y & L_x^2 & 0 \\ 0 & 0 & L_x^2 + L_y^2 \end{bmatrix} \quad (85)$$

Suponiendo la transferencia de masa a dos de las barras, los nuevos parámetros inerciales se definen de la siguiente forma:

$$m''_{PM} = m'_{PM} - m_{B_2PM} \quad (86)$$

$$m''_{x,PM} = m'_{x,PM} - L_x \cdot m_{B_2PM} \quad (87)$$

$$m''_{y,PM} = m'_{y,PM} - L_y \cdot m_{B_2PM} \quad (88)$$

$$I''_{OPM}(PM) = I'_{OPM}(PM) + (-m_{B_2PM}) \cdot \begin{bmatrix} L_y^2 & -L_x \cdot L_y & 0 \\ -L_x \cdot L_y & L_x^2 & 0 \\ 0 & 0 & L_x^2 + L_y^2 \end{bmatrix} \quad (89)$$

Y finalmente, suponiendo la transferencia de masa a las tres barras, los nuevos y definitivos parámetros dinámicos se definen de la siguiente forma:

$$m'''_{PM} = m''_{PM} - m_{B_3PM} \quad (90)$$

$$m'''_{x,PM} = m''_{x,PM} - L_x \cdot m_{B_3PM} \quad (91)$$

$$m'''_{y,PM} = m''_{y,PM} - L_y \cdot m_{B_3PM} \quad (92)$$

$$I'''_{OPM}(PM) = I''_{OPM}(PM) + (-m_{B_3PM}) \cdot \begin{bmatrix} L_y^2 & -L_x \cdot L_y & 0 \\ -L_x \cdot L_y & L_x^2 & 0 \\ 0 & 0 & L_x^2 + L_y^2 \end{bmatrix} \quad (93)$$

La masa transferida (m_{B_iPM}) desde la plataforma móvil a cada una de las barras, resulta ser la magnitud que haga que (m'_{y,B_i}) sea nulo, es decir, $m_{B_iPM} = -m_{y,B_i}/L_R$, con $i = 1,2,3$. Por tanto, sustituyendo en todos los términos para que queden en función de los parámetros geométricos del robot 3-PRS y parámetros inerciales iniciales del mismo, los nuevos y finales parámetros inerciales de las barras y la plataforma móvil serán:

$$m'_{B_i} = m_{B_i}^* - \frac{m_{y,B_i}}{L_R} = m_{B_i} + m_{Act_i} - \frac{m_{y,B_i}}{L_R} \quad \text{para } i = 1,2,3 \quad (94)$$

$$m'_{y,B_i} = 0 \quad \text{para } i = 1,2,3 \quad (95)$$

$$I'_{xx,B_i} = I_{xx,B_i} - L_R \cdot m_{y,B_i} \quad \text{para } i = 1,2,3 \quad (96)$$

$$m'''_{PM} = m_{PM} + \frac{1}{L_R} (m_{y,B_1} + m_{y,B_2} + m_{y,B_3}) \quad (97)$$

$$m'''_{x,PM} = m_{x,PM} + \frac{L_N}{2 \cdot L_R} (m_{y,B_1} + m_{y,B_2}) \quad (98)$$

$$m'''_{y,PM} = m_{y,PM} + \frac{L_N}{\sqrt{3} \cdot L_R} \left(\frac{m_{y,B_1}}{2} + \frac{m_{y,B_2}}{2} + m_{y,B_3} \right) \quad (99)$$

$$I'''_{OPM}(PM) = \begin{bmatrix} I'''_{xx,PM} & I'''_{xy,PM} & I'''_{xz,PM} \\ I'''_{xy,PM} & I'''_{yy,PM} & I'''_{yz,PM} \\ I'''_{xz,PM} & I'''_{yz,PM} & I'''_{zz,PM} \end{bmatrix} \quad (100)$$

donde los componentes del tensor de inercia resultan ser:

$$I'''_{xx,PM} = I_{xx,PM} + \frac{L_N^2}{4 \cdot L_R} \left(\frac{m_{y,B_1}}{3} + \frac{m_{y,B_2}}{3} + m_{y,B_3} \right) \quad (101)$$

$$I'''_{yy,PM} = I_{yy,PM} + \frac{L_N^2}{4 \cdot L_R} (m_{y,B_1} + m_{y,B_2}) \quad (102)$$

$$I'''_{zz,PM} = I_{yy,PM} + \frac{L_N^2}{3 \cdot L_R} (m_{y,B_1} + m_{y,B_2} + m_{y,B_3}) \quad (103)$$

$$I'''_{xy,PM} = I_{xy,PM} + \frac{L_N^2}{4 \cdot \sqrt{3} \cdot L_R} (m_{y,B_2} - m_{y,B_1}) \quad (104)$$

$$I'''_{xz,PM} = I_{xz,PM} \quad (105)$$

$$I'''_{yz,PM} = I_{yz,PM} \quad (106)$$

Una vez realizada la reducción del modelo mediante las dos técnicas mostradas, se obtiene lo que se denomina *Matriz de observación del Sistema Base* y viene denotada por W_b . Esta matriz, será la matriz de observación del robot 3-PRS que se empleará para la aplicación del método de estimación por *MV*.

4.5. Adquisición de datos

Cabe recordar que las variables necesarias para la identificación de parámetros son las fuerzas ejercidas por los actuadores y la posición de los mismos. Generalmente, no se realiza una medida directa de las variables necesarias, si no que se suele asumir una relación lineal entre corriente eléctrica y el par ejercido por el motor, es decir:

$$\tau_m = K_m \cdot i_m \quad (107)$$

donde τ_m es el par (o fuerza, en este caso) aplicado por el motor, i_m es la corriente eléctrica que circula por el motor y K_m es la constante que relaciona las dos anteriores variables. Normalmente, esta constante se suele estimar mediante la experimentación sobre motores.

Por otra parte, la gran mayoría de robots industriales vienen provistos de aparatos de medición de la variable posición con gran exactitud. Sin embargo, no cuentan por lo general, de aparatos para la medición directa de la velocidad y aceleración. Una forma de obtener dichas variables es mediante derivación numérica de la posición mediante diferencias centrales, pues resultan ser las más precisas. La velocidad y la aceleración de cualquier coordenada generalizada q_i , viene dada por:

$$\dot{q}_i(t) = \frac{(q_i(t+1) - q_i(t-1))}{2 \cdot \Delta t} \quad (108)$$

$$\ddot{q}_i(t) = \frac{(\dot{q}_i(t+1) - \dot{q}_i(t-1))}{2 \cdot \Delta t} \quad (109)$$

donde Δt corresponde con un diferencial de tiempo, es decir, un intervalo de tiempo muy pequeño.

4.6. Parámetros base del robot 3-PRS

Una vez eliminado y agrupado los parámetros inerciales que se han considerado anteriormente, los parámetros base a estimar se muestran en la Tabla 1. Los parámetros base a partir de ahora se agruparán en el vector ϕ_b , denominado vector de *Parámetros Base*.

$\phi_{b,1}$	$m_{PM} + \frac{1}{L_R} (m_{y,B_1} + m_{y,B_2} + m_{y,B_3})$	$\phi_{b,15}$	m_{z,B_2}
$\phi_{b,2}$	$m_{x,PM} + \frac{L_N}{2 \cdot L_R} (m_{y,B_1} + m_{y,B_2})$	$\phi_{b,16}$	$I_{xx,B_2} - L_R \cdot m_{y,B_2}$
$\phi_{b,3}$	$m_{y,PM} + \frac{L_N}{\sqrt{3} \cdot L_R} \left(\frac{m_{y,B_1}}{2} + \frac{m_{y,B_2}}{2} + m_{y,B_3} \right)$	$\phi_{b,17}$	$m_{B_3} + m_{Act_3} - \frac{m_{y,B_3}}{L_R}$
$\phi_{b,4}$	$m_{z,PM}$	$\phi_{b,18}$	m_{z,B_3}
$\phi_{b,5}$	$I_{xx,PM} + \frac{L_N^2}{4 \cdot L_R} \left(\frac{m_{y,B_1}}{3} + \frac{m_{y,B_2}}{3} + m_{y,B_3} \right)$	$\phi_{b,19}$	$I_{xx,B_3} - L_R \cdot m_{y,B_3}$
$\phi_{b,6}$	$I_{yy,PM} + \frac{L_N^2}{4 \cdot L_R} (m_{y,B_1} + m_{y,B_2})$	$\phi_{b,20}$	$\mu_{v,P1}$
$\phi_{b,7}$	$I_{yy,PM} + \frac{L_N^2}{3 \cdot L_R} (m_{y,B_1} + m_{y,B_2} + m_{y,B_3})$	$\phi_{b,21}$	$\mu_{v,P2}$
$\phi_{b,8}$	$I_{xy,PM}''' = I_{xy,PM} + \frac{L_N^2}{4 \cdot \sqrt{3} \cdot L_R} (m_{y,B_2} - m_{y,B_1})$	$\phi_{b,22}$	$\mu_{v,P3}$
$\phi_{b,9}$	$I_{xz,PM}$	$\phi_{b,23}$	$\mu_{v,R1}$
$\phi_{b,10}$	$I_{yz,PM}$	$\phi_{b,24}$	$\mu_{v,R2}$
$\phi_{b,11}$	$m_{B_1} + m_{Act_1} - \frac{m_{y,B_1}}{L_R}$	$\phi_{b,25}$	$\mu_{v,R3}$
$\phi_{b,12}$	m_{z,B_1}	$\phi_{b,26}$	$F_{C,P1}^{eq}$
$\phi_{b,13}$	$I_{xx,B_1} - L_R \cdot m_{y,B_1}$	$\phi_{b,27}$	$F_{C,P2}^{eq}$
$\phi_{b,14}$	$m_{B_2} + m_{Act_2} - \frac{m_{y,B_2}}{L_R}$	$\phi_{b,28}$	$F_{C,P3}^{eq}$

Tabla 1. Parámetros base simbólicos del robot 3-PRS

Los parámetros base, $\phi_{b,i}$, referidos a los subíndices i comprendidos entre 1-19 corresponden con los parámetros inerciales y los restantes, referidos a los subíndices

comprendidos entre 20-28 corresponden con los parámetros de fricción propuestos (viscosa y constante de Coulomb).

Mencionar que se han eliminado del modelo los parámetros de fricción correspondientes a los viscosos de los pares esféricos, los constantes de Coulomb de los pares de revolución y los constantes de Coulomb de los pares esféricos. Para estos parámetros, en estimaciones preliminares se han obtenido valores evidentemente erróneos, por lo que se ha decidido eliminarlos del modelo y eliminar así los posibles errores cometidos en esta parte del modelo. Por tanto, a la hora de la aplicación del método de MV, se estimarán un total de 28 parámetros dinámicos.

5. APLICACIÓN DEL MÉTODO DE ESTIMACIÓN POR MV A UN ROBOT 3-PRS Y RESULTADOS

5.1. Introducción

El principal fin del presente trabajo es la estimación de los parámetros dinámicos mediante el método de estimación por *MV*. Por ello, en esta sección se tratará en primer lugar sobre la parametrización de la trayectoria que realizan los actuadores lineales del robot 3-PRS y se estimarán los parámetros de la trayectoria. A continuación, se explicará cómo se ha desarrollado el método de estimación por *MV*, complementadas por el código desarrollado correspondiente. Posteriormente, se obtendrán los parámetros dinámicos del robot. Y como final del apartado, se estudiarán los parámetros estimados mediante representaciones gráficas y se realizará una validación del modelo.

5.2. Parametrización de la trayectoria

La parametrización que se ha realizado para la trayectoria, fue propuesta por *Swevers et al.* [43] y consiste en una parametrización mediante series finitas de Fourier. De esta manera, las posiciones, velocidades y aceleraciones, para la i -ésima coordenada independiente quedarían así:

$$q_i(t) = a_{i0} + \sum_{j=1}^{N_H} [a_{ij} \sin(j\omega t) + b_{ij} \cos(j\omega t)] \quad (110)$$

$$\dot{q}_i(t) = \sum_{j=1}^{N_H} [a_{ij} j\omega \cos(j\omega t) - b_{ij} j\omega \sin(j\omega t)] \quad (111)$$

$$\ddot{q}_i(t) = \sum_{j=1}^{N_H} [-a_{ij} j^2 \omega^2 \sin(j\omega t) - b_{ij} j^2 \omega^2 \cos(j\omega t)] \quad (112)$$

donde N_H es el número de armónicos empleados en la serie finita de Fourier, ω es la pulsación fundamental, y el periodo fundamental viene dado por la expresión $T_f = 2\pi/\omega$. Los parámetros de la trayectoria de los que dependen estas funciones son a_{i0} y los coeficientes a_{ij} y b_{ij} . Así, el vector de parámetros de la trayectoria δ , quedaría de la siguiente manera:

$$\delta = (\delta_1, \dots, \delta_{N_{GDL}}) \quad (113)$$

donde para cada grado de libertad (*GDL*) deberán de determinarse $2 \cdot N_H + 1$ parámetros:

$$\delta_i = (a_{i0}, a_{i1}, b_{i1}, \dots, a_{iN_H}, b_{iN_H}) \quad (114)$$

El uso de esta clase de parametrización se debe a las buenas propiedades que tiene a la hora de optimizar trayectorias y estimar parámetros dinámicos. En primer lugar, las trayectorias parametrizadas son periódicas y dan lugar a la repetición de la trayectoria en los experimentos reduciendo la relación ruido-síñal. Por otro lado, es posible estimar las características del ruido de la medición, tanto de la posición como de las fuerzas de los actuadores. Esto es de gran valor en este caso, pues se van a estimar los parámetros mediante el método de estimación por *MV*, como es el caso.

Además, las series de Fourier, dado un número suficiente de armónicos, son capaces de representar cualquier función periódica con la precisión deseada. Y finalmente, a la hora de obtener las velocidades y aceleraciones de las coordenadas independientes del movimiento medido del sistema, es posible derivar simplemente analíticamente las expresiones respecto del tiempo, sin necesidad de utilizar métodos de derivación numérica que introducen ruido a alta frecuencia.

En este trabajo, al tratarse de un robot localizado en la *Universidad Politécnica de Valencia*, el personal docente allí localizado, nos ha proporcionado los datos medidos tanto de la posición de los tres actuadores y de las fuerzas externas que ejercen para recorrer la trayectoria impuesta. Por este motivo, se han tenido que estimar los parámetros de la serie de Fourier de la trayectoria realizada. En la Figura 12, se muestra la trayectoria que han realizado cada uno de los actuadores que componen el robot 3-PRS.

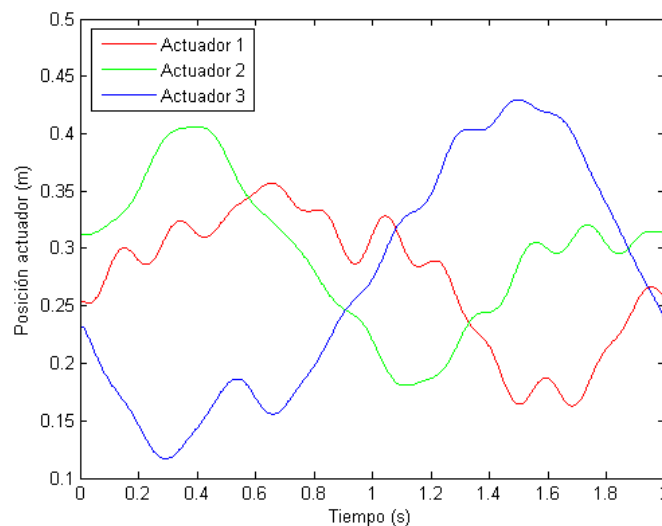


Figura 12. Un periodo de la trayectoria realizada por los 3 actuadores del robot 3-PRS

Se trata de una serie finita de Fourier que consta de $N_H = 11$ armónicos y por tanto, es función de 23 parámetros de la trayectoria para cada actuador. La pulsación fundamental de la serie es $\omega = \pi \text{ rad/s}$ y se han tomado $N_m = 4000$ mediciones de datos en $N_{per} = 5$

repeticiones consecutivas de la trayectoria, es decir, cada periodo consta de $N_m/N_{Per} = 800$ medición de datos.

La trayectoria realizada por cada uno de los actuadores está diseñada y optimizada para que la estimación de los parámetros dinámicos sea lo más precisa a la realidad posible. En primera instancia se han estimado los parámetros de trayectoria, contenidos en el vector de *Parámetros de Trayectoria* δ , (véase Sección 9.3.4) ajustando la serie de Fourier a la trayectoria realizada por cada uno de los actuadores, y resolviendo el sistema lineal siguiente:

$$q_{k,i} = \begin{bmatrix} 1 & \sin(\omega t_0) & \cos(\omega t_0) & \cdots & \sin(j\omega t_0) & \cos(j\omega t_0) \\ 1 & \sin(\omega t_1) & \cos(\omega t_1) & \cdots & \sin(j\omega t_1) & \cos(j\omega t_1) \\ 1 & \sin(\omega t_2) & \cos(\omega t_2) & \cdots & \sin(j\omega t_2) & \cos(j\omega t_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \sin(\omega t_{m-1}) & \cos(\omega t_{m-1}) & \cdots & \sin(j\omega t_{m-1}) & \cos(j\omega t_{m-1}) \\ 1 & \sin(\omega t_m) & \cos(\omega t_m) & \cdots & \sin(j\omega t_m) & \cos(j\omega t_m) \end{bmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ \vdots \\ a_{11} \\ b_{11} \end{pmatrix} \quad (115)$$

donde q_i corresponde con el vector *Coordenada Generalizada* que define la posición del actuador de la cadena cinemática i , y el suíndice $m = 0, \dots, N_m/N_{Per}$, hace referencia al número de puntos medidos en cada periodo de la trayectoria realizada. En la Tabla 2 se muestran los parámetros de la trayectoria estimados para cada uno de los actuadores:

δ	Actuador 1	Actuador 2	Actuador 3	δ	Actuador 1	Actuador 2	Actuador 3
a_0	0,27335485	0,29564953	0,27032503	b_6	0,00154399	0,00378063	-0,00331147
a_1	0,06910492	0,05340181	-0,14083116	a_7	-0,0028387	0,00080806	-0,00335409
b_1	-0,03117595	0,0577005	-0,02107945	b_7	-0,00403474	-0,00135278	-0,00011368
a_2	0,00950815	-0,00735861	-0,0110829	a_8	0,00431717	-0,00026887	0,00141004
b_2	0,015278	-0,03580231	-0,01919076	b_8	-0,00383783	0,00360638	0,00097421
a_3	-0,01031024	0,00434373	-0,00837452	a_9	-0,0037859	-0,00055385	-0,0022407
b_3	0,01067674	-0,01181654	-0,0009833	b_9	-0,00289048	0,00251067	2,13E-05
a_4	0,00330997	-0,00870686	0,00123442	a_{10}	1,64E-05	-0,00366333	-0,00038914
b_4	-0,00403585	0,00076625	0,0093308	b_{10}	0,00238549	-0,00141834	-0,00209293
a_5	-0,00572629	-1,98E-05	0,00816936	a_{11}	-0,00911247	0,00043304	0,00121136
b_5	0,00010665	0,00281261	0,00104372	b_{11}	-0,0003364	-0,00376233	0,00354174
a_6	-0,00217241	-0,00044145	-0,00102704				

Tabla 2. Parámetros de la trayectoria estimados para los 3 actuadores del robot 3-PRS

5.3. Desarrollo del método Máxima Verosimilitud

La estimación por *Máxima Verosimilitud* se ha realizado a través del algoritmo que se ha desarrollado (véase Sección 9.3.1) mediante el software *Matlab*[®], en el cual está implementado todo lo visto en la Sección 3.4.

En primer lugar, se ha realizado una primera estimación del vector *Parámetros de Trayectoria* y *Parámetros Dinámicos* que se incluirán en el vector denotado por ϑ_0 , es decir, se ha realizado una primera aproximación de los parámetros de la trayectoria para cada uno de los actuadores del robot 3-PRS y una primera aproximación de todos los parámetros dinámicos del mismo. La estimación de los parámetros de la trayectoria se ha realizado como se ha detallado en la Sección 5.2.

Una vez resuelto el sistema lineal para la obtención de los parámetros de la trayectoria de cada uno de los actuadores, se ha realizado una primera aproximación de los parámetros dinámicos mediante el método de estimación por *Mínimos Cuadrados (MC)* de acuerdo con lo visto en la Sección 3.3. Es decir, a través de las fuerzas externas medidas en cada uno de los actuadores τ_m y la matriz de observación del sistema base W_b se ha aplicado el método (véase Sección 9.3.1):

$$\hat{\phi}_{LS} = (W_b' \cdot W_b)^{-1} \cdot W_b' \cdot \tau_m \quad (116)$$

Una vez obtenida la primera aproximación de los parámetros a estimar se ha procedido a estimarlos mediante el método de estimación por *MV*. Para ello, se ha empleado la rutina *fminunc* de *Matlab*[®] que localiza mediante un proceso iterativo el mínimo de una función escalar objetivo. En este caso, la función objetivo será la función cuadrática K descrita en la Sección 3.4, que es función de los errores de las fuerzas de los actuadores, de sus posiciones y de las varianzas de estas (véase Sección 9.3.2).

Las varianzas se sabe que son conocidas y constantes, por ello se han aproximado como el 5% del valor máximo de la posición de los actuadores y la fuerza ejercida por los actuadores, siendo estas $\sigma_q^2 = 0.021 \text{ m}^2$ y $\sigma_\tau^2 = 0.112 \text{ N}^2$, respectivamente. La estimación de las varianzas de la posición y fuerza de los actuadores no se ha podido actualizar en cada iteración del algoritmo *MV*, por lo que se mantienen constantes a lo largo de las iteraciones.

Una vez obtenidos todos los valores de las variables necesarias para el inicio del proceso iterativo, ha evaluado en cada iteración la función K obteniendo una nueva aproximación de los *Parámetros Dinámicos* y *Parámetros de la Trayectoria*, renovando así en cada iteración el vector ϑ . El proceso iterativo termina cuando la diferencia entre la norma del vector ϑ estimado en la iteración anterior y actual es menor que una tolerancia TOL definida por defecto por *Matlab*[®], en este caso, $(\|\hat{\vartheta}_{j-1} - \hat{\vartheta}_j\|) < TOL = 10^{-7}$. El subíndice j corresponde con la iteración j -ésima, siendo $j = 1, \dots, N_{iter}$.

Una vez terminado el proceso iterativo, que así ha sido en la iteración número $N_{iter} = 2$ con un valor mínimo de la función $K = 6.08164e + 12$, se han obtenido el conjunto de los valores de los parámetros dinámicos y de la trayectoria estimados que minimizan la función K , y se han agrupado en el vector $\vartheta_{ML} = [\phi_{ML}^T \delta_{ML}^T]^T$. Los parámetros dinámicos estimados mediante el método de estimación por MV , ϕ_{ML} , se muestran en la Sección 5.4.

5.4. Parámetros dinámicos estimados

Una vez realizados los experimentos, el siguiente paso es la estimación de parámetros dinámicos mediante el método de MV . Los parámetros están expresados en unidades del *Sistema Internacional* (kg, m). En la Tabla 3 se muestran dichos parámetros dinámicos estimados:

$\phi_{ML,1}$	0,170107	$\phi_{ML,15}$	-0,004219
$\phi_{ML,2}$	-0,008149	$\phi_{ML,16}$	0,005323
$\phi_{ML,3}$	-0,043288	$\phi_{ML,17}$	0,054453
$\phi_{ML,4}$	-0,026152	$\phi_{ML,18}$	0,016499
$\phi_{ML,5}$	0,027760	$\phi_{ML,19}$	-0,003362
$\phi_{ML,6}$	-0,009261	$\phi_{ML,20}$	1,465675
$\phi_{ML,7}$	0,007724	$\phi_{ML,21}$	0,993647
$\phi_{ML,8}$	-0,034181	$\phi_{ML,22}$	1,258899
$\phi_{ML,9}$	-0,008096	$\phi_{ML,23}$	-0,658945
$\phi_{ML,10}$	-0,009784	$\phi_{ML,24}$	-0,330605
$\phi_{ML,11}$	-0,037740	$\phi_{ML,25}$	-0,076949
$\phi_{ML,12}$	-0,007598	$\phi_{ML,26}$	0,319082
$\phi_{ML,13}$	-0,041253	$\phi_{ML,27}$	0,660516
$\phi_{ML,14}$	0,000492	$\phi_{ML,28}$	0,568741

Tabla 3. Parámetros dinámicos del robot 3-PRS obtenidos por MV

5.5. Validación del modelo dinámico

En este apartado se tratará de comprobar que el modelo dinámico que se ha construido (e identificado) mediante datos experimentales, realmente representa el robot 3-PRS. Para ello se comprobará que el modelo, se ajusta a los datos experimentales con los que se ha obtenido. Para realizar dicha comprobación se compararán diferentes datos estimados mediante los métodos de estimación MC y MV en diferentes gráficas, como pueden ser, las fuerzas ejercidas

estimadas por los actuadores frente a las fuerzas ejercidas y las trayectorias realizadas estimadas de los mismos frente a las trayectorias realizadas.

5.5.1. Comparativa Trayectoria realizada vs. Trayectoria realizada estimada

En este primer apartado, en la Figura 13 se muestra la diferencia entre la trayectoria realizada y la estimada mediante el método de estimación por *MV* para cada uno de los actuadores del robot 3-PRS.

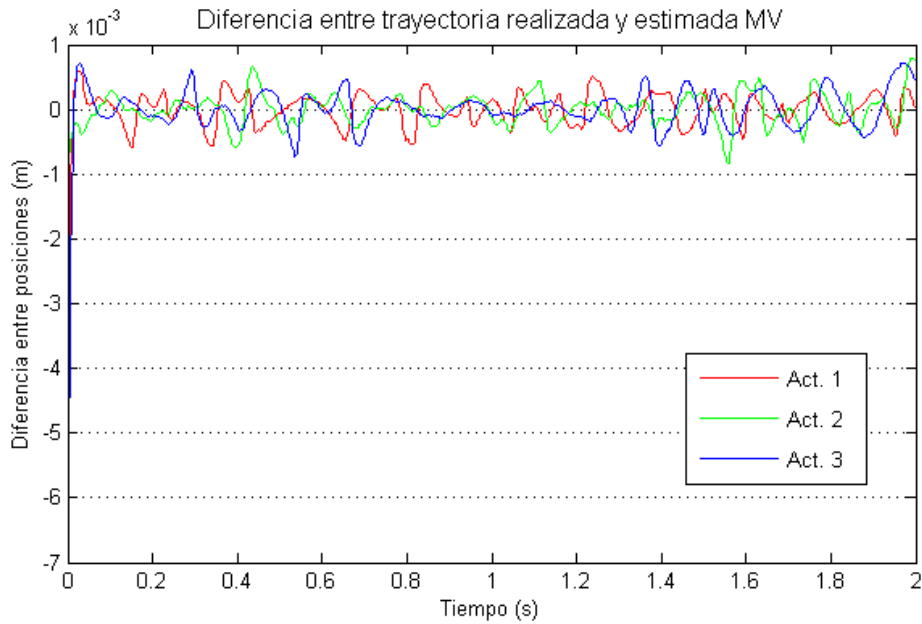


Figura 13. Diferencia entre la trayectoria realizada y estimada por *MV*

Se observa como la diferencia entre la trayectoria realizada y la estimada por *MV*, para el caso de los tres actuadores, es del orden de $10^{-3} m$ a lo largo de todo el periodo. Al ser la diferencia pequeña se podría decir que la trayectoria estimada por *MV* es semejante a la realmente realizada por cada uno de los actuadores, por tanto, se podría decir que se está ante una buena aproximación de los parámetros de la trayectoria, δ_{ML} .

5.5.2. Comparativa Fuerza ejercida vs. Fuerza ejercida estimada

En este caso, en las Figuras 14-15 se recogen las fuerzas externas realizadas y estimadas mediante el método de estimación por *MC* y *MV*, para cada uno de los actuadores. Se muestran un total de seis líneas en cada representación, cada color hace referencia a cada uno de los actuadores del robot 3-PRS y en cuanto al tipo de línea, la línea continua corresponde con la fuerza realizada por el actuador y la línea a puntos corresponde con la fuerza realizada estimada por *MC* o *MV*.

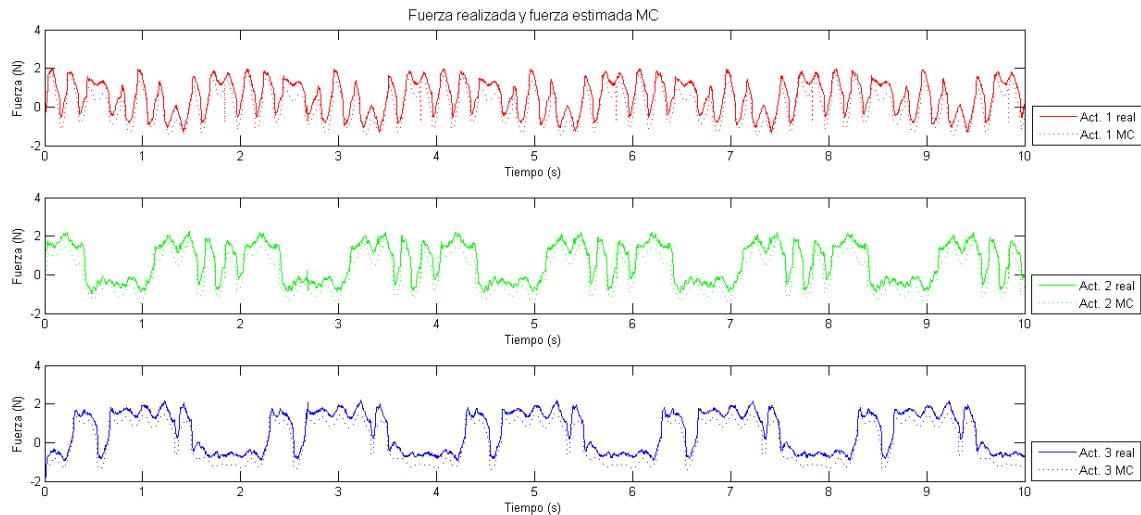


Figura 154. Fuerza realizada y estimada por *MC* para cada uno de los actuadores

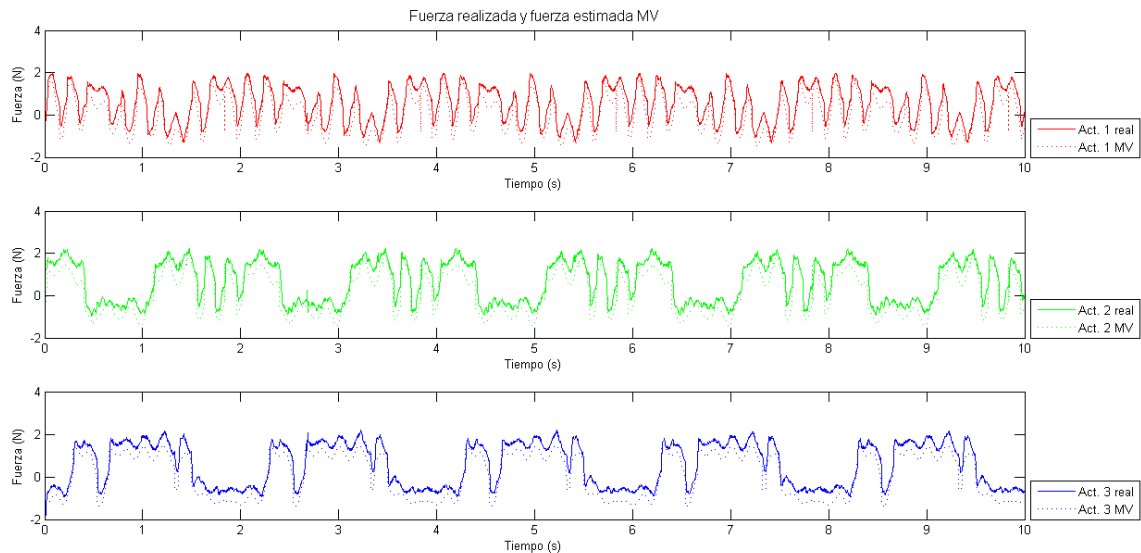


Figura 145. Fuerza realizada y estimada por *MV* para cada uno de los actuadores

Se puede observar como entre la fuerza realizada y la fuerza estimada, ya sea por el método de estimación por *MC* o *MV*, existe una especie de traslación entre ambas resultando que no se asemejen mucho entre ellas, es decir, se aprecia la existencia de un error sistemático de carácter no aleatorio, pues, a lo largo de todo el periodo se mantiene la mencionada traslación constante. En un principio, la estimación por *MC* debería ser bastante próxima al valor de la estimación final

de los parámetros dinámicos, ya que el tipo de robots como el 3-PRS apenas genera ruido en las mediciones de las articulaciones, y proporcionan buenos resultados al aplicar este método.

A la vista de esto, se podría decir que la estimación de parámetros dinámicos no ha sido del todo correcta, pues, si la estimación hubiera sido algo más precisa ambas curvas serían muy parecidas. Este error en la estimación de los parámetros se puede achacar al mal condicionamiento de la matriz W_{base} , del orden de 10^{10} , haciendo que se distorsione la estimación de los parámetros. Además, a pesar de haber desarrollado el procedimiento para la estimación de los parámetros dinámicos de manera correcta, debido a que el código desarrollado es extenso y consta de muchas partes posee algún fallo. Sin embargo, el procedimiento mismo empleado para la estimación de los parámetros del robot 3-PRS, se empleó para el caso de un péndulo doble, mediante el cual se comprobó que el método y el procedimiento seguido era el correcto.

5.5.3. Relación entre la velocidad y la fuerza en los actuadores

Debido al modelo de fricción constante de Coulomb empleado, se han observado que cuando se dan los cambios de signo de la velocidad de los actuadores hay un salto importante en la magnitud de las fuerzas, como se muestra en la Figura 16.

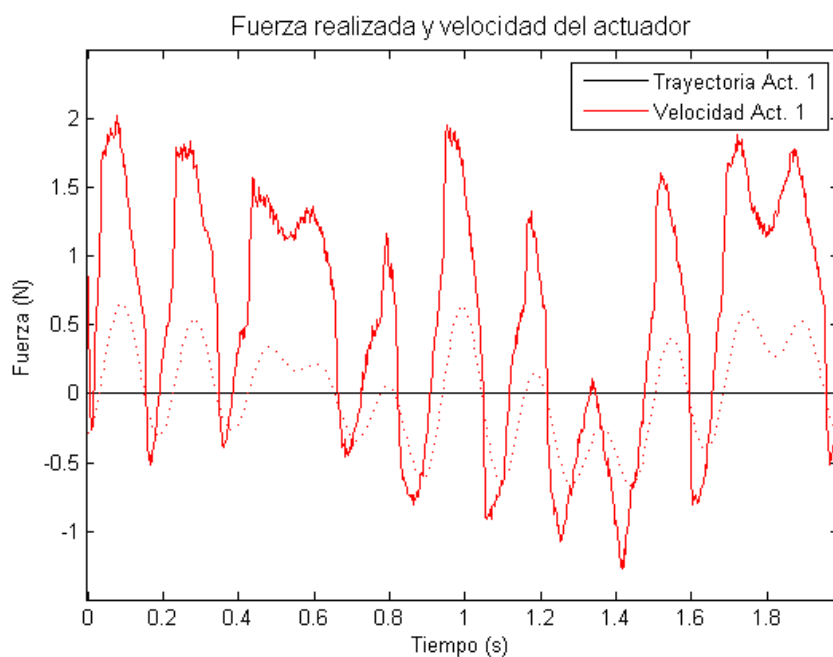


Figura 16. Trayectoria realizada y velocidad del actuador 1

Se observa que cuando la velocidad del actuador es entorno a cero, es decir, al cruzar la línea negra, los valores de las fuerzas se disparan. Como se ha mencionado esto es debido al modelo de fricción constante de Coulomb empleado en el modelo.

6. CONCLUSIONES

Este apartado va dirigido a obtener varias conclusiones y a remarcar la metodología empleada para la estimación de los parámetros dinámicos del robot 3-PRS. Concretamente, en este trabajo se ha presentado el modelado dinámico e identificación del mismo. Así pues, en los subapartados siguientes se tratará sobre ellos como apartado final del trabajo. En primer lugar, se abordará el tema del modelado dinámico, a continuación, aquello relacionado con el diseño de experimentos, posteriormente, se tratará sobre la identificación del modelo del robot 3-PRS, y a modo final del trabajo realizado se efectuarán una serie de conclusiones.

❖ MODELADO DINÁMICO

En este primer término denominado modelado dinámico, se presentó el desarrollo del modelo dinámico utilizado en la identificación de parámetros. Se empleó la formulación del Principio de Potencias Virtuales para obtener las ecuaciones dinámicas del robot 3-PRS. Posteriormente, se caracterizaron todos los torses considerados en el modelo, caracterizándolos en puntos que hiciesen que el modelo fuese lo más *simple* posible. A continuación, el modelo fue reescrito de forma lineal con respecto a los parámetros dinámicos a identificar. En el modelado se ha optado por considerar tan solo modelos de fricción lineal en los pares cinemáticos del tipo viscoso y constante de Coulomb. Sin embargo, se optó por no modelar los elementos activos, como las inercias de los motores, entre otros, por motivos de simplificación.

Mediante el modelado dinámico se pone de manifiesto que no todos los parámetros del modelo dinámico contribuyen en la misma medida a la dinámica del robot 3-PRS, sino que solo algunos de ellos actúan bien de forma independiente o bajo combinaciones lineales entre ellos. La determinación de dicho modelo, denominado modelo en parámetros base, se desarrolló en un primer paso eliminando las columnas nulas de la matriz W y sus parámetros asociados del vector ϕ , para a continuación, mediante el procedimiento simbólico basado en la transferencia de parámetros inerciales, calcular las expresiones simbólicas de los parámetros base y eliminar las columnas de W que aparecían como combinaciones lineales de otras.

❖ DISEÑO DE EXPERIMENTOS

El segundo aspecto abordado consistió en el diseño de experimentos orientado en obtener una estimación de los parámetros dinámicos lo más precisos a la realidad. A pesar de que este aspecto no se haya tratado como tal en este trabajo, debido a que el robot se encuentra localizado en la *Universidad Politécnica de Valencia (UPV)* caben destacar varios detalles. El primero, con el fin de minimizar los ruidos (errores) en las mediciones, en la *UPV* se diseñaron trayectorias mediante procedimientos de optimización hasta llegar a la conclusión de que las trayectorias definidas por series finitas de Fourier proporcionaban muy buenos resultados. Tanto es así, que se diseñaron trayectorias específicas de esta clase para cada uno de los actuadores del robot 3-PRS. Y posteriormente, en lo referente al procesamiento de datos obtenidos en los experimentos la información fue incluida en el

procedimiento de aplicación del método de estimación por *MV* para la estimación de los parámetros.

A partir de las mediciones proporcionadas por la *UPV*, en este trabajo se ha realizado el ajuste de los parámetros de trayectoria que mejor representaban estos datos, debiendo determinar el número de armónicos utilizado, la frecuencia fundamental de estos y el valor de los parámetros de la trayectoria.

❖ IDENTIFICACIÓN DEL MODELO

Por último, el aspecto final abordado fue aquél relacionado con la identificación de los parámetros dinámicos del modelo del robot 3-PRS. Para la identificación de los parámetros dinámicos se desarrolló una serie de rutinas en el software *Matlab*[®] en el que está implementada la estimación por *MV*. La aplicación del método se hizo de acuerdo al trabajo propuesto por *Swevers et al.* [44]. A través de la rutina *fminunc* se obtuvo el valor del vector de parámetros dinámicos y de la trayectoria que minimizaba la función cuadrática K , que es función de los errores de las fuerzas y trayectoria realizadas por los actuadores del robot. Y finalmente, una vez realizada la estimación, se realizó lo que viene denominándose *Validación del Modelo*, en el que se discute si los parámetros identificados representan el robot 3-PRS, ajustándose lo máximo posible a la realidad. Esta validación se realizó con la propia trayectoria de estimación, quedando como trabajo futuro realizarla con una trayectoria diferente.

Por tanto, las principales conclusiones del presente trabajo se pueden resumir en:

- ◆ A través de la modelización del robot 3-PRS mediante el *Principio de Potencias Virtuales*, se ha puesto de manifiesto que es posible construir el modelo dinámico del robot y que sus ecuaciones dinámicas pueden escribirse de forma lineal con respecto de los parámetros dinámicos.
- ◆ En el modelado dinámico del robot 3-PRS, se ha llegado a la conclusión que algunos parámetros no influyen en la dinámica (no aparecen en las ecuaciones) y otros lo hacen como combinaciones lineales. Para hacer el sistema identificable se han calculado las expresiones simbólicas de los parámetros base y se han eliminado las columnas correspondientes de la matriz W para obtener una matriz W_b de rango máximo.
- ◆ Se puede decir, que el método de estimación por *MV*, no es el método más adecuado para la estimación de parámetros de un robot del tipo 3-PRS, pues este tipo de robots generan poco ruido en sus articulaciones y no es necesaria la utilización de un método tan sofisticado cuando el error en la medición de la posición de los *GDL* se ha con precisión. Sin embargo, en el péndulo doble que se estudió antes, sí que se pudo aplicar el método de *MV* a un problema en el que el error de medición de los *GDL* era significativo (al tratarse de un experimento computacional podía elegirse el nivel de error añadido a las medidas de posición).
- ◆ Se han observado que cuando se dan los cambios de signo de la velocidad de los actuadores hay un salto importante en la magnitud de las fuerzas, por lo que se deduce

que en los actuadores la fricción puede modelarse, al menos en parte, como una fricción constante de Coulomb.

7. LÍNEAS FUTURAS DE TRABAJO

En este apartado final se plantearán algunas de las líneas de futuro trabajo como continuación de este trabajo, como son las siguiente:

- ◆ El experimento realizado sobre el robot 3-PRS en este trabajo se ha basado en la parametrización de la trayectoria por series finitas de Fourier. En un futuro se podría realizar una parametrización mixta utilizando series de Fourier junto con Splines para ajustar de forma más precisa la trayectoria medida.
- ◆ Para la aplicación del método de MV , se ha empleado la rutina *fminunc* implementada en *Matlab*[®] como función para obtener el valor de los parámetros dinámicos y de la trayectoria que minimizasen la función K . Se establece en un futuro realizar dicha minimización de la función mediante métodos más elementales, como *Steepest Descent (SD)*, y posteriormente, analizar y comparar los resultados de la identificación obtenidos entre un método y otro.
- ◆ Como se ha visto, a pesar de que se ha procedido de manera correcta para la estimación de los parámetros dinámicos del robot 3-PRS, debido a la extensión del código no se ha sido capaz de estimarlos correctamente. Esta errónea estimación podría deberse a un error puntual en el código. Es por esto, que como futuro trabajo se establece corregir el código y obtener una estimación de los parámetros dinámicos que proporcione una precisa estimación de las fuerzas de los actuadores.
- ◆ Se establece en un futuro realizar la completa validación del modelo utilizando trayectoria que no hayan sido utilizadas en el proceso de estimación de los parámetros.

8. BIBLIOGRAFÍA

- [1] H. Abdellatif and B. Heimann, “*Experimental identification of the dynamics model of 6-dof parallel manipulators*”, *Robotica* (2009), pp. 329–348.
- [2] H. Abdellatif, B. Heimann, and M. Grotjahn, “*Statistical approach for biasfree identification of parallel manipulator affected by large measurement noise*”, 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville (Spain), IEEE, December 12–15 2005, pp. 3357–3362.
- [3] H. Abdellatif, B. Heimann, O. Hornung and M. Grotjahn, “*Identification and appropriate parametrization of parallel robot dynamic models by using estimation statistical properties*”, International Conference on Intelligent Robots and Systems (Edmonton, Canada), August 2005, pp. 157-162.
- [4] H. Alonso Moreno, “*Robots paralelos, conceptos y aplicaciones*”, en *Conf. CONAEE* (2012). [En línea]. Disponible en: <https://goo.gl/ZQB8SM>. [Accedido: 15-abr-2017]
- [5] J. Ángeles, “*Fundamentals of Robotic Mechanical Systems. Theory, Methods and Algorithms*”, Springer US, (2007).
- [6] R. J. Arinyo, “*The least squares method*”, Grup d’Informàtica a l’Enginyeria, Escola Tècnica Superior d’Enginyeria Industrial, Universitat Politècnica de Catalunya. [En línea]. Disponible en: <https://goo.gl/jl0gSA>. [Accedido: 15-may-2017]
- [7] B. Armstrong, “*On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics*”, *International Journal of Robotics Research* **8** (1989), no. 6, pp. 28-48.
- [8] C.G. Atkenson, C.H. An and J.M. Hollerbach, “*Estimation of inertial parameters of manipulator loads and links*”, *International Journal of Robotics Research* **5** (1986), no. 3, pp. 101-119.
- [9] G. Bermúdez, “*Robots móviles. Teoría, aplicaciones y experiencias*”, Universidad Distrital Francisco José de Caldas, Bogotá (Colombia), (2012).
- [10] C. L. Bottasso, A. Croce and S. Cacciola, “*Estimation of blade structural properties from experimental data*”, *Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano (Milano, Italy)* **16** (2012), pp. 501-518.
- [11] G. Calafiore, M. Indri and B. Bona, “*Robot dynamic calibration: optimal excitation trajectories and experimental parameter estimation*”, *Journal of Robotics Research* **10** (1991), no. 3, pp. 189-199.
- [12] K. Chen and D.G. Beale, “*A new symbolic method to determine base inertial parameters for general spatial mechanisms*”, *Design Engineering Technical Conferences and Computer and Information in Engineering Conference, ASME*, September-October 2002, pp. 731-735.

- [13] K. Chen, D.G. Beale and D. Wang, "A new method to determine the base inertial parameters of planar mechanisms", *Mechanism and Machine Theory* **37** (2002), pp. 971-984.
- [14] M. Díaz-Rodríguez, X. Iriarte, V. Mata, and J. Ros, "On the experiment design for direct dynamic parameter identification of parallel robots", *Advanced Robotics* **23** (2009), no. 3, pp. 329–348.
- [15] S. Garrido, "Identificación, Estimación y Control de Sistemas No-lineales mediante RGO", tesis doctoral, Univ. Carlos III, Leganés, 1999.
- [16] M. Gautier, "Numerical calculation of the base inertial parameters of robots", *Journal of Robotic Systems* **8** (1991), no. 4, pp. 485-506.
- [17] M. Gautier and W. Khalil, "A direct determination of minimum inertial parameters of robots", *Proceedings of the IEEE International Conference on Robotics and Automation*, 1988, pp. 1682-1687.
- [18] M. Ghodoussi and Y. Nakamura, "Principal base parameters of open and closed kinematic chains", *Proceedings of the IEEE International Conference of Robotics and Automation*, April 1991, pp. 84-89.
- [19] V. González, "Robots Industriales", [En línea]. Disponible en: <https://goo.gl/6CMgmZ>. [Accedido: 20-may-2017]
- [20] M. Grotjahn, B. Heimann and H. Abdellatif, "Identification of friction and rigid-body dynamics of parallel kinematic structures for model-based control", *Multibody System Dynamics* **11** (2004), pp. 273-294.
- [21] X. Iriarte, "Identificación de robots manipuladores: Reducción de modelos y diseño de experimentos", tesis doctoral, Universidad Pública de Navarra, Pamplona, (2010).
- [22] S.M. Kay, "Fundamentals of statistical signal processing: estimation theory", Prentice-Hall, Inc., Upper Saddle River, NJ, USA, (1993).
- [23] W. Khalil and F. Bennis, "Symbolic calculation of the base inertial parameters of closed-loop robots", *International Journal of Robotics Research* **14** (1995), no. 2, pp. 112-128.
- [24] W. Khalil and E. Dombre, "Modeling, identification and control of robots", Kogan Page Science, (2002).
- [25] W. Khalil and J. Kleinfinger, "A new geometric notation for open and closed-loop robots", *Proceedings of the IEEE International Conference on Robotics and Automation*, vol.3, (April 1986), pp. 1174-1179.
- [26] P.K. Khosla and T. Kanade, "Parameter identification of robot dynamics", *Proceedings of 24th Conference on Decision and Control*, 1985, pp. 1754-1760.

- [27] H. Mayeda, K. Yoshida and K. Osuka, “*Base parameters of manipulator dynamics models*”, Transactions of Robotics and Automation **6** (1990), no. 3, pp. 312-321.
- [28] L. A. Mejía, “*Metodología para la Identificación de Parámetros Dinámicos en Sistemas Mecánicos de Baja Movilidad: Aplicación a una Suspensión de Vehículo Automóvil*”, tesis doctoral, Univ. Politécnica de Valencia, Valencia, 2015.
- [29] NIST/SEMATECH, “*e-Handbook of Statistical Methods*”, [En línea]. Disponible en: <https://goo.gl/B2cYG0>. [Accedido: 17-may-2017]
- [30] M. M. Olsen, and H. G. Petersen, “*A New Method for Estimating Parameters of a Dynamic Robot Model*”, Transactions on Robotics and Automation **17** (2001), pp. 95-100.
- [31] M. M. Olsen, J. Swevers, and W. Verdonck, “*Maximum likelihood identification of a dynamic robot model: implementation issues*”, International Journal of Robotics Research **21** (2002), no. 2, pp. 89–96.
- [32] K.J. Park, “*Fourier-based optimal excitation trajectories for the dynamic identification of robots*”, Robotica **24** (2006), pp. 625-633.
- [33] S. Pértegas Díaz S, “*La distribución normal*”, Unidad de Epidemiología Clínica y Bioestadística (Complejo Hospitalario Juan Canalejo), A Coruña, 2001 [En línea]. Disponible en: <https://goo.gl/fbphFS>. [Accedido: 28-feb-2017]
- [34] C. Presse and M. Gautier, “*New criteria of exciting trajectories for robot identification*”, Proceedings of the IEEE International Conference on Robotics and Automation, May 1993, pp. 907-912.
- [35] B. Raucent, G. Campion, G. Bastin, J.C. Samin and P.Y. Willems, “*Identification of the barycentric parameters of robot manipulators from external measurements*”, Automatica **28** (1992), no. 5, pp. 1011-1016.
- [36] J.M. Rico, “*El Principio del Trabajo Virtual*”, Departamento de Ingeniería Mecánica, Universidad de Guanajuato, Guanajuato (México), (2006).
- [37] J.J. Rincón, “*Introducción a la Identificación de Sistemas Dinámicos*”, Facultad de Ingeniería Eléctrica, Universidad Michoacana de San Nicolás de Hidalgo, Morelia (México).
- [38] J.M. Rojo, “*Regresión lineal múltiple*”, Instituto de Economía y Geografía, Madrid, (2007).
- [39] J. Ros, “*Dinámica*”, Departamento de Ingeniería Mecánica, Energética, y de Materiales, Universidad Pública de Navarra, Pamplona, (2000). [En línea]. Disponible en: <https://goo.gl/QguLdK>. [Accedido: 4-may-2017]
- [40] O. Salgado, “*Síntesis, Análisis y Diseño de Manipuladores Paralelos de Baja Movilidad*”, Departamento de Ingeniería Mecánica, Universidad del País Vasco, Bilbao, (2008).

- [41] S. Sheu and M.W. Walker, "*Identifying the independent inertial parameter space of robot manipulators*", International Journal of Robotics Research **10** (1991), no. 6, pp. 668-683.
- [42] Y. Sun and J.M. Hollerbach, "*Observability index selection for robot calibration*", Proceedings of the IEEE International Conference on Robotics and Automation, May 19-23 (2008), pp. 831-836.
- [43] J. Swevers, C. Ganseman, J. De Schutter, and H. Van Brussel, "*Experimental robot identification using optimised periodic trajectories*", Mechanical Systems and Signal Processing **10** (1996), 561-577.
- [44] J. Swevers, C. Ganseman, D. B. Tükel, J. De Schutter, and H. Van Brussel, "*Optimal robot excitation and identification*", Transactions on Robotics and Automation **13** (1997), pp. 730-740.
- [45] A. Vivas, "*Robótica paralela: Aplicaciones industriales, modelado y control*", Departamento de Electrónica, Instrumentación y Control, Universidad del Cauca, Popayán (Colombia).

En Pamplona, a 23 de junio del 2017,

Fdo.

Julen Bacaicoa Díaz

Estudiante de Grado en Ingeniería en Tecnologías Industriales

9. ANEXOS

9.1. Código para la obtención ecuaciones dinámicas robot 3-PRS, mediante el software 3d_Mec

En el siguiente apartado se presenta el código correspondiente para la obtención de las ecuaciones dinámicas del robot 3-PRS a través del software *3d_mec*, en él, se definen las acciones y se caracterizan los torsesores, para finalmente, aplicar el *Principio de las Potencias Virtuales* y obtener las ecuaciones dinámicas.

9.1.1. Robot_3_prs.mec

```
// -----  
// -----  
// -----DETERMINATION OF DE DYNAMIC ECUATIONS-----  
//-----OF PARALLEL ROBOT 3 PRS-----  
// -----  
// -----  
  
// GEOMETRIC PARAMETERS  
// -----  
// LN distance between the vertices of the platform  
LN:=0.519818  
  
// LR length between the revolution and spheric joints  
LR:=0.54  
  
// LC1 first cylinder length  
LC1:=0.5  
// LC2 second cylinder length  
LC2:=0.5  
  
// CONSTANTS  
// -----  
// LB lower triangle side  
LB:=0.6662  
  
// Gravity constant  
g:=9.8  
  
//Diameter of the spheres  
WP:=0.05  
WC:=0.03  
  
// SQR3==2*COS(30°)  
SQR3:=1.73205080756888  
PI:=3.14159265358979  
  
// CORDINATES  
// -----  
// q1, q2, q3 are the height's of each prismatic actuator  
q1:=0.8  
D(q1):=dq1
```

```

D(dq1):=ddq1

q2:=0.8
D(q2):=dq2
D(dq2):=ddq2

q3:=0.8
D(q3):=dq3
D(dq3):=ddq3

// q4, q5, q6 are the angular coordinate of each bar
q4:=1.5
D(q4):=dq4
D(dq4):=ddq4

q5:=1.5
D(q5):=dq5
D(dq5):=ddq5

q6:=1.5
D(q6):=dq6
D(dq6):=ddq6

// x, y, z are the initial cordinates of the mobile platform
x:=0.0
D(x):=dx
D(dx):=ddx

y:=0.0
D(y):=dy
D(dy):=ddy

z:=0.0
D(z):=dz
D(dz):=ddz

// ji, ti, ri are the initial angular cordinates of the mobile
platform
ji:=0.0
D(ji):=dji
D(dji):=ddji

ti:=0.0
D(ti):=dti
D(dti):=ddti

ri:=0.0
D(ri):=dri
D(dri):=ddri

// ALPHAS
// -----
// A new base is built to determine de orientation of each
prismatic joint

```

```

// A1:=0+PI
// A2:=2*PI/3+PI
// A3:=-2*PI/3+PI

// In order not to appear numbers in the symbolic expressions,
these are the new definitions of the angles

BI:=2
HIRU:=3

A1:=0+PI
A2:=BI*PI/HIRU+PI
A3:=-BI*PI/HIRU+PI

// AUXILIARY BASES FIXED TO THE GROUND
// -----
BASE(B01,xyz,3,A1)
BASE(B02,xyz,3,A2)
BASE(B03,xyz,3,A3)

BASE(B1,B01,1,-q4)
BASE(B2,B02,1,-q5)
BASE(B3,B03,1,-q6)

// POSITION VECTORS, R1, R2 AND R3
// -----
POINT(R1,O,{[0,-LB/SQR3,0]}B01)
POINT(R2,O,{[0,-LB/SQR3,0]}B02)
POINT(R3,O,{[0,-LB/SQR3,0]}B03)

// REFERENCES FOR THE CHAIN'S FIRST LINK
// -----
REF(ref1,R1,B01)
REF(ref2,R2,B02)
REF(ref3,R3,B03)

// Visualization Point and orientation configuration
VISUAL(-16.0,23.0,5.2,0.0,0.0,0.6)

// DRAWING REFERENCES
// -----
DRAW_REFERENCE(ref1,0.5)
DRAW_REFERENCE(ref2,0.5)
DRAW_REFERENCE(ref3,0.5)

// DEFINITION OF THE END OF THE CHAIN'S SECOND LINK
// -----
POINT(Q1,R1,{[0,0,q1]}B01)
POINT(Q2,R2,{[0,0,q2]}B02)
POINT(Q3,R3,{[0,0,q3]}B03)

//DRAWING THE BAR 1 REFERENCE
// -----

```

```

REF(ref4,Q1,B1)
DRAW_REFERENCE(ref4,0.5)

//DRAWING THE ABSOLUTE REFERENCE
// -----
DRAW_REFERENCE(abs,0.5)

//DRAWING THE FIRST LINKS
// -----
DRAW_CYLINDER(R1,B01,WC,LC1,3,1,RGB(1,0.5,0))
DRAW_CYLINDER(R2,B02,WC,LC1,3,1,RGB(0,1,0.5))
DRAW_CYLINDER(R3,B03,WC,LC1,3,1,RGB(0.5,0,1))

// DRAWING THE SECOND LINK'S END
// -----
DRAW_SPHERE(Q1,B01,0.8*WP,RGB(0,1,0))
DRAW_SPHERE(Q2,B02,0.8*WP,RGB(0,1,0))
DRAW_SPHERE(Q3,B03,0.8*WP,RGB(0,1,0))

// MOBILE PLATFORM BASE (BP3)
// -----
BASE(BP1,xyz,1,ji)
BASE(BP2,BP1,2,ti)
BASE(BP3,BP2,3,ri)

// MOBILE PLATFORM POINTS
// -----
POINT(OP,O,{[x,y,z]}xyz)

POINT(P1,OP,{[-LN/BI,-LN/SQR3/BI,0]}BP3)
POINT(P2,OP,{[LN/BI,-LN/SQR3/BI,0]}BP3)
POINT(P3,OP,{[0,LN/SQR3,0]}BP3)

// DRAWING THE PLATFORM VERTICES
// -----
DRAW_SPHERE(P1,BP3,0.8*WP,RGB(1,0,0))
DRAW_SPHERE(P2,BP3,0.8*WP,RGB(0,1,0))
DRAW_SPHERE(P3,BP3,0.8*WP,RGB(0,0,1))

// DRAWING MOBILE PLATFORM REFERENCE
// -----
REF(refP,OP,BP3)
DRAW_REFERENCE(refP,0.5)

// DRAWING THE MOBILE PLATFORM
// -----
DRAW_CYLINDER_P(P1,P2,0.03,1,RGB(0.3,0.25,0.1))
DRAW_CYLINDER_P(P2,P3,0.03,1,RGB(0.3,0.25,0.1))
DRAW_CYLINDER_P(P3,P1,0.03,1,RGB(0.3,0.25,0.1))

// DEFINITION OF THE BEGINNING OF THE CHAIN'S SECOND LINK
// -----
POINT(S1,Q1,{[0,0,-LC2]}B01)
POINT(S2,Q2,{[0,0,-LC2]}B02)

```



```

POINT(S3,Q3,{[0,0,-LC2]}B03)

// DRAWING THE BEGINNING OF THE CHAIN'S SECOND LINK
// -----
DRAW_SPHERE(S1,B01,WP,RGB(1,1,0))
DRAW_SPHERE(S2,B02,WP,RGB(1,1,0))
DRAW_SPHERE(S3,B03,WP,RGB(1,1,0))

// DRAWING THE SECOND LINKS
// -----
DRAW_CYLINDER_P(Q1,S1,0.025,1,RGB(1,0,0))
DRAW_CYLINDER_P(Q2,S2,0.025,1,RGB(0,1,0))
DRAW_CYLINDER_P(Q3,S3,0.025,1,RGB(0,0,1))

//DEFINITION OF THE END OF REVOLUTION LINKS
// -----
POINT(T1,Q1,{[0,LR,0]}B1)
POINT(T2,Q2,{[0,LR,0]}B2)
POINT(T3,Q3,{[0,LR,0]}B3)

//DRAWING THE REVOLUTION LINK'S END
// -----
DRAW_SPHERE(T1,BP1,0.8*WP,RGB(1,0,0))
DRAW_SPHERE(T2,BP2,0.8*WP,RGB(0,1,0))
DRAW_SPHERE(T3,BP3,0.8*WP,RGB(0,0,1))

// DRAWING THE REVOLUTION LINKS
// -----
DRAW_CYLINDER_P(Q1,T1,0.025,1,RGB(0.5,0.5,0.5))
DRAW_CYLINDER_P(Q2,T2,0.025,1,RGB(0.5,0.5,0.5))
DRAW_CYLINDER_P(Q3,T3,0.025,1,RGB(0.5,0.5,0.5))

//SAVING AN IMAGE OF THE ROBOT 3-PRS
SAVE_GRAPHIC_OUTPUT("robot_3_prs_not_ready.jpg")

PAUSE

// -----
// -----
// -----GEOMETRIC LINKAGE EQUATIONS-----
// -----
// -----

NON_LINEAR_EQUATION_SYSTEM(q4,q5,q6,x,y,z,ji,ti,ri)

    {P(T1,P1)}xyz={ [0,0,0] }xyz
    {P(T2,P2)}xyz={ [0,0,0] }xyz
    {P(T3,P3)}xyz={ [0,0,0] }xyz

END

PRINT( {P(T1,P1)}xyz )
PRINT( {P(T2,P2)}xyz )
PRINT( {P(T3,P3)}xyz )

```

```

// -----
// -----
// -----KINETIC LINKAGE EQUATIONS-----
// -----
// -----

LINEAR_EQUATION_SYSTEM(dq4,dq5,dq6,dx,dy,dz,dji,dti,dri)

    D({P(T1,P1)}xyz)={ [0,0,0] }xyz
    D({P(T2,P2)}xyz)={ [0,0,0] }xyz
    D({P(T3,P3)}xyz)={ [0,0,0] }xyz

END

PRINT( D({P(T1,P1)}xyz) )
PRINT( D({P(T2,P2)}xyz) )
PRINT( D({P(T3,P3)}xyz) )

PRINT( D(D({P(T1,P1)}xyz)) )
PRINT( D(D({P(T2,P2)}xyz)) )
PRINT( D(D({P(T3,P3)}xyz)) )

//SAVING AN IMAGE OF THE ROBOT 3-PRS
SAVE_GRAPHIC_OUTPUT("robot_3_prs_ready.jpg")

PAUSE

// -----
// -----
// -----DYNAMICS -----
// -----
// -----

// -----INERTIA FORCES-----
// -----
// Mobile Platform
// -----

IPxx:=1.5
IPxy:=0
IPxz:=0
IPyy:=1.5
IPyz:=0
IPzz:=1.0

{I_OP(OP)}BP3:={ [[IPxx,IPxy,IPxz],[IPxy,IPyy,IPyz],[IPxz,IPyz,IP
zz]] }BP3
mPx:=0
mPy:=0
// Cuidado porque esto se substituiria simbolicamente
// mPy:=-LP/SQR3
mPz:=0

```

```

mP:=1

POINT(GP,OP,1/mP*{[mPx,mPy,mPz]}BP3)
FiP:=-mP*D(D(P(O,GP),abs),abs)
MiP_OP:=-D(I_OP(OP)*OM(abs,refP),abs)-
mP*P(OP,GP)^D(D(P(O,OP),abs),abs)

// -----
// Bar 1
// -----

IB1Fxx:=1
IB1Fxy:=0
IB1Fxz:=0
IB1Fyy:=1
IB1Fyz:=0
IB1Fzz:=1

{I_Q1(B1F)}B1:={[[IB1Fxx,IB1Fxy,IB1Fxz],[IB1Fxy,IB1Fyy,IB1Fyz],[
IB1Fxz,IB1Fyz,IB1Fzz]]}B1
mB1Fx:=0
mB1Fy:=+1
mB1Fz:=0
mB1F:=1

POINT(GB1F,Q1,1/mB1F*{[mB1Fx,mB1Fy,mB1Fz]}B1)
FiB1F:=-mB1F*D(D(P(O,GB1F),abs),abs)
MiB1F_Q1:=-D(I_Q1(B1F)*OM(abs,B1),abs)-
mB1F*P(Q1,GB1F)^D(D(P(O,Q1),abs),abs)

// -----
// Bar 2
// -----

IB2Fxx:=1
IB2Fxy:=0
IB2Fxz:=0
IB2Fyy:=1
IB2Fyz:=0
IB2Fzz:=1

{I_Q2(B2F)}B2:={[[IB2Fxx,IB2Fxy,IB2Fxz],[IB2Fxy,IB2Fyy,IB2Fyz],[
IB2Fxz,IB2Fyz,IB2Fzz]]}B2
mB2Fx:=0
mB2Fy:=+1
mB2Fz:=0
mB2F:=1

POINT(GB2F,Q2,1/mB2F*{[mB2Fx,mB2Fy,mB2Fz]}B2)
FiB2F:=-mB2F*D(D(P(O,GB2F),abs),abs)
MiB2F_Q2:=-D(I_Q2(B2F)*OM(abs,B2),abs)-
mB2F*P(Q2,GB2F)^D(D(P(O,Q2),abs),abs)

// -----

```

```

// Bar 3
// -----

IB3Fxx:=1
IB3Fxy:=0
IB3Fxz:=0
IB3Fyy:=1
IB3Fyz:=0
IB3Fzz:=1

{I_Q3(B3F)}B3:={[[IB3Fxx,IB3Fxy,IB3Fxz],[IB3Fxy,IB3Fyy,IB3Fyz],[
IB3Fxz,IB3Fyz,IB3Fzz]]}B3
mB3Fx:=0
mB3Fy:=+1
mB3Fz:=0
mB3F:=1

POINT(GB3F,Q3,1/mB3F*{[mB3Fx,mB3Fy,mB3Fz]}B3)
FiB3F:=-mB3F*D(D(P(O,GB3F),abs),abs)
MiB3F_Q3:=-D(I_Q3(B3F)*OM(abs,B3),abs)-
mB3F*P(Q3,GB3F)^D(D(P(O,Q3),abs),abs)

// -----
// -----
// -----VELOCITIES CALCULATION-----
// -----
// -----

// Velocity of the center of gravity of the platform
Vabs(OP):=D(P(O,OP),abs)
Vabs(GP):=D(P(O,GP),abs)

// Velocity of point 1 of the platform and G of the Bar 1
Vabs(P1):=D(P(O,P1),abs)
Vabs(GB1F):=D(P(O,GB1F),abs)

// Velocity of point 2 of the platform and G of the Bar 2
Vabs(P2):=D(P(O,P2),abs)
Vabs(GB2F):=D(P(O,GB2F),abs)

// Velocity of point 3 of the platform and G of the Bar 3
Vabs(P3):=D(P(O,P3),abs)
Vabs(GB3F):=D(P(O,GB3F),abs)

// Velocity of the point Q1-Q3
Vabs(Q1):=D(P(O,Q1),abs)
Vabs(Q2):=D(P(O,Q2),abs)
Vabs(Q3):=D(P(O,Q3),abs)

// Velocity of the point T1-T3 as belonging to the Second part
of the actuator
Vabs(T1):=D(P(O,T1),abs)
Vabs(T2):=D(P(O,T2),abs)
Vabs(T3):=D(P(O,T3),abs)

```

```

//Velocity of Bar i with respect to the mobile platform
PRINT({OM(BP3,B1)}xyz)
PRINT({OM(BP3,B2)}xyz)
PRINT({OM(BP3,B3)}xyz)

//Velocity of the mobile platform with respect to Bar i
PRINT({OM(B1,BP3)}xyz)
PRINT({OM(B2,BP3)}xyz)
PRINT({OM(B3,BP3)}xyz)

PAUSE

// -----
// -----
// -----JOINTS WRENCHES CALCULATION-----
// -----
// -----

// -----
// ----- KINEMATIC CHAIN 1 -----
// -----

// Joint P1 Wrench (ACTUATING ON THE PLATFORM)
FS1x:=0
FS1y:=0
FS1z:=0
{FS1}B1:={ [FS1x,FS1y,FS1z]}B1

// -----
// ----- KINEMATIC CHAIN 2 -----
// -----

// Joint P2 Wrench (ACTUATING ON THE PLATFORM)
FS2x:=0
FS2y:=0
FS2z:=0
{FS2}B2:={ [FS2x,FS2y,FS2z]}B2

// -----
// ----- KINEMATIC CHAIN 3 -----
// -----

// Joint P3 Wrench (ACTUATING ON THE PLATFORM)
FS3x:=0
FS3y:=0
FS3z:=0
{FS3}B3:={ [FS3x,FS3y,FS3z]}B3

// -----
// -----
// -----VISCOUS AND COULOMB FRICTION PARAMETERS-----
// -----
// -----

```

```

// VISCOUS FRICTION COEFFICIENTS
// -----
// Viscous Prismatic friction coefficients
VISP1:=0
VISP2:=0
VISP3:=0

// -----
// Viscous Revolution friction coefficients
VISR1:=0
VISR2:=0
VISR3:=0

// -----
// Viscous Spherical friction coefficients
VISS1:=0
VISS2:=0
VISS3:=0

// COSTANT COULOMB FRICTION COEFFICIENTS
// -----
// Coulomb Prismatic friction coefficients
FCCP1z:=0
FCCP2z:=0
FCCP3z:=0

// -----
// Coulomb Revolution friction coefficients
MCCR1z:=0
MCCR2z:=0
MCCR3z:=0

// -----
// Coulomb Spherical friction coefficients
MCCS1:=0
MCCS2:=0
MCCS3:=0

// -----
// -----
// -----FRICITION WRENCH CALCULATION-----
// -----
// -----

//DEFINITION OF VARIABLE sign_ (sign of the corresponding
generalized coordinate derivate)
sign_dq1:=0
sign_dq2:=0
sign_dq3:=0
sign_dq4:=0
sign_dq5:=0
sign_dq6:=0

```

```

// The variable omega is the MAX of {OM(BP3,B1)}xyz and
TOL=0.001 (Matlab). Omega should not be equal to zero
omega_bar1:=1
omega_bar2:=1
omega_bar3:=1

// -----
// ----- KINEMATIC CHAIN 1 -----
// -----

// Viscous Prismatic Joint 1 Torsor VP1
//Active Actuator 1 First link on Actuator 1 Second link
{FVP1}B01:={ [0,0,-VISP1*dq1]}B01

// -----

// Coulomb Prismatic Joint 1 Torsor CP1
//Active Actuator 1 First link on Actuator 1 Second link
{FCP1}B01:={ [0,0,-FCCP1z*sign_dq1]}B01

//-----

//Viscous Revolution Joint 1 Torsor VR1
//Active Actuator 1 on Bar 1
{MVR1_Q1}B1:={ [VISR1*dq4,0,0]}B1

//-----

// Coulomb Revolution Joint 1 Torsor CR1
//Active Actuator 1 on Bar 1
{MCR1_Q1}B1:={ [MCCR1z*sign_dq4,0,0]}B1

//-----

//Viscous Spherical Joint 1 Torsor VS1
//Active Bar 1 on Mobile Platform
//MVS1_T1:=-VISS1*{OM(BP3,B1)}xyz

// action exerted on mobile platform
MVS1_T1:=-VISS1*{OM(B1,BP3)}xyz

//-----

// Coulomb Spherical Joint 1 Torsor CS1
//Active Bar 1 on Mobile Platform
//MCS1_T1:=(-MCCS1*{OM(BP3,B1)}xyz)/omega_bar1

// Action exerted on mobile platform
MCS1_T1:=(-MCCS1*{OM(B1,BP3)}xyz)/omega_bar1

// -----
// ----- KINEMATIC CHAIN 2 -----
// -----

```

```

// Viscous Prismatic Joint 2 Torsor VP2
//Active Actuator 2 First link on Actuator 2 Second link
{FVP2}B02:={[0,0, - VISP2*dq2]}B02

//-----

// Coulomb Prismatic Joint 2 Torsor CP2
//Active Actuator 2 First link on Actuator 2 Second link
{FCP2}B02:={[0,0, - FCCP2z*sign_dq2]}B02

//-----

//Viscous Revolution Joint 2 Torsor VR2
//Active Actuator 2 on Bar 2
{MVR2_Q2}B2:={[VISR2*dq5,0,0]}B2

//-----

// Coulomb Revolution Joint 2 Torsor CR2
//Active Actuator 2 on Bar 2
{MCR2_Q2}B2:={[MCCR2z*sign_dq5,0,0]}B2

//-----

//Viscous Spherical Joint 2 Torsor VS2
//Active Bar 2 on Mobile Platform
MVS2_T2:=-VISS2*{OM(BP3,B2)}xyz

//-----

// Coulomb Spherical Joint 2 Torsor CS2
//Active Bar 2 on Mobile Platform
//MCS2_T2:=-MCCS2*({OM(BP3,B2)}xyz)/omega_bar2

// Action exerted on mobile platform
MCS2_T2:=(-MCCS2*{OM(B2,BP3)}xyz)/omega_bar2

// -----
// ----- KINEMATIC CHAIN 3 -----
// -----

// Viscous Prismatic Joint 3 Torsor VP3
//Active Actuator 3 First link on Actuator 3 Second link
{FVP3}B03:={[0,0, - VISP3*dq3]}B03

//-----

// Coulomb Prismatic Joint 3 Torsor CP3
//Active Actuator 3 First link on Actuator 3 Second link
{FCP3}B03:={[0,0, - FCCP3z*sign_dq3]}B03

//-----

```



```

//Viscous Revolution Joint 3 Torsor VR3
//Active Actuator 3 on Bar 3
{MVR3_Q3}B3:={ [VISR3*dq6,0,0] }B3

//-----

// Coulomb Revolution Joint 3 Torsor CR3
//Active Actuator 3 on Bar 3
{MCR3_Q3}B3:={ [MCCR3z*sign_dq6,0,0] }B3

//-----

//Viscous Spherical Joint 3 Torsor VS3
//Active Bar 3 on Mobile Platform
MVS3_T3:=-VISS3*{OM(BP3,B3)}xyz

//-----

// Coulomb Spherical Joint 3 Torsor CS3
//Active Bar 3 on Mobile Platform
//MCS3_T3:=-MCCS3*({OM(BP3,B3)}xyz)/omega_bar3

// Action exerted on mobile platform
MCS3_T3:=(-MCCS3*{OM(B3,BP3)}xyz)/omega_bar3

// -----
// -----
// -----ACTUATORS WRENCHES-----
// -----
// -----

// Actuator FORCE1 on actuator 1
FORCE1z:=0
{FORCE1}B01:={ [0,0,FORCE1z] }B01

// Actuator FORCE2 on actuator 2
FORCE2z:=0
{FORCE2}B02:={ [0,0,FORCE2z] }B02

// Actuator FORCE3 in actuator 3
FORCE3z:=0
{FORCE3}B03:={ [0,0,FORCE3z] }B03

PAUSE

// -----
// DYNAMIC EQUATIONS. DERIVATION WITH RESPECT TO dq1
// -----

//INERTIA WRENCHES OF MOBILE PLATFORM AND 3 BARS, GRAVITY
WRENCHES OF MOBILE PLATFORM AND 3 BARS AND 3 ACTUATOR FORCES

```

```

PRINT(+{FiP}xyz*D({Vabs(OP)}xyz,dq1)+{MiP_OP}xyz*D({OM(abs,BP3)}
xyz,dq1)+{FiB1F}xyz*D({Vabs(Q1)}xyz,dq1)+{MiB1F_Q1}xyz*D({OM(abs
,B1)}xyz,dq1)+{FiB2F}xyz*D({Vabs(Q2)}xyz,dq1)+{MiB2F_Q2}xyz*D({O
M(abs,B2)}xyz,dq1)+{FiB3F}xyz*D({Vabs(Q3)}xyz,dq1)+{MiB3F_Q3}xyz
*D({OM(abs,B3)}xyz,dq1)+{[0,0,-
mP*g]}xyz*D({Vabs(GP)}xyz,dq1)+{[0,0,-
mB1F*g]}xyz*D({Vabs(GB1F)}xyz,dq1)+{[0,0,-
mB2F*g]}xyz*D({Vabs(GB2F)}xyz,dq1)+{[0,0,-
mB3F*g]}xyz*D({Vabs(GB3F)}xyz,dq1)+{FORCE1}B01*D({Vabs(Q1)}B01,d
q1)+{FORCE2}B02*D({Vabs(Q2)}B02,dq1)+{FORCE3}B03*D({Vabs(Q3)}B03
,dq1) )

```

```

//KINLAGE WRENCHES BETWEEN MOBILE PLATFORM AND 3 BARS, AND
PRISMATIC, REVOLUTION AND SPHERICAL FRICTION WRENCHES (VISCOUS
AND COULOMB FRICTIONS)

```

```

PRINT(+{FS1}B1*D({Vabs(P1)}B1,dq1)-
{FS1}B1*D({Vabs(T1)}B1,dq1)+{FVP1}B01*D({Vabs(Q1)}B01,dq1)+{FCP1
}B01*D({Vabs(Q1)}B01,dq1)+{MVR1_Q1}B1*D({OM(abs,B1)}B1,dq1)+{MCR
1_Q1}B1*D({OM(abs,B1)}B1,dq1)+{MVS1_T1}B1*D({OM(abs,BP3)}B1,dq1)
-
{MVS1_T1}B1*D({OM(abs,B1)}B1,dq1)+{MCS1_T1}B1*D({OM(abs,BP3)}B1,
dq1)-{MCS1_T1}B1*D({OM(abs,B1)}B1,dq1))

```

```

PRINT(+{FS2}B2*D({Vabs(P2)}B2,dq1)-
{FS2}B2*D({Vabs(T2)}B2,dq1)+{FVP2}B02*D({Vabs(Q2)}B02,dq1)+{FCP2
}B02*D({Vabs(Q2)}B02,dq1)+{MVR2_Q2}B2*D({OM(abs,B2)}B2,dq1)+{MCR
2_Q2}B2*D({OM(abs,B2)}B2,dq1)+{MVS2_T2}B2*D({OM(abs,BP3)}B2,dq1)
-
{MVS2_T2}B2*D({OM(abs,B2)}B2,dq1)+{MCS2_T2}B2*D({OM(abs,BP3)}B2,
dq1)-{MCS2_T2}B2*D({OM(abs,B2)}B2,dq1))

```

```

PRINT(+{FS3}B3*D({Vabs(P3)}B3,dq1)-
{FS3}B3*D({Vabs(T3)}B3,dq1)+{FVP3}B03*D({Vabs(Q3)}B03,dq1)+{FCP3
}B03*D({Vabs(Q3)}B03,dq1)+{MVR3_Q3}B3*D({OM(abs,B3)}B3,dq1)+{MCR
3_Q3}B3*D({OM(abs,B3)}B3,dq1)+{MVS3_T3}B3*D({OM(abs,BP3)}B3,dq1)
-
{MVS3_T3}B3*D({OM(abs,B3)}B3,dq1)+{MCS3_T3}B3*D({OM(abs,BP3)}B3,
dq1)-{MCS3_T3}B3*D({OM(abs,B3)}B3,dq1))

```

```

// -----
// DYNAMIC EQUATIONS. DERIVATION WITH RESPECT TO dq2
// -----

```

```

PRINT(+{FiP}xyz*D({Vabs(OP)}xyz,dq2)+{MiP_OP}xyz*D({OM(abs,BP3)}
xyz,dq2)+{FiB1F}xyz*D({Vabs(Q1)}xyz,dq2)+{MiB1F_Q1}xyz*D({OM(abs
,B1)}xyz,dq2)+{FiB2F}xyz*D({Vabs(Q2)}xyz,dq2)+{MiB2F_Q2}xyz*D({O
M(abs,B2)}xyz,dq2)+{FiB3F}xyz*D({Vabs(Q3)}xyz,dq2)+{MiB3F_Q3}xyz
*D({OM(abs,B3)}xyz,dq2)+{[0,0,-
mP*g]}xyz*D({Vabs(GP)}xyz,dq2)+{[0,0,-
mB1F*g]}xyz*D({Vabs(GB1F)}xyz,dq2)+{[0,0,-
mB2F*g]}xyz*D({Vabs(GB2F)}xyz,dq2)+{[0,0,-
mB3F*g]}xyz*D({Vabs(GB3F)}xyz,dq2)+{FORCE1}B01*D({Vabs(Q1)}B01,d

```

q2) + {FORCE2} B02 * D ({Vabs (Q2) } B02, dq2) + {FORCE3} B03 * D ({Vabs (Q3) } B03, dq2))

PRINT (+ {FS1} B1 * D ({Vabs (P1) } B1, dq2) -
 {FS1} B1 * D ({Vabs (T1) } B1, dq2) + {FVP1} B01 * D ({Vabs (Q1) } B01, dq2) + {FCP1} B01 * D ({Vabs (Q1) } B01, dq2) + {MVR1_Q1} B1 * D ({OM (abs, B1) } B1, dq2) + {MCR1_Q1} B1 * D ({OM (abs, B1) } B1, dq2) + {MVS1_T1} B1 * D ({OM (abs, BP3) } B1, dq2) -
 {MVS1_T1} B1 * D ({OM (abs, B1) } B1, dq2) + {MCS1_T1} B1 * D ({OM (abs, BP3) } B1, dq2) - {MCS1_T1} B1 * D ({OM (abs, B1) } B1, dq2))

PRINT (+ {FS2} B2 * D ({Vabs (P2) } B2, dq2) -
 {FS2} B2 * D ({Vabs (T2) } B2, dq2) + {FVP2} B02 * D ({Vabs (Q2) } B02, dq2) + {FCP2} B02 * D ({Vabs (Q2) } B02, dq2) + {MVR2_Q2} B2 * D ({OM (abs, B2) } B2, dq2) + {MCR2_Q2} B2 * D ({OM (abs, B2) } B2, dq2) + {MVS2_T2} B2 * D ({OM (abs, BP3) } B2, dq2) -
 {MVS2_T2} B2 * D ({OM (abs, B2) } B2, dq2) + {MCS2_T2} B2 * D ({OM (abs, BP3) } B2, dq2) - {MCS2_T2} B2 * D ({OM (abs, B2) } B2, dq2))

PRINT (+ {FS3} B3 * D ({Vabs (P3) } B3, dq2) -
 {FS3} B3 * D ({Vabs (T3) } B3, dq2) + {FVP3} B03 * D ({Vabs (Q3) } B03, dq2) + {FCP3} B03 * D ({Vabs (Q3) } B03, dq2) + {MVR3_Q3} B3 * D ({OM (abs, B3) } B3, dq2) + {MCR3_Q3} B3 * D ({OM (abs, B3) } B3, dq2) + {MVS3_T3} B3 * D ({OM (abs, BP3) } B3, dq2) -
 {MVS3_T3} B3 * D ({OM (abs, B3) } B3, dq2) + {MCS3_T3} B3 * D ({OM (abs, BP3) } B3, dq2) - {MCS3_T3} B3 * D ({OM (abs, B3) } B3, dq2))

// -----
 // DYNAMIC EQUATIONS. DERIVATION WITH RESPECT TO dq3
 // -----

PRINT (+ {FiP} xyz * D ({Vabs (OP) } xyz, dq3) + {MiP_OP} xyz * D ({OM (abs, BP3) } xyz, dq3) + {FiB1F} xyz * D ({Vabs (Q1) } xyz, dq3) + {MiB1F_Q1} xyz * D ({OM (abs, B1) } xyz, dq3) + {FiB2F} xyz * D ({Vabs (Q2) } xyz, dq3) + {MiB2F_Q2} xyz * D ({OM (abs, B2) } xyz, dq3) + {FiB3F} xyz * D ({Vabs (Q3) } xyz, dq3) + {MiB3F_Q3} xyz * D ({OM (abs, B3) } xyz, dq3) + { [0, 0, -
 mP*g] } xyz * D ({Vabs (GP) } xyz, dq3) + { [0, 0, -
 mB1F*g] } xyz * D ({Vabs (GB1F) } xyz, dq3) + { [0, 0, -
 mB2F*g] } xyz * D ({Vabs (GB2F) } xyz, dq3) + { [0, 0, -
 mB3F*g] } xyz * D ({Vabs (GB3F) } xyz, dq3) + {FORCE1} B01 * D ({Vabs (Q1) } B01, dq3) + {FORCE2} B02 * D ({Vabs (Q2) } B02, dq3) + {FORCE3} B03 * D ({Vabs (Q3) } B03, dq3))

PRINT (+ {FS1} B1 * D ({Vabs (P1) } B1, dq3) -
 {FS1} B1 * D ({Vabs (T1) } B1, dq3) + {FVP1} B01 * D ({Vabs (Q1) } B01, dq3) + {FCP1} B01 * D ({Vabs (Q1) } B01, dq3) + {MVR1_Q1} B1 * D ({OM (abs, B1) } B1, dq3) + {MCR1_Q1} B1 * D ({OM (abs, B1) } B1, dq3) + {MVS1_T1} B1 * D ({OM (abs, BP3) } B1, dq3) -
 {MVS1_T1} B1 * D ({OM (abs, B1) } B1, dq3) + {MCS1_T1} B1 * D ({OM (abs, BP3) } B1, dq3) - {MCS1_T1} B1 * D ({OM (abs, B1) } B1, dq3))

PRINT (+ {FS2} B2 * D ({Vabs (P2) } B2, dq3) -
 {FS2} B2 * D ({Vabs (T2) } B2, dq3) + {FVP2} B02 * D ({Vabs (Q2) } B02, dq3) + {FCP2} B02 * D ({Vabs (Q2) } B02, dq3) + {MVR2_Q2} B2 * D ({OM (abs, B2) } B2, dq3) + {MCR2_Q2} B2 * D ({OM (abs, B2) } B2, dq3) + {MVS2_T2} B2 * D ({OM (abs, BP3) } B2, dq3) -
 {MVS2_T2} B2 * D ({OM (abs, B2) } B2, dq3) + {MCS2_T2} B2 * D ({OM (abs, BP3) } B2, dq3) - {MCS2_T2} B2 * D ({OM (abs, B2) } B2, dq3))

```

2_Q2}B2*D({OM(abs,B2)}B2,dq3)+{MVS2_T2}B2*D({OM(abs,BP3)}B2,dq3)
-
{MVS2_T2}B2*D({OM(abs,B2)}B2,dq3)+{MCS2_T2}B2*D({OM(abs,BP3)}B2,
dq3)-{MCS2_T2}B2*D({OM(abs,B2)}B2,dq3))

PRINT(+{FS3}B3*D({Vabs(P3)}B3,dq3)-
{FS3}B3*D({Vabs(T3)}B3,dq3)+{FVP3}B03*D({Vabs(Q3)}B03,dq3)+{FCP3
}B03*D({Vabs(Q3)}B03,dq3)+{MVR3_Q3}B3*D({OM(abs,B3)}B3,dq3)+{MCR
3_Q3}B3*D({OM(abs,B3)}B3,dq3)+{MVS3_T3}B3*D({OM(abs,BP3)}B3,dq3)
-
{MVS3_T3}B3*D({OM(abs,B3)}B3,dq3)+{MCS3_T3}B3*D({OM(abs,BP3)}B3,
dq3)-{MCS3_T3}B3*D({OM(abs,B3)}B3,dq3))

// -----
// DYNAMIC EQUATIONS. DERIVATION WITH RESPECT TO dq4
// -----

PRINT(+{FiP}xyz*D({Vabs(OP)}xyz,dq4)+{MiP_OP}xyz*D({OM(abs,BP3)}
xyz,dq4)+{FiB1F}xyz*D({Vabs(Q1)}xyz,dq4)+{MiB1F_Q1}xyz*D({OM(abs
,B1)}xyz,dq4)+{FiB2F}xyz*D({Vabs(Q2)}xyz,dq4)+{MiB2F_Q2}xyz*D({O
M(abs,B2)}xyz,dq4)+{FiB3F}xyz*D({Vabs(Q3)}xyz,dq4)+{MiB3F_Q3}xyz
*D({OM(abs,B3)}xyz,dq4)+{ [0,0,-
mP*g]}xyz*D({Vabs(GP)}xyz,dq4)+{ [0,0,-
mB1F*g]}xyz*D({Vabs(GB1F)}xyz,dq4)+{ [0,0,-
mB2F*g]}xyz*D({Vabs(GB2F)}xyz,dq4)+{ [0,0,-
mB3F*g]}xyz*D({Vabs(GB3F)}xyz,dq4)+{FORCE1}B01*D({Vabs(Q1)}B01,d
q4)+{FORCE2}B02*D({Vabs(Q2)}B02,dq4)+{FORCE3}B03*D({Vabs(Q3)}B03
,dq4) )

PRINT(+{FS1}B1*D({Vabs(P1)}B1,dq4)-
{FS1}B1*D({Vabs(T1)}B1,dq4)+{FVP1}B01*D({Vabs(Q1)}B01,dq4)+{FCP1
}B01*D({Vabs(Q1)}B01,dq4)+{MVR1_Q1}B1*D({OM(abs,B1)}B1,dq4)+{MCR
1_Q1}B1*D({OM(abs,B1)}B1,dq4)+{MVS1_T1}B1*D({OM(abs,BP3)}B1,dq4)
-
{MVS1_T1}B1*D({OM(abs,B1)}B1,dq4)+{MCS1_T1}B1*D({OM(abs,BP3)}B1,
dq4)-{MCS1_T1}B1*D({OM(abs,B1)}B1,dq4))

PRINT(+{FS2}B2*D({Vabs(P2)}B2,dq4)-
{FS2}B2*D({Vabs(T2)}B2,dq4)+{FVP2}B02*D({Vabs(Q2)}B02,dq4)+{FCP2
}B02*D({Vabs(Q2)}B02,dq4)+{MVR2_Q2}B2*D({OM(abs,B2)}B2,dq4)+{MCR
2_Q2}B2*D({OM(abs,B2)}B2,dq4)+{MVS2_T2}B2*D({OM(abs,BP3)}B2,dq4)
-
{MVS2_T2}B2*D({OM(abs,B2)}B2,dq4)+{MCS2_T2}B2*D({OM(abs,BP3)}B2,
dq4)-{MCS2_T2}B2*D({OM(abs,B2)}B2,dq4))

PRINT(+{FS3}B3*D({Vabs(P3)}B3,dq4)-
{FS3}B3*D({Vabs(T3)}B3,dq4)+{FVP3}B03*D({Vabs(Q3)}B03,dq4)+{FCP3
}B03*D({Vabs(Q3)}B03,dq4)+{MVR3_Q3}B3*D({OM(abs,B3)}B3,dq4)+{MCR
3_Q3}B3*D({OM(abs,B3)}B3,dq4)+{MVS3_T3}B3*D({OM(abs,BP3)}B3,dq4)
-
{MVS3_T3}B3*D({OM(abs,B3)}B3,dq4)+{MCS3_T3}B3*D({OM(abs,BP3)}B3,
dq4)-{MCS3_T3}B3*D({OM(abs,B3)}B3,dq4))

// -----

```

```

// DYNAMIC EQUATIONS. DERIVATION WITH RESPECT TO dq5
// -----

PRINT(+{FiP}xyz*D({Vabs(OP)}xyz,dq5)+{MiP_OP}xyz*D({OM(abs,BP3)}
xyz,dq5)+{FiB1F}xyz*D({Vabs(Q1)}xyz,dq5)+{MiB1F_Q1}xyz*D({OM(abs
,B1)}xyz,dq5)+{FiB2F}xyz*D({Vabs(Q2)}xyz,dq5)+{MiB2F_Q2}xyz*D({O
M(abs,B2)}xyz,dq5)+{FiB3F}xyz*D({Vabs(Q3)}xyz,dq5)+{MiB3F_Q3}xyz
*D({OM(abs,B3)}xyz,dq5)+{[0,0,-
mP*g]}xyz*D({Vabs(GP)}xyz,dq5)+{[0,0,-
mB1F*g]}xyz*D({Vabs(GB1F)}xyz,dq5)+{[0,0,-
mB2F*g]}xyz*D({Vabs(GB2F)}xyz,dq5)+{[0,0,-
mB3F*g]}xyz*D({Vabs(GB3F)}xyz,dq5)+{FORCE1}B01*D({Vabs(Q1)}B01,d
q5)+{FORCE2}B02*D({Vabs(Q2)}B02,dq5)+{FORCE3}B03*D({Vabs(Q3)}B03
,dq5) )

PRINT(+{FS1}B1*D({Vabs(P1)}B1,dq5)-
{FS1}B1*D({Vabs(T1)}B1,dq5)+{FVP1}B01*D({Vabs(Q1)}B01,dq5)+{FCP1
}B01*D({Vabs(Q1)}B01,dq5)+{MVR1_Q1}B1*D({OM(abs,B1)}B1,dq5)+{MCR
1_Q1}B1*D({OM(abs,B1)}B1,dq5)+{MVS1_T1}B1*D({OM(abs,BP3)}B1,dq5)
-
{MVS1_T1}B1*D({OM(abs,B1)}B1,dq5)+{MCS1_T1}B1*D({OM(abs,BP3)}B1,
dq5)-{MCS1_T1}B1*D({OM(abs,B1)}B1,dq5) )

PRINT(+{FS2}B2*D({Vabs(P2)}B2,dq5)-
{FS2}B2*D({Vabs(T2)}B2,dq5)+{FVP2}B02*D({Vabs(Q2)}B02,dq5)+{FCP2
}B02*D({Vabs(Q2)}B02,dq5)+{MVR2_Q2}B2*D({OM(abs,B2)}B2,dq5)+{MCR
2_Q2}B2*D({OM(abs,B2)}B2,dq5)+{MVS2_T2}B2*D({OM(abs,BP3)}B2,dq5)
-
{MVS2_T2}B2*D({OM(abs,B2)}B2,dq5)+{MCS2_T2}B2*D({OM(abs,BP3)}B2,
dq5)-{MCS2_T2}B2*D({OM(abs,B2)}B2,dq5) )

PRINT(+{FS3}B3*D({Vabs(P3)}B3,dq5)-
{FS3}B3*D({Vabs(T3)}B3,dq5)+{FVP3}B03*D({Vabs(Q3)}B03,dq5)+{FCP3
}B03*D({Vabs(Q3)}B03,dq5)+{MVR3_Q3}B3*D({OM(abs,B3)}B3,dq5)+{MCR
3_Q3}B3*D({OM(abs,B3)}B3,dq5)+{MVS3_T3}B3*D({OM(abs,BP3)}B3,dq5)
-
{MVS3_T3}B3*D({OM(abs,B3)}B3,dq5)+{MCS3_T3}B3*D({OM(abs,BP3)}B3,
dq5)-{MCS3_T3}B3*D({OM(abs,B3)}B3,dq5) )

// -----
// DYNAMIC EQUATIONS. DERIVATION WITH RESPECT TO dq6
// -----

PRINT(+{FiP}xyz*D({Vabs(OP)}xyz,dq6)+{MiP_OP}xyz*D({OM(abs,BP3)}
xyz,dq6)+{FiB1F}xyz*D({Vabs(Q1)}xyz,dq6)+{MiB1F_Q1}xyz*D({OM(abs
,B1)}xyz,dq6)+{FiB2F}xyz*D({Vabs(Q2)}xyz,dq6)+{MiB2F_Q2}xyz*D({O
M(abs,B2)}xyz,dq6)+{FiB3F}xyz*D({Vabs(Q3)}xyz,dq6)+{MiB3F_Q3}xyz
*D({OM(abs,B3)}xyz,dq6)+{[0,0,-
mP*g]}xyz*D({Vabs(GP)}xyz,dq6)+{[0,0,-
mB1F*g]}xyz*D({Vabs(GB1F)}xyz,dq6)+{[0,0,-
mB2F*g]}xyz*D({Vabs(GB2F)}xyz,dq6)+{[0,0,-
mB3F*g]}xyz*D({Vabs(GB3F)}xyz,dq6)+{FORCE1}B01*D({Vabs(Q1)}B01,d
q6)+{FORCE2}B02*D({Vabs(Q2)}B02,dq6)+{FORCE3}B03*D({Vabs(Q3)}B03
,dq6) )

```

```

PRINT(+{FS1}B1*D({Vabs(P1)}B1,dq6) -
{FS1}B1*D({Vabs(T1)}B1,dq6)+{FVP1}B01*D({Vabs(Q1)}B01,dq6)+{FCP1}
B01*D({Vabs(Q1)}B01,dq6)+{MVR1_Q1}B1*D({OM(abs,B1)}B1,dq6)+{MCR
1_Q1}B1*D({OM(abs,B1)}B1,dq6)+{MVS1_T1}B1*D({OM(abs,BP3)}B1,dq6)
-
{MVS1_T1}B1*D({OM(abs,B1)}B1,dq6)+{MCS1_T1}B1*D({OM(abs,BP3)}B1,
dq6)-{MCS1_T1}B1*D({OM(abs,B1)}B1,dq6))

```

```

PRINT(+{FS2}B2*D({Vabs(P2)}B2,dq6) -
{FS2}B2*D({Vabs(T2)}B2,dq6)+{FVP2}B02*D({Vabs(Q2)}B02,dq6)+{FCP2}
B02*D({Vabs(Q2)}B02,dq6)+{MVR2_Q2}B2*D({OM(abs,B2)}B2,dq6)+{MCR
2_Q2}B2*D({OM(abs,B2)}B2,dq6)+{MVS2_T2}B2*D({OM(abs,BP3)}B2,dq6)
-
{MVS2_T2}B2*D({OM(abs,B2)}B2,dq6)+{MCS2_T2}B2*D({OM(abs,BP3)}B2,
dq6)-{MCS2_T2}B2*D({OM(abs,B2)}B2,dq6))

```

```

PRINT(+{FS3}B3*D({Vabs(P3)}B3,dq6) -
{FS3}B3*D({Vabs(T3)}B3,dq6)+{FVP3}B03*D({Vabs(Q3)}B03,dq6)+{FCP3}
B03*D({Vabs(Q3)}B03,dq6)+{MVR3_Q3}B3*D({OM(abs,B3)}B3,dq6)+{MCR
3_Q3}B3*D({OM(abs,B3)}B3,dq6)+{MVS3_T3}B3*D({OM(abs,BP3)}B3,dq6)
-
{MVS3_T3}B3*D({OM(abs,B3)}B3,dq6)+{MCS3_T3}B3*D({OM(abs,BP3)}B3,
dq6)-{MCS3_T3}B3*D({OM(abs,B3)}B3,dq6))

```

```

// -----
// DYNAMIC EQUATIONS. DERIVATION WITH RESPECT TO dx
// -----

```

```

PRINT(+{FiP}xyz*D({Vabs(OP)}xyz,dx)+{MiP_OP}xyz*D({OM(abs,BP3)}x
yz,dx)+{FiB1F}xyz*D({Vabs(Q1)}xyz,dx)+{MiB1F_Q1}xyz*D({OM(abs,B1
)}xyz,dx)+{FiB2F}xyz*D({Vabs(Q2)}xyz,dx)+{MiB2F_Q2}xyz*D({OM(abs
,B2)}xyz,dx)+{FiB3F}xyz*D({Vabs(Q3)}xyz,dx)+{MiB3F_Q3}xyz*D({OM(
abs,B3)}xyz,dx)+{[0,0,-mP*g]}xyz*D({Vabs(GP)}xyz,dx)+{[0,0,-
mB1F*g]}xyz*D({Vabs(GB1F)}xyz,dx)+{[0,0,-
mB2F*g]}xyz*D({Vabs(GB2F)}xyz,dx)+{[0,0,-
mB3F*g]}xyz*D({Vabs(GB3F)}xyz,dx)+{FORCE1}B01*D({Vabs(Q1)}B01,dx
)+{FORCE2}B02*D({Vabs(Q2)}B02,dx)+{FORCE3}B03*D({Vabs(Q3)}B03,dx
) )

```

```

PRINT(+{FS1}B1*D({Vabs(P1)}B1,dx) -
{FS1}B1*D({Vabs(T1)}B1,dx)+{FVP1}B01*D({Vabs(Q1)}B01,dx)+{FCP1}B
01*D({Vabs(Q1)}B01,dx)+{MVR1_Q1}B1*D({OM(abs,B1)}B1,dx)+{MCR1_Q1
}B1*D({OM(abs,B1)}B1,dx)+{MVS1_T1}B1*D({OM(abs,BP3)}B1,dx) -
{MVS1_T1}B1*D({OM(abs,B1)}B1,dx)+{MCS1_T1}B1*D({OM(abs,BP3)}B1,d
x)-{MCS1_T1}B1*D({OM(abs,B1)}B1,dx))

```

```

PRINT(+{FS2}B2*D({Vabs(P2)}B2,dx) -
{FS2}B2*D({Vabs(T2)}B2,dx)+{FVP2}B02*D({Vabs(Q2)}B02,dx)+{FCP2}B
02*D({Vabs(Q2)}B02,dx)+{MVR2_Q2}B2*D({OM(abs,B2)}B2,dx)+{MCR2_Q2
}B2*D({OM(abs,B2)}B2,dx)+{MVS2_T2}B2*D({OM(abs,BP3)}B2,dx) -
{MVS2_T2}B2*D({OM(abs,B2)}B2,dx)+{MCS2_T2}B2*D({OM(abs,BP3)}B2,d
x)-{MCS2_T2}B2*D({OM(abs,B2)}B2,dx))

```

```

PRINT (+{FS3}B3*D({Vabs (P3) }B3, dx) -
{FS3}B3*D({Vabs (T3) }B3, dx) +{FVP3}B03*D({Vabs (Q3) }B03, dx) +{FCP3}B
03*D({Vabs (Q3) }B03, dx) +{MVR3_Q3}B3*D({OM(abs, B3) }B3, dx) +{MCR3_Q3
}B3*D({OM(abs, B3) }B3, dx) +{MVS3_T3}B3*D({OM(abs, BP3) }B3, dx) -
{MVS3_T3}B3*D({OM(abs, B3) }B3, dx) +{MCS3_T3}B3*D({OM(abs, BP3) }B3, d
x) -{MCS3_T3}B3*D({OM(abs, B3) }B3, dx) )

```

```

// -----
// DYNAMIC EQUATIONS. DERIVATION WITH RESPECT TO dy
// -----

```

```

PRINT (+{FiP}xyz*D({Vabs (OP) }xyz, dy) +{MiP_OP}xyz*D({OM(abs, BP3) }x
yz, dy) +{FiB1F}xyz*D({Vabs (Q1) }xyz, dy) +{MiB1F_Q1}xyz*D({OM(abs, B1
) }xyz, dy) +{FiB2F}xyz*D({Vabs (Q2) }xyz, dy) +{MiB2F_Q2}xyz*D({OM(abs
, B2) }xyz, dy) +{FiB3F}xyz*D({Vabs (Q3) }xyz, dy) +{MiB3F_Q3}xyz*D({OM(
abs, B3) }xyz, dy) +{ [0, 0, -mP*g] }xyz*D({Vabs (GP) }xyz, dy) +{ [0, 0, -
mB1F*g] }xyz*D({Vabs (GB1F) }xyz, dy) +{ [0, 0, -
mB2F*g] }xyz*D({Vabs (GB2F) }xyz, dy) +{ [0, 0, -
mB3F*g] }xyz*D({Vabs (GB3F) }xyz, dy) +{FORCE1}B01*D({Vabs (Q1) }B01, dy
) +{FORCE2}B02*D({Vabs (Q2) }B02, dy) +{FORCE3}B03*D({Vabs (Q3) }B03, dy
) )

```

```

PRINT (+{FS1}B1*D({Vabs (P1) }B1, dy) -
{FS1}B1*D({Vabs (T1) }B1, dy) +{FVP1}B01*D({Vabs (Q1) }B01, dy) +{FCP1}B
01*D({Vabs (Q1) }B01, dy) +{MVR1_Q1}B1*D({OM(abs, B1) }B1, dy) +{MCR1_Q1
}B1*D({OM(abs, B1) }B1, dy) +{MVS1_T1}B1*D({OM(abs, BP3) }B1, dy) -
{MVS1_T1}B1*D({OM(abs, B1) }B1, dy) +{MCS1_T1}B1*D({OM(abs, BP3) }B1, d
y) -{MCS1_T1}B1*D({OM(abs, B1) }B1, dy) )

```

```

PRINT (+{FS2}B2*D({Vabs (P2) }B2, dy) -
{FS2}B2*D({Vabs (T2) }B2, dy) +{FVP2}B02*D({Vabs (Q2) }B02, dy) +{FCP2}B
02*D({Vabs (Q2) }B02, dy) +{MVR2_Q2}B2*D({OM(abs, B2) }B2, dy) +{MCR2_Q2
}B2*D({OM(abs, B2) }B2, dy) +{MVS2_T2}B2*D({OM(abs, BP3) }B2, dy) -
{MVS2_T2}B2*D({OM(abs, B2) }B2, dy) +{MCS2_T2}B2*D({OM(abs, BP3) }B2, d
y) -{MCS2_T2}B2*D({OM(abs, B2) }B2, dy) )

```

```

PRINT (+{FS3}B3*D({Vabs (P3) }B3, dy) -
{FS3}B3*D({Vabs (T3) }B3, dy) +{FVP3}B03*D({Vabs (Q3) }B03, dy) +{FCP3}B
03*D({Vabs (Q3) }B03, dy) +{MVR3_Q3}B3*D({OM(abs, B3) }B3, dy) +{MCR3_Q3
}B3*D({OM(abs, B3) }B3, dy) +{MVS3_T3}B3*D({OM(abs, BP3) }B3, dy) -
{MVS3_T3}B3*D({OM(abs, B3) }B3, dy) +{MCS3_T3}B3*D({OM(abs, BP3) }B3, d
y) -{MCS3_T3}B3*D({OM(abs, B3) }B3, dy) )

```

```

// -----
// DYNAMIC EQUATIONS. DERIVATION WITH RESPECT TO dz
// -----

```

```

PRINT (+{FiP}xyz*D({Vabs (OP) }xyz, dz) +{MiP_OP}xyz*D({OM(abs, BP3) }x
yz, dz) +{FiB1F}xyz*D({Vabs (Q1) }xyz, dz) +{MiB1F_Q1}xyz*D({OM(abs, B1
) }xyz, dz) +{FiB2F}xyz*D({Vabs (Q2) }xyz, dz) +{MiB2F_Q2}xyz*D({OM(abs
, B2) }xyz, dz) +{FiB3F}xyz*D({Vabs (Q3) }xyz, dz) +{MiB3F_Q3}xyz*D({OM(
abs, B3) }xyz, dz) +{ [0, 0, -mP*g] }xyz*D({Vabs (GP) }xyz, dz) +{ [0, 0, -
mB1F*g] }xyz*D({Vabs (GB1F) }xyz, dz) +{ [0, 0, -
mB2F*g] }xyz*D({Vabs (GB2F) }xyz, dz) +{ [0, 0, -

```

```
mB3F*g]}xyz*D({Vabs(GB3F)}xyz,dz)+{FORCE1}B01*D({Vabs(Q1)}B01,dz)
)+{FORCE2}B02*D({Vabs(Q2)}B02,dz)+{FORCE3}B03*D({Vabs(Q3)}B03,dz
) )
```

```
PRINT(+{FS1}B1*D({Vabs(P1)}B1,dz)-
{FS1}B1*D({Vabs(T1)}B1,dz)+{FVP1}B01*D({Vabs(Q1)}B01,dz)+{FCP1}B
01*D({Vabs(Q1)}B01,dz)+{MVR1_Q1}B1*D({OM(abs,B1)}B1,dz)+{MCR1_Q1
}B1*D({OM(abs,B1)}B1,dz)+{MVS1_T1}B1*D({OM(abs,BP3)}B1,dz)-
{MVS1_T1}B1*D({OM(abs,B1)}B1,dz)+{MCS1_T1}B1*D({OM(abs,BP3)}B1,d
z)-{MCS1_T1}B1*D({OM(abs,B1)}B1,dz))
```

```
PRINT(+{FS2}B2*D({Vabs(P2)}B2,dz)-
{FS2}B2*D({Vabs(T2)}B2,dz)+{FVP2}B02*D({Vabs(Q2)}B02,dz)+{FCP2}B
02*D({Vabs(Q2)}B02,dz)+{MVR2_Q2}B2*D({OM(abs,B2)}B2,dz)+{MCR2_Q2
}B2*D({OM(abs,B2)}B2,dz)+{MVS2_T2}B2*D({OM(abs,BP3)}B2,dz)-
{MVS2_T2}B2*D({OM(abs,B2)}B2,dz)+{MCS2_T2}B2*D({OM(abs,BP3)}B2,d
z)-{MCS2_T2}B2*D({OM(abs,B2)}B2,dz))
```

```
PRINT(+{FS3}B3*D({Vabs(P3)}B3,dz)-
{FS3}B3*D({Vabs(T3)}B3,dz)+{FVP3}B03*D({Vabs(Q3)}B03,dz)+{FCP3}B
03*D({Vabs(Q3)}B03,dz)+{MVR3_Q3}B3*D({OM(abs,B3)}B3,dz)+{MCR3_Q3
}B3*D({OM(abs,B3)}B3,dz)+{MVS3_T3}B3*D({OM(abs,BP3)}B3,dz)-
{MVS3_T3}B3*D({OM(abs,B3)}B3,dz)+{MCS3_T3}B3*D({OM(abs,BP3)}B3,d
z)-{MCS3_T3}B3*D({OM(abs,B3)}B3,dz))
```

```
// -----
// DYNAMIC EQUATIONS. DERIVATION WITH RESPECT TO dji
// -----
```

```
PRINT(+{FiP}xyz*D({Vabs(OP)}xyz,dji)+{MiP_OP}xyz*D({OM(abs,BP3)}
xyz,dji)+{FiB1F}xyz*D({Vabs(Q1)}xyz,dji)+{MiB1F_Q1}xyz*D({OM(abs
,B1)}xyz,dji)+{FiB2F}xyz*D({Vabs(Q2)}xyz,dji)+{MiB2F_Q2}xyz*D({O
M(abs,B2)}xyz,dji)+{FiB3F}xyz*D({Vabs(Q3)}xyz,dji)+{MiB3F_Q3}xyz
*D({OM(abs,B3)}xyz,dji)+{[0,0,-
mP*g]}xyz*D({Vabs(GP)}xyz,dji)+{[0,0,-
mB1F*g]}xyz*D({Vabs(GB1F)}xyz,dji)+{[0,0,-
mB2F*g]}xyz*D({Vabs(GB2F)}xyz,dji)+{[0,0,-
mB3F*g]}xyz*D({Vabs(GB3F)}xyz,dji)+{FORCE1}B01*D({Vabs(Q1)}B01,d
ji)+{FORCE2}B02*D({Vabs(Q2)}B02,dji)+{FORCE3}B03*D({Vabs(Q3)}B03
,dji) )
```

```
PRINT(+{FS1}B1*D({Vabs(P1)}B1,dji)-
{FS1}B1*D({Vabs(T1)}B1,dji)+{FVP1}B01*D({Vabs(Q1)}B01,dji)+{FCP1
}B01*D({Vabs(Q1)}B01,dji)+{MVR1_Q1}B1*D({OM(abs,B1)}B1,dji)+{MCR
1_Q1}B1*D({OM(abs,B1)}B1,dji)+{MVS1_T1}B1*D({OM(abs,BP3)}B1,dji)
-
{MVS1_T1}B1*D({OM(abs,B1)}B1,dji)+{MCS1_T1}B1*D({OM(abs,BP3)}B1,
dji)-{MCS1_T1}B1*D({OM(abs,B1)}B1,dji))
```

```
PRINT(+{FS2}B2*D({Vabs(P2)}B2,dji)-
{FS2}B2*D({Vabs(T2)}B2,dji)+{FVP2}B02*D({Vabs(Q2)}B02,dji)+{FCP2
}B02*D({Vabs(Q2)}B02,dji)+{MVR2_Q2}B2*D({OM(abs,B2)}B2,dji)+{MCR
2_Q2}B2*D({OM(abs,B2)}B2,dji)+{MVS2_T2}B2*D({OM(abs,BP3)}B2,dji)
-
```



```
{MVS2_T2}B2*D({OM(abs,B2)}B2,dji)+{MCS2_T2}B2*D({OM(abs,BP3)}B2,
dji)-{MCS2_T2}B2*D({OM(abs,B2)}B2,dji))
```

```
PRINT(+{FS3}B3*D({Vabs(P3)}B3,dji)-
{FS3}B3*D({Vabs(T3)}B3,dji)+{FVP3}B03*D({Vabs(Q3)}B03,dji)+{FCP3
}B03*D({Vabs(Q3)}B03,dji)+{MVR3_Q3}B3*D({OM(abs,B3)}B3,dji)+{MCR
3_Q3}B3*D({OM(abs,B3)}B3,dji)+{MVS3_T3}B3*D({OM(abs,BP3)}B3,dji)
-
{MVS3_T3}B3*D({OM(abs,B3)}B3,dji)+{MCS3_T3}B3*D({OM(abs,BP3)}B3,
dji)-{MCS3_T3}B3*D({OM(abs,B3)}B3,dji))
```

```
// -----
// DYNAMIC EQUATIONS. DERIVATION WITH RESPECT TO dti
// -----
```

```
PRINT(+{FiP}xyz*D({Vabs(OP)}xyz,dti)+{MiP_OP}xyz*D({OM(abs,BP3)}
xyz,dti)+{FiB1F}xyz*D({Vabs(Q1)}xyz,dti)+{MiB1F_Q1}xyz*D({OM(abs
,B1)}xyz,dti)+{FiB2F}xyz*D({Vabs(Q2)}xyz,dti)+{MiB2F_Q2}xyz*D({O
M(abs,B2)}xyz,dti)+{FiB3F}xyz*D({Vabs(Q3)}xyz,dti)+{MiB3F_Q3}xyz
*D({OM(abs,B3)}xyz,dti)+{[0,0,-
mP*g]}xyz*D({Vabs(GP)}xyz,dti)+{[0,0,-
mB1F*g]}xyz*D({Vabs(GB1F)}xyz,dti)+{[0,0,-
mB2F*g]}xyz*D({Vabs(GB2F)}xyz,dti)+{[0,0,-
mB3F*g]}xyz*D({Vabs(GB3F)}xyz,dti)+{FORCE1}B01*D({Vabs(Q1)}B01,d
ti)+{FORCE2}B02*D({Vabs(Q2)}B02,dti)+{FORCE3}B03*D({Vabs(Q3)}B03
,dti) )
```

```
PRINT(+{FS1}B1*D({Vabs(P1)}B1,dti)-
{FS1}B1*D({Vabs(T1)}B1,dti)+{FVP1}B01*D({Vabs(Q1)}B01,dti)+{FCP1
}B01*D({Vabs(Q1)}B01,dti)+{MVR1_Q1}B1*D({OM(abs,B1)}B1,dti)+{MCR
1_Q1}B1*D({OM(abs,B1)}B1,dti)+{MVS1_T1}B1*D({OM(abs,BP3)}B1,dti)
-
{MVS1_T1}B1*D({OM(abs,B1)}B1,dti)+{MCS1_T1}B1*D({OM(abs,BP3)}B1,
dti)-{MCS1_T1}B1*D({OM(abs,B1)}B1,dti))
```

```
PRINT(+{FS2}B2*D({Vabs(P2)}B2,dti)-
{FS2}B2*D({Vabs(T2)}B2,dti)+{FVP2}B02*D({Vabs(Q2)}B02,dti)+{FCP2
}B02*D({Vabs(Q2)}B02,dti)+{MVR2_Q2}B2*D({OM(abs,B2)}B2,dti)+{MCR
2_Q2}B2*D({OM(abs,B2)}B2,dti)+{MVS2_T2}B2*D({OM(abs,BP3)}B2,dti)
-
{MVS2_T2}B2*D({OM(abs,B2)}B2,dti)+{MCS2_T2}B2*D({OM(abs,BP3)}B2,
dti)-{MCS2_T2}B2*D({OM(abs,B2)}B2,dti))
```

```
PRINT(+{FS3}B3*D({Vabs(P3)}B3,dti)-
{FS3}B3*D({Vabs(T3)}B3,dti)+{FVP3}B03*D({Vabs(Q3)}B03,dti)+{FCP3
}B03*D({Vabs(Q3)}B03,dti)+{MVR3_Q3}B3*D({OM(abs,B3)}B3,dti)+{MCR
3_Q3}B3*D({OM(abs,B3)}B3,dti)+{MVS3_T3}B3*D({OM(abs,BP3)}B3,dti)
-
{MVS3_T3}B3*D({OM(abs,B3)}B3,dti)+{MCS3_T3}B3*D({OM(abs,BP3)}B3,
dti)-{MCS3_T3}B3*D({OM(abs,B3)}B3,dti))
```

```
// -----
// DYNAMIC EQUATIONS. DERIVATION WITH RESPECT TO dri
// -----
```

```

PRINT(+{FiP}xyz*D({Vabs(OP)}xyz,dri)+{MiP_OP}xyz*D({OM(abs,BP3)}
xyz,dri)+{FiB1F}xyz*D({Vabs(Q1)}xyz,dri)+{MiB1F_Q1}xyz*D({OM(abs
,B1)}xyz,dri)+{FiB2F}xyz*D({Vabs(Q2)}xyz,dri)+{MiB2F_Q2}xyz*D({O
M(abs,B2)}xyz,dri)+{FiB3F}xyz*D({Vabs(Q3)}xyz,dri)+{MiB3F_Q3}xyz
*D({OM(abs,B3)}xyz,dri)+{[0,0,-
mP*g]}xyz*D({Vabs(GP)}xyz,dri)+{[0,0,-
mB1F*g]}xyz*D({Vabs(GB1F)}xyz,dri)+{[0,0,-
mB2F*g]}xyz*D({Vabs(GB2F)}xyz,dri)+{[0,0,-
mB3F*g]}xyz*D({Vabs(GB3F)}xyz,dri)+{FORCE1}B01*D({Vabs(Q1)}B01,d
ri)+{FORCE2}B02*D({Vabs(Q2)}B02,dri)+{FORCE3}B03*D({Vabs(Q3)}B03
,dri) )

```

```

PRINT(+{FS1}B1*D({Vabs(P1)}B1,dri)-
{FS1}B1*D({Vabs(T1)}B1,dri)+{FVP1}B01*D({Vabs(Q1)}B01,dri)+{FCP1
}B01*D({Vabs(Q1)}B01,dri)+{MVR1_Q1}B1*D({OM(abs,B1)}B1,dri)+{MCR
1_Q1}B1*D({OM(abs,B1)}B1,dri)+{MVS1_T1}B1*D({OM(abs,BP3)}B1,dri)
-
{MVS1_T1}B1*D({OM(abs,B1)}B1,dri)+{MCS1_T1}B1*D({OM(abs,BP3)}B1,
dri)-{MCS1_T1}B1*D({OM(abs,B1)}B1,dri))

```

```

PRINT(+{FS2}B2*D({Vabs(P2)}B2,dri)-
{FS2}B2*D({Vabs(T2)}B2,dri)+{FVP2}B02*D({Vabs(Q2)}B02,dri)+{FCP2
}B02*D({Vabs(Q2)}B02,dri)+{MVR2_Q2}B2*D({OM(abs,B2)}B2,dri)+{MCR
2_Q2}B2*D({OM(abs,B2)}B2,dri)+{MVS2_T2}B2*D({OM(abs,BP3)}B2,dri)
-
{MVS2_T2}B2*D({OM(abs,B2)}B2,dri)+{MCS2_T2}B2*D({OM(abs,BP3)}B2,
dri)-{MCS2_T2}B2*D({OM(abs,B2)}B2,dri))

```

```

PRINT(+{FS3}B3*D({Vabs(P3)}B3,dri)-
{FS3}B3*D({Vabs(T3)}B3,dri)+{FVP3}B03*D({Vabs(Q3)}B03,dri)+{FCP3
}B03*D({Vabs(Q3)}B03,dri)+{MVR3_Q3}B3*D({OM(abs,B3)}B3,dri)+{MCR
3_Q3}B3*D({OM(abs,B3)}B3,dri)+{MVS3_T3}B3*D({OM(abs,BP3)}B3,dri)
-
{MVS3_T3}B3*D({OM(abs,B3)}B3,dri)+{MCS3_T3}B3*D({OM(abs,BP3)}B3,
dri)-{MCS3_T3}B3*D({OM(abs,B3)}B3,dri))

```

9.2. Algoritmo para la obtención modelo dinámico y reducción del modelo del robot 3-PRS, mediante el software Matlab®

En los apartados posteriores se muestra el código que se ha desarrollado para la realización del modelo cinemático y modelo dinámico del robot 3-PRS a través de *Matlab®*, así como, la reducción del modelo a parámetros base mediante la *DVS*. Además, se muestra el código utilizado para la obtención de los parámetros de la trayectoria impuesta.

9.2.1. main_symbolic.m

```
function main_symbolic

disp('Defining variables ...');
% -----
% COORDINATES AND DERIVATIVES
% -----
% Actuators
syms q1 dq1 ddq1 real
syms q2 dq2 ddq2 real
syms q3 dq3 ddq3 real
syms q4 dq4 ddq4 real
syms q5 dq5 ddq5 real
syms q6 dq6 ddq6 real

% Platform
syms x dx ddx real
syms y dy ddy real
syms z dz ddz real
syms ji dji ddji real
syms ti dti ddti real
syms ri dri ddri real

% -----
% GEOMETRICAL AND OTHER CONSTANTS
% -----
syms LN LR LB real
syms SQR3 real
syms PI real
syms g real

% -----
% JOINT FORCES AND TORQUES
% -----
% Spherical joint 1
syms FS1x FS1y FS1z real
% Spherical joint 2
syms FS2x FS2y FS2z real
% Spherical joint 3
syms FS3x FS3y FS3z real

% Actuators forces
syms FORCE1z FORCE2z FORCE3z real

% -----
% FRICTION FORCES
% -----
```

```

% Prismatic Constant Coulomb friction forces
syms FCCP1z FCCP2z FCCP3z real

% -----
% THE SYMBOLIC SIGNS OF THE VELOCITIES
% -----
syms sign_dq1 sign_dq2 sign_dq3 real
syms sign_dq4 sign_dq5 sign_dq6 real

% -----
% THE SYMBOLIC ANGULAR VELOCITIES
% -----
syms omega_bar1 omega_bar2 omega_bar3 real;

% -----
% INERTIAL PARAMETERS DEFINITION
% -----
parameters_definitions;
parameters_vectors;

% -----
% KINEMATIC VARIABLES
% -----
% Generalized Coordinates Vector
Q=[ q1;q2;q3;
    q4;q5;q6;
    x;y;z;
    ji;ti;ri];

% Generalized Coordinates Velocities Vector
DQ=[ dq1;dq2;dq3;
     dq4;dq5;dq6;
     dx;dy;dz;
     dji;dqi;dri];

% Generalized Coordinates Accelerations Vector
DDQ=[ ddq1;ddq2;ddq3;
       ddq4;ddq5;ddq6;
       ddx;ddy;ddz;
       ddji;ddti;ddri];

% Dependant Generalized Coordinates Vector
Qdep=[q4;q5;q6;
      x;y;z;
      ji;ti;ri];

% Dependant Generalized Coordinates Velocities Vector
DQdep=[dq4;dq5;dq6;
       dx;dy;dz;
       dji;dqi;dri];

% Dependant Generalized Coordinates Acelerations Vector
DDQdep=[ddq4;ddq5;ddq6;
        ddx;ddy;ddz;
        ddji;ddti;ddri];

% Independant Generalized Coordinates Vector
Qind=[q1;q2;q3];

```

```

% Independant Generalized Coordinates Velocities Vector
DQind=[dq1;dq2;dq3];

% Independant Generalized Coordinates Acelerations Vector
DDQind=[ddq1;ddq2;ddq3];

% Generalized Coordinates Vector for Kinematic Simulation
KQ=Q;

% Generalized Coordinates Velocities Vector for Kinematic Simulation
KDQ=DQ;

% Generalized Coordinates Accelerations Vector for Kinematic
Simulation
KDDQ=DDQ;

% Dependant Generalized Coordinates Vector for Kinematic Simulation
KQdep=Qdep;

% Dependant Generalized Coordinates Velocities Vector for Kinematic
% Simulation
KDQdep=DQdep;

% Dependant Generalized Coordinates Accelerations Vector for Kinematic
% Simulation
KDDQdep=DDQdep;

% Independant Generalized Coordinates Vector for Kinematic Simulation
KQind=Qind;

% Independant Generalized Coordinates Velocities Vector for Kinematic
% Simulation
KDQind=DQind;

% Independant Generalized Coordinates Accelerations Vector for
Kinematic
% Simulation
KDDQind=DDQind;

% -----
% DYNAMIC VARIABLES
% -----
Epsilon= [FS1x;FS1y;FS1z;
          FS2x;FS2y;FS2z;
          FS3x;FS3y;FS3z];

DDQ_Epsilon=[DDQ; Epsilon];

% Spherical joint forces
FS=[FS1x;FS1y;FS1z;
    FS2x;FS2y;FS2z;
    FS3x;FS3y;FS3z];

% Actuators forces
FORCE=[ FORCE1z;FORCE2z;FORCE3z];

```

```

FALL=[FORCE;FS];

disp('Variables DEFINED')

%-----
% KINEMATIC EQUATIONS
%-----
    BI=2;
    HIRU=3;

if false
    kinematic_eq;

    % Simplify the kinematic equations
    disp('Simplifying Kinematic Equations')
    PI=pi;
    SQR3=sqrt(3);
    pos_eq=subs(pos_eq);
    vel_eq=subs(vel_eq);
    ace_eq=subs(ace_eq);

    pos_eq=simplify(pos_eq);
    vel_eq=simplify(vel_eq);
    ace_eq=simplify(ace_eq);

    % Saving Kinematic equations
    save pos_eq pos_eq;
    save vel_eq vel_eq;
    save ace_eq ace_eq;
else
    disp('Loading Kinematic Equations')
    load pos_eq;
    load vel_eq;
    load ace_eq;
end

%-----
% Kinematics MATRICES and VECTORS
%-----
disp('Calculating Kinematic matrices and vectors');
KJAC=jacobian(pos_eq,KQdep);
PHI_T=vel_eq-KJAC*KDQdep;
C=ace_eq-KJAC*KDDQdep;

% Kinematic MATRICES and VECTORS simplifications
KJAC=simplify(KJAC);
PHI_T=simplify(PHI_T);
C=simplify(C);

% Saving Kinematic MATRICES and VECTORS
save KJAC KJAC;
save PHI_T PHI_T;
save C C;

matlabFunction( pos_eq,'file','evalpos_eq','var',{Q,PARAM_GEOM})
matlabFunction( KJAC,'file','evalKJAC','var',{Q,PARAM_GEOM})

```

```

matlabFunction( PHI_T, 'file', 'evalPHI_T', 'var', {Q, DQind, PARAM_GEOM})
matlabFunction( C, 'file', 'evalC', 'var', {Q, DQ, DDQind, PARAM_GEOM})

%-----
% DYNAMIC EQUATIONS
%-----
if false
    disp('Loading Dynamic Equations...');
    dynamic_eq_origin;

    EQ=[eq_01;eq_02;eq_03;eq_04;eq_05;eq_06;
        eq_07;eq_08;eq_09;eq_10;eq_11;eq_12];

    disp('Simplifying Dynamic Equations');

    EQ=subs(EQ);
    EQ=simplify(EQ);
    EQ;

    save EQ EQ;
    disp('Dynamic Equations Saved');
else
    load EQ.mat;
    disp('Dynamic Equations Loaded')
end

%-----
% IDENTIFICATION EQUATIONS
%-----
% Observation matrix calculation of the open chain system
disp('Open chain observation matrix deriving and simplifying');
K=jacobian(EQ,PARAM_V_LIN);
K=simplify(K);

% Submatrix of K where the RHS is different from 0
Ki=K(1:3,:);

% Submatrix of K where the RHS is equal to 0
Kd=K(4:12,:);

% Jacobian matrix of the acceleration equations w.r.t. the generalized
% coordinates
disp('Jacobian matrix of the kinematic equations (A) deriving and
simplifying');

% Alternatively
V=jacobian(EQ,Epsilon);
V=simplify(V);

% Submatrix of K where the RHS is different from 0
Vi=V(1:3,:);
Ai=Vi';

% Submatrix of K where the RHS is equal to 0
Vd=V(4:12,:);
Ad=Vd';

```

```

save K K;
save Ki Ki;
save Kd Kd;
save Ai Ai;
save Ad Ad;

```

```

matlabFunction( Ki, 'file', 'evalKi', 'vars', {Q, DQ, DDQ, PARAM_GEOM, SIGN})
matlabFunction(
Kd, 'file', 'evalKd', 'vars', {Q, DQ, DDQ, PARAM_GEOM, SIGN, OMEGA})
matlabFunction( Ai, 'file', 'evalAi', 'vars', {Q})
matlabFunction( Ad, 'file', 'evalAd', 'vars', {Q, PARAM_GEOM})

```

9.2.2. parameters_definitions.m

```

% -----
% INERTIAL PARAMETERS
% -----
% Platform Inertia
syms mP mPx mPy mPz real
syms IPxx IPyy IPzz IPxy IPxz IPyz real
% Bar 1 Inertia
syms mB1F mB1Fx mB1Fy mB1Fz real
syms IB1Fxx IB1Fyy IB1Fzz IB1Fxy IB1Fxz IB1Fyz real
% Bar 2 Inertia
syms mB2F mB2Fx mB2Fy mB2Fz real
syms IB2Fxx IB2Fyy IB2Fzz IB2Fxy IB2Fxz IB2Fyz real
% Bar 3 Inertia
syms mB3F mB3Fx mB3Fy mB3Fz real
syms IB3Fxx IB3Fyy IB3Fzz IB3Fxy IB3Fxz IB3Fyz real

% -----
% VISCOUS FRICTION PARAMETERS
% -----
syms VISP1 VISP2 VISP3 real
syms VISR1 VISR2 VISR3 real
syms VISS1 VISS2 VISS3 real

% -----
% COULOMB FRICTION PARAMETERS
% -----
syms FCCP1z FCCP2z FCCP3z real
syms MCCR1z MCCR2z MCCR3z real
syms MCCS1 MCCS2 MCCS3 real

% -----
% GEOMETRIC PARAMETERS
% -----
syms LN LR LB g real

% -----
% OTHER PARAMETERS
% -----
syms SQR3 PI
syms sign_dq1 sign_dq2 sign_dq3 sign_dq4 sign_dq5 sign_dq6 real
syms omega_bar1 omega_bar2 omega_bar3 real

```


9.2.3. parameters_vectors.m

```

PARAM_V_LIN=[
% Platform Inertia
    mP;mPx;mPy;mPz;IPxx;IPyy;IPzz;IPxy;IPxz;IPyz; %10
% Bar 1 Inertia
    mB1F;mB1Fx;mB1Fy;mB1Fz;IB1Fxx;IB1Fyy;IB1Fzz;IB1Fxy;IB1Fxz;IB1Fyz;
%3
% Bar 2 Inertia
    mB2F;mB2Fx;mB2Fy;mB2Fz;IB2Fxx;IB2Fyy;IB2Fzz;IB2Fxy;IB2Fxz;IB2Fyz;
%3
% Bar 3 Inertia
    mB3F;mB3Fx;mB3Fy;mB3Fz;IB3Fxx;IB3Fyy;IB3Fzz;IB3Fxy;IB3Fxz;IB3Fyz;
%3
% Viscous friction on the Prismatic joint
    VISP1;VISP2;VISP3; %22,21,22
% Viscous friction on the Revolution joint
    VISR1;VISR2;VISR3; %23,24,25
% Viscous friction on the Spherical joint
    VISS1;VISS2;VISS3; %26,27,28
% Coulomb constant friction on the Prismatic joint
    FCCP1z;FCCP2z;FCCP3z; % 29,30,31
% Coulomb constant friction on the Revolution joint
    MCCR1z;MCCR2z;MCCR3z; %32,33,34
% Coulomb constant friction on the Spherical joint
    MCCS1;MCCS2;MCCS3]; % 35,36,37

% Geometric parameters
PARAM_GEOM=[LN;LR;LB;g;SQR3;PI];

% Other constants
SIGN=[ sign_dq1;sign_dq2;sign_dq3;
      sign_dq4;sign_dq5;sign_dq6];

OMEGA=[omega_bar1;omega_bar2;omega_bar3];

```

9.2.4. kinematic_eq.m

```

%-----
% KINEMATIC EQUATIONS
%-----

disp('Swallowing kinematic equations')
sprintf('Swallowing .');

pos_eq=[
    x-cos(ti)*cos(ri)*LN/BI+cos(ti)*sin(ri)*LN/BI/SQR3-
    sin(PI)*LB/SQR3+sin(PI)*cos(-q4)*LR;
    y-cos(ji)*sin(ri)*LN/BI-sin(ji)*sin(ti)*cos(ri)*LN/BI-
    cos(ji)*cos(ri)*LN/BI/SQR3+sin(ji)*sin(ti)*sin(ri)*LN/BI/SQR3+cos(PI)*
    LB/SQR3-cos(PI)*cos(-q4)*LR;
    z-sin(ji)*sin(ri)*LN/BI+cos(ji)*sin(ti)*cos(ri)*LN/BI-
    sin(ji)*cos(ri)*LN/BI/SQR3-cos(ji)*sin(ti)*sin(ri)*LN/BI/SQR3-q1-sin(-
    q4)*LR;
%
    x+cos(ti)*cos(ri)*LN/BI+cos(ti)*sin(ri)*LN/BI/SQR3-
    sin(PI*BI/HIRU+PI)*LB/SQR3+sin(PI*BI/HIRU+PI)*cos(-q5)*LR;

```

```

        y+cos(ji)*sin(ri)*LN/BI+sin(ji)*sin(ti)*cos(ri)*LN/BI-
cos(ji)*cos(ri)*LN/BI/SQR3+sin(ji)*sin(ti)*sin(ri)*LN/BI/SQR3+cos(PI*B
I/HIRU+PI)*LB/SQR3-cos(PI*BI/HIRU+PI)*cos(-q5)*LR;
        z+sin(ji)*sin(ri)*LN/BI-cos(ji)*sin(ti)*cos(ri)*LN/BI-
sin(ji)*cos(ri)*LN/BI/SQR3-cos(ji)*sin(ti)*sin(ri)*LN/BI/SQR3-q2-sin(-
q5)*LR;
%
        x-cos(ti)*sin(ri)*LN/SQR3-sin(-PI*BI/HIRU+PI)*LB/SQR3+sin(-
PI*BI/HIRU+PI)*cos(-q6)*LR;
        y+cos(ji)*cos(ri)*LN/SQR3-sin(ji)*sin(ti)*sin(ri)*LN/SQR3+cos(-
PI*BI/HIRU+PI)*LB/SQR3-cos(-PI*BI/HIRU+PI)*cos(-q6)*LR;
        z+sin(ji)*cos(ri)*LN/SQR3+cos(ji)*sin(ti)*sin(ri)*LN/SQR3-q3-sin(-
q6)*LR;

];
%
fprintf(' ');
%
vel_eq=[
        dx+dti*sin(ti)*cos(ri)*LN/BI+cos(ti)*dri*sin(ri)*LN/BI-
dti*sin(ti)*sin(ri)*LN/BI/SQR3+cos(ti)*dri*cos(ri)*LN/BI/SQR3+sin(PI)
dq4*sin(-q4)*LR;
        dy+dji*sin(ji)*sin(ri)*LN/BI-cos(ji)*dri*cos(ri)*LN/BI-
dji*cos(ji)*sin(ti)*cos(ri)*LN/BI-
sin(ji)*dti*cos(ti)*cos(ri)*LN/BI+sin(ji)*sin(ti)*dri*sin(ri)*LN/BI+dj
i*sin(ji)*cos(ri)*LN/BI/SQR3+cos(ji)*dri*sin(ri)*LN/BI/SQR3+dji*cos(ji
)*sin(ti)*sin(ri)*LN/BI/SQR3+sin(ji)*dti*cos(ti)*sin(ri)*LN/BI/SQR3+si
n(ji)*sin(ti)*dri*cos(ri)*LN/BI/SQR3-cos(PI)*dq4*sin(-q4)*LR;
        dz-dji*cos(ji)*sin(ri)*LN/BI-sin(ji)*dri*cos(ri)*LN/BI-
dji*sin(ji)*sin(ti)*cos(ri)*LN/BI+cos(ji)*dti*cos(ti)*cos(ri)*LN/BI-
cos(ji)*sin(ti)*dri*sin(ri)*LN/BI-
dji*cos(ji)*cos(ri)*LN/BI/SQR3+sin(ji)*dri*sin(ri)*LN/BI/SQR3+dji*sin(
ji)*sin(ti)*sin(ri)*LN/BI/SQR3-cos(ji)*dti*cos(ti)*sin(ri)*LN/BI/SQR3-
cos(ji)*sin(ti)*dri*cos(ri)*LN/BI/SQR3-dq1+dq4*cos(-q4)*LR;
%
        dx-dti*sin(ti)*cos(ri)*LN/BI-cos(ti)*dri*sin(ri)*LN/BI-
dti*sin(ti)*sin(ri)*LN/BI/SQR3+cos(ti)*dri*cos(ri)*LN/BI/SQR3+sin(PI*B
I/HIRU+PI)*dq5*sin(-q5)*LR;
        dy-
dji*sin(ji)*sin(ri)*LN/BI+cos(ji)*dri*cos(ri)*LN/BI+dji*cos(ji)*sin(ti
)*cos(ri)*LN/BI+sin(ji)*dti*cos(ti)*cos(ri)*LN/BI-
sin(ji)*sin(ti)*dri*sin(ri)*LN/BI+dji*sin(ji)*cos(ri)*LN/BI/SQR3+cos(j
i)*dri*sin(ri)*LN/BI/SQR3+dji*cos(ji)*sin(ti)*sin(ri)*LN/BI/SQR3+sin(j
i)*dti*cos(ti)*sin(ri)*LN/BI/SQR3+sin(ji)*sin(ti)*dri*cos(ri)*LN/BI/SQ
R3-cos(PI*BI/HIRU+PI)*dq5*sin(-q5)*LR;

dz+dji*cos(ji)*sin(ri)*LN/BI+sin(ji)*dri*cos(ri)*LN/BI+dji*sin(ji)*sin
(ti)*cos(ri)*LN/BI-
cos(ji)*dti*cos(ti)*cos(ri)*LN/BI+cos(ji)*sin(ti)*dri*sin(ri)*LN/BI-
dji*cos(ji)*cos(ri)*LN/BI/SQR3+sin(ji)*dri*sin(ri)*LN/BI/SQR3+dji*sin(
ji)*sin(ti)*sin(ri)*LN/BI/SQR3-cos(ji)*dti*cos(ti)*sin(ri)*LN/BI/SQR3-
cos(ji)*sin(ti)*dri*cos(ri)*LN/BI/SQR3-dq2+dq5*cos(-q5)*LR;
%
        dx+dti*sin(ti)*sin(ri)*LN/SQR3-cos(ti)*dri*cos(ri)*LN/SQR3+sin(-
PI*BI/HIRU+PI)*dq6*sin(-q6)*LR;
        dy-dji*sin(ji)*cos(ri)*LN/SQR3-cos(ji)*dri*sin(ri)*LN/SQR3-
dji*cos(ji)*sin(ti)*sin(ri)*LN/SQR3-
sin(ji)*dti*cos(ti)*sin(ri)*LN/SQR3-

```

```

sin(ji)*sin(ti)*dri*cos(ri)*LN/SQR3-cos(-PI*BI/HIRU+PI)*dq6*sin(-
q6)*LR;
    dz+dji*cos(ji)*cos(ri)*LN/SQR3-sin(ji)*dri*sin(ri)*LN/SQR3-
dji*sin(ji)*sin(ti)*sin(ri)*LN/SQR3+cos(ji)*dti*cos(ti)*sin(ri)*LN/SQR
3+cos(ji)*sin(ti)*dri*cos(ri)*LN/SQR3-dq3+dq6*cos(-q6)*LR;

];
%
fprintf(' ');
%
ace_eq=[
    ddx+sin(ti)*cos(ri)*LN*ddti/BI+dti^2*cos(ti)*cos(ri)*LN/BI-
2*dti*sin(ti)*dri*sin(ri)*LN/BI+cos(ti)*sin(ri)*LN*ddri/BI+cos(ti)*dri
^2*cos(ri)*LN/BI-sin(ti)*sin(ri)*LN*ddti/BI/SQR3-
dti^2*cos(ti)*sin(ri)*LN/BI/SQR3-
2*dti*sin(ti)*dri*cos(ri)*LN/BI/SQR3+cos(ti)*cos(ri)*LN*ddri/BI/SQR3-
cos(ti)*dri^2*sin(ri)*LN/BI/SQR3+sin(PI)*sin(-q4)*LR*ddq4-
sin(PI)*dq4^2*cos(-q4)*LR;

ddy+sin(ji)*sin(ri)*LN*ddji/BI+dji^2*cos(ji)*sin(ri)*LN/BI+2*dji*sin(j
i)*dri*cos(ri)*LN/BI-
cos(ji)*cos(ri)*LN*ddri/BI+cos(ji)*dri^2*sin(ri)*LN/BI-
cos(ji)*sin(ti)*cos(ri)*LN*ddji/BI+dji^2*sin(ji)*sin(ti)*cos(ri)*LN/BI
-
2*dji*cos(ji)*dti*cos(ti)*cos(ri)*LN/BI+2*dji*cos(ji)*sin(ti)*dri*sin(
ri)*LN/BI-
sin(ji)*cos(ti)*cos(ri)*LN*ddti/BI+sin(ji)*dti^2*sin(ti)*cos(ri)*LN/BI
+2*sin(ji)*dti*cos(ti)*dri*sin(ri)*LN/BI+sin(ji)*sin(ti)*sin(ri)*LN*dd
ri/BI+sin(ji)*sin(ti)*dri^2*cos(ri)*LN/BI+sin(ji)*cos(ri)*LN*ddji/BI/S
QR3+dji^2*cos(ji)*cos(ri)*LN/BI/SQR3-
2*dji*sin(ji)*dri*sin(ri)*LN/BI/SQR3+cos(ji)*sin(ri)*LN*ddri/BI/SQR3+c
os(ji)*dri^2*cos(ri)*LN/BI/SQR3+cos(ji)*sin(ti)*sin(ri)*LN*ddji/BI/SQR
3-
dji^2*sin(ji)*sin(ti)*sin(ri)*LN/BI/SQR3+2*dji*cos(ji)*dti*cos(ti)*sin
(ri)*LN/BI/SQR3+2*dji*cos(ji)*sin(ti)*dri*cos(ri)*LN/BI/SQR3+sin(ji)*c
os(ti)*sin(ri)*LN*ddti/BI/SQR3-
sin(ji)*dti^2*sin(ti)*sin(ri)*LN/BI/SQR3+2*sin(ji)*dti*cos(ti)*dri*cos
(ri)*LN/BI/SQR3+sin(ji)*sin(ti)*cos(ri)*LN*ddri/BI/SQR3-
sin(ji)*sin(ti)*dri^2*sin(ri)*LN/BI/SQR3-cos(PI)*sin(-
q4)*LR*ddq4+cos(PI)*dq4^2*cos(-q4)*LR;
    ddz-cos(ji)*sin(ri)*LN*ddji/BI+dji^2*sin(ji)*sin(ri)*LN/BI-
2*dji*cos(ji)*dri*cos(ri)*LN/BI-
sin(ji)*cos(ri)*LN*ddri/BI+sin(ji)*dri^2*sin(ri)*LN/BI-
sin(ji)*sin(ti)*cos(ri)*LN*ddji/BI-
dji^2*cos(ji)*sin(ti)*cos(ri)*LN/BI-
2*dji*sin(ji)*dti*cos(ti)*cos(ri)*LN/BI+2*dji*sin(ji)*sin(ti)*dri*sin(
ri)*LN/BI+cos(ji)*cos(ti)*cos(ri)*LN*ddti/BI-
cos(ji)*dti^2*sin(ti)*cos(ri)*LN/BI-
2*cos(ji)*dti*cos(ti)*dri*sin(ri)*LN/BI-
cos(ji)*sin(ti)*sin(ri)*LN*ddri/BI-
cos(ji)*sin(ti)*dri^2*cos(ri)*LN/BI-
cos(ji)*cos(ri)*LN*ddji/BI/SQR3+dji^2*sin(ji)*cos(ri)*LN/BI/SQR3+2*dji
*cos(ji)*dri*sin(ri)*LN/BI/SQR3+sin(ji)*sin(ri)*LN*ddri/BI/SQR3+sin(ji
)*dri^2*cos(ri)*LN/BI/SQR3+sin(ji)*sin(ti)*sin(ri)*LN*ddji/BI/SQR3+dji
^2*cos(ji)*sin(ti)*sin(ri)*LN/BI/SQR3+2*dji*sin(ji)*dti*cos(ti)*sin(ri
)*LN/BI/SQR3+2*dji*sin(ji)*sin(ti)*dri*cos(ri)*LN/BI/SQR3-
cos(ji)*cos(ti)*sin(ri)*LN*ddti/BI/SQR3+cos(ji)*dti^2*sin(ti)*sin(ri)*
LN/BI/SQR3-2*cos(ji)*dti*cos(ti)*dri*cos(ri)*LN/BI/SQR3-

```

```

cos(ji)*sin(ti)*cos(ri)*LN*ddri/BI/SQR3+cos(ji)*sin(ti)*dri^2*sin(ri)*
LN/BI/SQR3-ddq1+cos(-q4)*LR*ddq4+dq4^2*sin(-q4)*LR;
%
    ddx-sin(ti)*cos(ri)*LN*ddti/BI-
dti^2*cos(ti)*cos(ri)*LN/BI+2*dti*sin(ti)*dri*sin(ri)*LN/BI-
cos(ti)*sin(ri)*LN*ddri/BI-cos(ti)*dri^2*cos(ri)*LN/BI-
sin(ti)*sin(ri)*LN*ddti/BI/SQR3-dti^2*cos(ti)*sin(ri)*LN/BI/SQR3-
2*dti*sin(ti)*dri*cos(ri)*LN/BI/SQR3+cos(ti)*cos(ri)*LN*ddri/BI/SQR3-
cos(ti)*dri^2*sin(ri)*LN/BI/SQR3+sin(PI*BI/HIRU+PI)*sin(-q5)*LR*ddq5-
sin(PI*BI/HIRU+PI)*dq5^2*cos(-q5)*LR;
    ddy-sin(ji)*sin(ri)*LN*ddji/BI-dji^2*cos(ji)*sin(ri)*LN/BI-
2*dji*sin(ji)*dri*cos(ri)*LN/BI+cos(ji)*cos(ri)*LN*ddri/BI-
cos(ji)*dri^2*sin(ri)*LN/BI+cos(ji)*sin(ti)*cos(ri)*LN*ddji/BI-
dji^2*sin(ji)*sin(ti)*cos(ri)*LN/BI+2*dji*cos(ji)*dti*cos(ti)*cos(ri)*
LN/BI-
2*dji*cos(ji)*sin(ti)*dri*sin(ri)*LN/BI+sin(ji)*cos(ti)*cos(ri)*LN*ddt
i/BI-sin(ji)*dti^2*sin(ti)*cos(ri)*LN/BI-
2*sin(ji)*dti*cos(ti)*dri*sin(ri)*LN/BI-
sin(ji)*sin(ti)*sin(ri)*LN*ddri/BI-
sin(ji)*sin(ti)*dri^2*cos(ri)*LN/BI+sin(ji)*cos(ri)*LN*ddji/BI/SQR3+dj
i^2*cos(ji)*cos(ri)*LN/BI/SQR3-
2*dji*sin(ji)*dri*sin(ri)*LN/BI/SQR3+cos(ji)*sin(ri)*LN*ddri/BI/SQR3+c
os(ji)*dri^2*cos(ri)*LN/BI/SQR3+cos(ji)*sin(ti)*sin(ri)*LN*ddji/BI/SQR
3-
dji^2*sin(ji)*sin(ti)*sin(ri)*LN/BI/SQR3+2*dji*cos(ji)*dti*cos(ti)*sin
(ri)*LN/BI/SQR3+2*dji*cos(ji)*sin(ti)*dri*cos(ri)*LN/BI/SQR3+sin(ji)*c
os(ti)*sin(ri)*LN*ddti/BI/SQR3-
sin(ji)*dti^2*sin(ti)*sin(ri)*LN/BI/SQR3+2*sin(ji)*dti*cos(ti)*dri*cos
(ri)*LN/BI/SQR3+sin(ji)*sin(ti)*cos(ri)*LN*ddri/BI/SQR3-
sin(ji)*sin(ti)*dri^2*sin(ri)*LN/BI/SQR3-cos(PI*BI/HIRU+PI)*sin(-
q5)*LR*ddq5+cos(PI*BI/HIRU+PI)*dq5^2*cos(-q5)*LR;
    ddz+cos(ji)*sin(ri)*LN*ddji/BI-
dji^2*sin(ji)*sin(ri)*LN/BI+2*dji*cos(ji)*dri*cos(ri)*LN/BI+sin(ji)*co
s(ri)*LN*ddri/BI-
sin(ji)*dri^2*sin(ri)*LN/BI+sin(ji)*sin(ti)*cos(ri)*LN*ddji/BI+dji^2*c
os(ji)*sin(ti)*cos(ri)*LN/BI+2*dji*sin(ji)*dti*cos(ti)*cos(ri)*LN/BI-
2*dji*sin(ji)*sin(ti)*dri*sin(ri)*LN/BI-
cos(ji)*cos(ti)*cos(ri)*LN*ddti/BI+cos(ji)*dti^2*sin(ti)*cos(ri)*LN/BI
+2*cos(ji)*dti*cos(ti)*dri*sin(ri)*LN/BI+cos(ji)*sin(ti)*sin(ri)*LN*dd
ri/BI+cos(ji)*sin(ti)*dri^2*cos(ri)*LN/BI-
cos(ji)*cos(ri)*LN*ddji/BI/SQR3+dji^2*sin(ji)*cos(ri)*LN/BI/SQR3+2*dji
*cos(ji)*dri*sin(ri)*LN/BI/SQR3+sin(ji)*sin(ri)*LN*ddri/BI/SQR3+sin(ji
)*dri^2*cos(ri)*LN/BI/SQR3+sin(ji)*sin(ti)*sin(ri)*LN*ddji/BI/SQR3+dji
^2*cos(ji)*sin(ti)*sin(ri)*LN/BI/SQR3+2*dji*sin(ji)*dti*cos(ti)*sin(ri
)*LN/BI/SQR3+2*dji*sin(ji)*sin(ti)*dri*cos(ri)*LN/BI/SQR3-
cos(ji)*cos(ti)*sin(ri)*LN*ddti/BI/SQR3+cos(ji)*dti^2*sin(ti)*sin(ri)*
LN/BI/SQR3-2*cos(ji)*dti*cos(ti)*dri*cos(ri)*LN/BI/SQR3-
cos(ji)*sin(ti)*cos(ri)*LN*ddri/BI/SQR3+cos(ji)*sin(ti)*dri^2*sin(ri)*
LN/BI/SQR3-ddq2+cos(-q5)*LR*ddq5+dq5^2*sin(-q5)*LR;
%
    ddx+sin(ti)*sin(ri)*LN*ddti/SQR3+dti^2*cos(ti)*sin(ri)*LN/SQR3+2*dti*s
in(ti)*dri*cos(ri)*LN/SQR3-
cos(ti)*cos(ri)*LN*ddri/SQR3+cos(ti)*dri^2*sin(ri)*LN/SQR3+sin(-
PI*BI/HIRU+PI)*sin(-q6)*LR*ddq6-sin(-PI*BI/HIRU+PI)*dq6^2*cos(-q6)*LR;
    ddy-sin(ji)*cos(ri)*LN*ddji/SQR3-
dji^2*cos(ji)*cos(ri)*LN/SQR3+2*dji*sin(ji)*dri*sin(ri)*LN/SQR3-
cos(ji)*sin(ri)*LN*ddri/SQR3-cos(ji)*dri^2*cos(ri)*LN/SQR3-
cos(ji)*sin(ti)*sin(ri)*LN*ddji/SQR3+dji^2*sin(ji)*sin(ti)*sin(ri)*LN/
SQR3-2*dji*cos(ji)*dti*cos(ti)*sin(ri)*LN/SQR3-

```

```

2*dji*cos(ji)*sin(ti)*dri*cos(ri)*LN/SQR3-
sin(ji)*cos(ti)*sin(ri)*LN*ddti/SQR3+sin(ji)*dti^2*sin(ti)*sin(ri)*LN/
SQR3-2*sin(ji)*dti*cos(ti)*dri*cos(ri)*LN/SQR3-
sin(ji)*sin(ti)*cos(ri)*LN*ddri/SQR3+sin(ji)*sin(ti)*dri^2*sin(ri)*LN/
SQR3-cos(-PI*BI/HIRU+PI)*sin(-q6)*LR*ddq6+cos(-
PI*BI/HIRU+PI)*dq6^2*cos(-q6)*LR;
    ddz+cos(ji)*cos(ri)*LN*ddji/SQR3-dji^2*sin(ji)*cos(ri)*LN/SQR3-
2*dji*cos(ji)*dri*sin(ri)*LN/SQR3-sin(ji)*sin(ri)*LN*ddri/SQR3-
sin(ji)*dri^2*cos(ri)*LN/SQR3-sin(ji)*sin(ti)*sin(ri)*LN*ddji/SQR3-
dji^2*cos(ji)*sin(ti)*sin(ri)*LN/SQR3-
2*dji*sin(ji)*dti*cos(ti)*sin(ri)*LN/SQR3-
2*dji*sin(ji)*sin(ti)*dri*cos(ri)*LN/SQR3+cos(ji)*cos(ti)*sin(ri)*LN*d
dti/SQR3-
cos(ji)*dti^2*sin(ti)*sin(ri)*LN/SQR3+2*cos(ji)*dti*cos(ti)*dri*cos(ri
)*LN/SQR3+cos(ji)*sin(ti)*cos(ri)*LN*ddri/SQR3-
cos(ji)*sin(ti)*dri^2*sin(ri)*LN/SQR3-ddq3+cos(-
q6)*LR*ddq6+dq6^2*sin(-q6)*LR;

];
%
fprintf('\n');
%
disp('Kinematic Equations SWALLOWED');

```

9.2.5. dynamic_eq_origin.m

```

%-----
% DYNAMIC EQUATIONS
%-----

fprintf('Swallowing Dynamic equations. (In Origin)\n')
fprintf('.....\n')
%
fprintf('.');
%
eq_01=-ddq1*mB1F+cos(-q4)*ddq4*mB1Fy+dq4^2*sin(-q4)*mB1Fy-sin(-
q4)*ddq4*mB1Fz+dq4^2*cos(-q4)*mB1Fz-mB1F*g+FORCE1z-sin(-q4)*FS1y-cos(-
q4)*FS1z-dq1*VISP1-FCCP1z*sign_dq1;
%
fprintf('.');
%
eq_02=-ddq2*mB2F+cos(-q5)*ddq5*mB2Fy+dq5^2*sin(-q5)*mB2Fy-sin(-
q5)*ddq5*mB2Fz+dq5^2*cos(-q5)*mB2Fz-mB2F*g+FORCE2z-sin(-q5)*FS2y-cos(-
q5)*FS2z-dq2*VISP2-FCCP2z*sign_dq2;
%
fprintf('.');
%
eq_03=-ddq3*mB3F+cos(-q6)*ddq6*mB3Fy+dq6^2*sin(-q6)*mB3Fy-sin(-
q6)*ddq6*mB3Fz+dq6^2*cos(-q6)*mB3Fz-mB3F*g+FORCE3z-sin(-q6)*FS3y-cos(-
q6)*FS3z-dq3*VISP3-FCCP3z*sign_dq3;
%
fprintf('.');
%
eq_04=-cos(PI)^2*ddq4*IB1Fxx+cos(PI)^2*cos(-q4)*ddq1*mB1Fy-
cos(PI)^2*sin(-q4)*ddq1*mB1Fz-sin(PI)^2*ddq4*IB1Fxx+sin(PI)^2*cos(-
q4)*ddq1*mB1Fy-sin(PI)^2*sin(-q4)*ddq1*mB1Fz+cos(-q4)*mB1Fy*g-sin(-
q4)*mB1Fz*g+sin(-q4)^2*LR*FS1z+cos(-q4)^2*LR*FS1z-dq4*VISR1-
MCCR1z*sign_dq4-cos(PI)*sin(ti)*dri*VISS1-cos(PI)*dji*VISS1-
cos(PI)^2*dq4*VISS1-

```

```

sin(PI)*cos(ji)*dti*VISS1+sin(PI)*sin(ji)*cos(ti)*dri*VISS1-
sin(PI)^2*dq4*VISS1-cos(PI)*sin(ti)*dri*MCCS1/omega_bar1-
cos(PI)*dji*MCCS1/omega_bar1-cos(PI)^2*dq4*MCCS1/omega_bar1-
sin(PI)*cos(ji)*dti*MCCS1/omega_bar1+sin(PI)*sin(ji)*cos(ti)*dri*MCCS1
/omega_bar1-sin(PI)^2*dq4*MCCS1/omega_bar1;
%
fprintf(' ');
%
eq_05=-cos(PI*BI/HIRU+PI)^2*ddq5*IB2Fxx+cos(PI*BI/HIRU+PI)^2*cos(-
q5)*ddq2*mB2Fy-cos(PI*BI/HIRU+PI)^2*sin(-q5)*ddq2*mB2Fz-
sin(PI*BI/HIRU+PI)^2*ddq5*IB2Fxx+sin(PI*BI/HIRU+PI)^2*cos(-
q5)*ddq2*mB2Fy-sin(PI*BI/HIRU+PI)^2*sin(-q5)*ddq2*mB2Fz+cos(-
q5)*mB2Fy*g-sin(-q5)*mB2Fz*g+sin(-q5)^2*LR*FS2z+cos(-q5)^2*LR*FS2z-
dq5*VISR2-
MCCR2z*sign_dq5+cos(PI*BI/HIRU+PI)^2*dq5*VISS2+cos(PI*BI/HIRU+PI)*sin(
ti)*dri*VISS2+cos(PI*BI/HIRU+PI)*dji*VISS2+sin(PI*BI/HIRU+PI)^2*dq5*VI
SS2-
sin(PI*BI/HIRU+PI)*sin(ji)*cos(ti)*dri*VISS2+sin(PI*BI/HIRU+PI)*cos(ji
)*dti*VISS2-cos(PI*BI/HIRU+PI)*sin(ti)*dri*MCCS2/omega_bar2-
cos(PI*BI/HIRU+PI)*dji*MCCS2/omega_bar2-
cos(PI*BI/HIRU+PI)^2*dq5*MCCS2/omega_bar2-
sin(PI*BI/HIRU+PI)*cos(ji)*dti*MCCS2/omega_bar2+sin(PI*BI/HIRU+PI)*sin
(ji)*cos(ti)*dri*MCCS2/omega_bar2-
sin(PI*BI/HIRU+PI)^2*dq5*MCCS2/omega_bar2;
%
fprintf(' ');
%
eq_06=-cos(-PI*BI/HIRU+PI)^2*ddq6*IB3Fxx+cos(-PI*BI/HIRU+PI)^2*cos(-
q6)*ddq3*mB3Fy-cos(-PI*BI/HIRU+PI)^2*sin(-q6)*ddq3*mB3Fz-sin(-
PI*BI/HIRU+PI)^2*ddq6*IB3Fxx+sin(-PI*BI/HIRU+PI)^2*cos(-
q6)*ddq3*mB3Fy-sin(-PI*BI/HIRU+PI)^2*sin(-q6)*ddq3*mB3Fz+cos(-
q6)*mB3Fy*g-sin(-q6)*mB3Fz*g+sin(-q6)^2*LR*FS3z+cos(-q6)^2*LR*FS3z-
dq6*VISR3-MCCR3z*sign_dq6+cos(-PI*BI/HIRU+PI)^2*dq6*VISS3+cos(-
PI*BI/HIRU+PI)*sin(ti)*dri*VISS3+cos(-PI*BI/HIRU+PI)*dji*VISS3+sin(-
PI*BI/HIRU+PI)^2*dq6*VISS3-sin(-
PI*BI/HIRU+PI)*sin(ji)*cos(ti)*dri*VISS3+sin(-
PI*BI/HIRU+PI)*cos(ji)*dti*VISS3-cos(-
PI*BI/HIRU+PI)*sin(ti)*dri*MCCS3/omega_bar3-cos(-
PI*BI/HIRU+PI)*dji*MCCS3/omega_bar3-cos(-
PI*BI/HIRU+PI)^2*dq6*MCCS3/omega_bar3-sin(-
PI*BI/HIRU+PI)*cos(ji)*dti*MCCS3/omega_bar3+sin(-
PI*BI/HIRU+PI)*sin(ji)*cos(ti)*dri*MCCS3/omega_bar3-sin(-
PI*BI/HIRU+PI)^2*dq6*MCCS3/omega_bar3;
%
fprintf(' ');
%
eq_07=-ddx*mP+sin(ti)*cos(ri)*ddti*mPx+dti^2*cos(ti)*cos(ri)*mPx-
2*dti*sin(ti)*dri*sin(ri)*mPx+cos(ti)*sin(ri)*ddri*mPx+cos(ti)*dri^2*c
os(ri)*mPx-sin(ti)*sin(ri)*ddti*mPy-dti^2*cos(ti)*sin(ri)*mPy-
2*dti*sin(ti)*dri*cos(ri)*mPy+cos(ti)*cos(ri)*ddri*mPy-
cos(ti)*dri^2*sin(ri)*mPy-
cos(ti)*ddti*mPz+dti^2*sin(ti)*mPz+cos(PI)*FS1x-sin(PI)*cos(-
q4)*FS1y+sin(PI)*sin(-q4)*FS1z+cos(PI*BI/HIRU+PI)*FS2x-
sin(PI*BI/HIRU+PI)*cos(-q5)*FS2y+sin(PI*BI/HIRU+PI)*sin(-
q5)*FS2z+cos(-PI*BI/HIRU+PI)*FS3x-sin(-PI*BI/HIRU+PI)*cos(-
q6)*FS3y+sin(-PI*BI/HIRU+PI)*sin(-q6)*FS3z;
%
fprintf(' ');
%

```

```

eq_08=-
ddy*mP+sin(ji)*sin(ri)*ddji*mPx+dji^2*cos(ji)*sin(ri)*mPx+2*dji*sin(ji)
)*dri*cos(ri)*mPx-cos(ji)*cos(ri)*ddri*mPx+cos(ji)*dri^2*sin(ri)*mPx-
cos(ji)*sin(ti)*cos(ri)*ddji*mPx+dji^2*sin(ji)*sin(ti)*cos(ri)*mPx-
2*dji*cos(ji)*dti*cos(ti)*cos(ri)*mPx+2*dji*cos(ji)*sin(ti)*dri*sin(ri)
)*mPx-
sin(ji)*cos(ti)*cos(ri)*ddti*mPx+sin(ji)*dti^2*sin(ti)*cos(ri)*mPx+2*s
in(ji)*dti*cos(ti)*dri*sin(ri)*mPx+sin(ji)*sin(ti)*sin(ri)*ddri*mPx+si
n(ji)*sin(ti)*dri^2*cos(ri)*mPx+sin(ji)*cos(ri)*ddji*mPy+dji^2*cos(ji)
)*cos(ri)*mPy-
2*dji*sin(ji)*dri*sin(ri)*mPy+cos(ji)*sin(ri)*ddri*mPy+cos(ji)*dri^2*c
os(ri)*mPy+cos(ji)*sin(ti)*sin(ri)*ddji*mPy-
dji^2*sin(ji)*sin(ti)*sin(ri)*mPy+2*dji*cos(ji)*dti*cos(ti)*sin(ri)*mP
y+2*dji*cos(ji)*sin(ti)*dri*cos(ri)*mPy+sin(ji)*cos(ti)*sin(ri)*ddti*m
Py-
sin(ji)*dti^2*sin(ti)*sin(ri)*mPy+2*sin(ji)*dti*cos(ti)*dri*cos(ri)*mP
y+sin(ji)*sin(ti)*cos(ri)*ddri*mPy-
sin(ji)*sin(ti)*dri^2*sin(ri)*mPy+cos(ji)*cos(ti)*ddji*mPz-
dji^2*sin(ji)*cos(ti)*mPz-2*dji*cos(ji)*dti*sin(ti)*mPz-
sin(ji)*sin(ti)*ddti*mPz-
sin(ji)*dti^2*cos(ti)*mPz+sin(PI)*FS1x+cos(PI)*cos(-q4)*FS1y-
cos(PI)*sin(-q4)*FS1z+sin(PI*BI/HIRU+PI)*FS2x+cos(PI*BI/HIRU+PI)*cos(-
q5)*FS2y-cos(PI*BI/HIRU+PI)*sin(-q5)*FS2z+sin(-
PI*BI/HIRU+PI)*FS3x+cos(-PI*BI/HIRU+PI)*cos(-q6)*FS3y-cos(-
PI*BI/HIRU+PI)*sin(-q6)*FS3z;
%
fprintf(' ');
%
eq_09=-ddz*mP-cos(ji)*sin(ri)*ddji*mPx+dji^2*sin(ji)*sin(ri)*mPx-
2*dji*cos(ji)*dri*cos(ri)*mPx-
sin(ji)*cos(ri)*ddri*mPx+sin(ji)*dri^2*sin(ri)*mPx-
sin(ji)*sin(ti)*cos(ri)*ddji*mPx-dji^2*cos(ji)*sin(ti)*cos(ri)*mPx-
2*dji*sin(ji)*dti*cos(ti)*cos(ri)*mPx+2*dji*sin(ji)*sin(ti)*dri*sin(ri)
)*mPx+cos(ji)*cos(ti)*cos(ri)*ddti*mPx-
cos(ji)*dti^2*sin(ti)*cos(ri)*mPx-
2*cos(ji)*dti*cos(ti)*dri*sin(ri)*mPx-
cos(ji)*sin(ti)*sin(ri)*ddri*mPx-cos(ji)*sin(ti)*dri^2*cos(ri)*mPx-
cos(ji)*cos(ri)*ddji*mPy+dji^2*sin(ji)*cos(ri)*mPy+2*dji*cos(ji)*dri*s
in(ri)*mPy+sin(ji)*sin(ri)*ddri*mPy+sin(ji)*dri^2*cos(ri)*mPy+sin(ji)*
sin(ti)*sin(ri)*ddji*mPy+dji^2*cos(ji)*sin(ti)*sin(ri)*mPy+2*dji*sin(j
i)*dti*cos(ti)*sin(ri)*mPy+2*dji*sin(ji)*sin(ti)*dri*cos(ri)*mPy-
cos(ji)*cos(ti)*sin(ri)*ddti*mPy+cos(ji)*dti^2*sin(ti)*sin(ri)*mPy-
2*cos(ji)*dti*cos(ti)*dri*cos(ri)*mPy-
cos(ji)*sin(ti)*cos(ri)*ddri*mPy+cos(ji)*sin(ti)*dri^2*sin(ri)*mPy+sin
(ji)*cos(ti)*ddji*mPz+dji^2*cos(ji)*cos(ti)*mPz-
2*dji*sin(ji)*dti*sin(ti)*mPz+cos(ji)*sin(ti)*ddti*mPz+cos(ji)*dti^2*c
os(ti)*mPz-mP*g+sin(-q4)*FS1y+cos(-q4)*FS1z+sin(-q5)*FS2y+cos(-
q5)*FS2z+sin(-q6)*FS3y+cos(-q6)*FS3z;
%
fprintf(' ');
%
eq_10=-
cos(ti)^2*cos(ri)^2*ddji*IPxx+2*dji*dti*cos(ti)*sin(ti)*cos(ri)^2*IPxx
+2*dji*cos(ti)^2*dri*cos(ri)*sin(ri)*IPxx-
cos(ti)*cos(ri)*sin(ri)*ddti*IPxx-
dti*cos(ti)*dri*cos(ri)^2*IPxx+2*cos(ti)^2*cos(ri)*sin(ri)*ddji*IPxy-
4*dji*dti*cos(ti)*sin(ti)*cos(ri)*sin(ri)*IPxy+2*dji*cos(ti)^2*dri*cos
(ri)^2*IPxy-
cos(ti)*cos(ri)^2*ddti*IPxy+4*dti*cos(ti)*dri*cos(ri)*sin(ri)*IPxy-
cos(ti)*sin(ti)^2*cos(ri)*ddri*IPxz-

```

$2 * \cos (t i) * \sin (t i) * \cos (r i) * d d j i * I P x z - 2 * d j i * d t i * \cos (t i) ^ 2 * \cos (r i) * I P x z -$
 $\cos (t i) ^ 3 * \cos (r i) * d d r i * I P x z -$
 $2 * d j i * \cos (t i) ^ 2 * d r i * \sin (r i) ^ 2 * I P x y + \cos (t i) * \sin (r i) ^ 2 * d d t i * I P x y -$
 $\cos (t i) ^ 2 * \sin (r i) ^ 2 * d d j i * I P y y + 2 * d j i * d t i * \cos (t i) * \sin (t i) * \sin (r i) ^ 2 * I P y y$
 $-$
 $2 * d j i * \cos (t i) ^ 2 * d r i * \cos (r i) * \sin (r i) * I P y y + \cos (t i) * \cos (r i) * \sin (r i) * d d t i *$
 $I P y y -$
 $d t i * \cos (t i) * d r i * \sin (r i) ^ 2 * I P y y + \cos (t i) * \sin (t i) ^ 2 * \sin (r i) * d d r i * I P y z + 2 * c$
 $o s (t i) * \sin (t i) * \sin (r i) * d d j i * I P y z + 2 * d j i * d t i * \cos (t i) ^ 2 * \sin (r i) * I P y z + \cos ($
 $t i) ^ 3 * \sin (r i) * d d r i * I P y z + 2 * d j i * d t i * \sin (t i) ^ 2 * \cos (r i) * I P x z + 2 * d j i * \cos (t i)$
 $* \sin (t i) * d r i * \sin (r i) * I P x z - \sin (t i) * \sin (r i) * d d t i * I P x z -$
 $d t i * \sin (t i) * d r i * \cos (r i) * I P x z -$
 $2 * d j i * d t i * \sin (t i) ^ 2 * \sin (r i) * I P y z + 2 * d j i * \cos (t i) * \sin (t i) * d r i * \cos (r i) * I P y$
 $z - \sin (t i) * \cos (r i) * d d t i * I P y z + d t i * \sin (t i) * d r i * \sin (r i) * I P y z -$
 $\sin (t i) ^ 3 * d d r i * I P z z - \sin (t i) ^ 2 * d d j i * I P z z -$
 $2 * d j i * d t i * \cos (t i) * \sin (t i) * I P z z -$
 $\cos (t i) ^ 2 * \sin (t i) * d d r i * I P z z + d t i ^ 2 * \sin (t i) * \cos (r i) * \sin (r i) * I P x x + d t i ^ 2 * s$
 $i n (t i) * \cos (r i) ^ 2 * I P x y + d t i * \sin (t i) ^ 3 * d r i * \cos (r i) * I P x z + d t i * \cos (t i) ^ 2 * \sin$
 $(t i) * d r i * \cos (r i) * I P x z - d t i ^ 2 * \sin (t i) * \sin (r i) ^ 2 * I P x y -$
 $d t i ^ 2 * \sin (t i) * \cos (r i) * \sin (r i) * I P y y - d t i * \sin (t i) ^ 3 * d r i * \sin (r i) * I P y z -$
 $d t i * \cos (t i) ^ 2 * \sin (t i) * d r i * \sin (r i) * I P y z - d t i ^ 2 * \cos (t i) * \sin (r i) * I P x z -$
 $d t i ^ 2 * \cos (t i) * \cos (r i) * I P y z - d t i * \cos (t i) * \sin (t i) ^ 2 * d r i * I P z z -$
 $d t i * \cos (t i) ^ 3 * d r i * I P z z + d t i * \cos (t i) * d r i * \sin (r i) ^ 2 * I P x x + \cos (t i) * \sin (t i) ^ 2$
 $* d r i ^ 2 * \sin (r i) * I P x z + \cos (t i) ^ 3 * d r i ^ 2 * \sin (r i) * I P x z + d t i * \cos (t i) * d r i * \cos ($
 $r i) ^ 2 * I P y y + \cos (t i) * \sin (t i) ^ 2 * d r i ^ 2 * \cos (r i) * I P y z + \cos (t i) ^ 3 * d r i ^ 2 * \cos (r i)$
 $) * I P y z - \cos (j i) * \sin (r i) * d d z * m P x - \sin (j i) * \sin (t i) * \cos (r i) * d d z * m P x -$
 $\cos (j i) * \cos (r i) * d d z * m P y + \sin (j i) * \sin (t i) * \sin (r i) * d d z * m P y + \sin (j i) * \cos (t i)$
 $) * d d z * m P z + \sin (j i) * \sin (r i) * d d y * m P x -$
 $\cos (j i) * \sin (t i) * \cos (r i) * d d y * m P x + \sin (j i) * \cos (r i) * d d y * m P y + \cos (j i) * \sin (t i)$
 $) * \sin (r i) * d d y * m P y + \cos (j i) * \cos (t i) * d d y * m P z - \cos (j i) * \sin (r i) * m P x * g -$
 $\sin (j i) * \sin (t i) * \cos (r i) * m P x * g -$
 $\cos (j i) * \cos (r i) * m P y * g + \sin (j i) * \sin (t i) * \sin (r i) * m P y * g + \sin (j i) * \cos (t i) * m P$
 $z * g + \sin (P I) * \sin (j i) * \sin (r i) * L N * F S l x / B I -$
 $\sin (P I) * \cos (j i) * \sin (t i) * \cos (r i) * L N * F S l x / B I + \sin (P I) * \sin (j i) * \cos (r i) * L N *$
 $F S l x / B I / S Q R 3 + \sin (P I) * \cos (j i) * \sin (t i) * \sin (r i) * L N * F S l x / B I / S Q R 3 + \cos (P I) * c$
 $o s (-q 4) * \sin (j i) * \sin (r i) * L N * F S l y / B I - \cos (P I) * \cos (-$
 $q 4) * \cos (j i) * \sin (t i) * \cos (r i) * L N * F S l y / B I + \cos (P I) * \cos (-$
 $q 4) * \sin (j i) * \cos (r i) * L N * F S l y / B I / S Q R 3 + \cos (P I) * \cos (-$
 $q 4) * \cos (j i) * \sin (t i) * \sin (r i) * L N * F S l y / B I / S Q R 3 - \sin (-$
 $q 4) * \cos (j i) * \sin (r i) * L N * F S l y / B I - \sin (-$
 $q 4) * \sin (j i) * \sin (t i) * \cos (r i) * L N * F S l y / B I - \sin (-$
 $q 4) * \cos (j i) * \cos (r i) * L N * F S l y / B I / S Q R 3 + \sin (-$
 $q 4) * \sin (j i) * \sin (t i) * \sin (r i) * L N * F S l y / B I / S Q R 3 - \cos (P I) * \sin (-$
 $q 4) * \sin (j i) * \sin (r i) * L N * F S l z / B I + \cos (P I) * \sin (-$
 $q 4) * \cos (j i) * \sin (t i) * \cos (r i) * L N * F S l z / B I - \cos (P I) * \sin (-$
 $q 4) * \sin (j i) * \cos (r i) * L N * F S l z / B I / S Q R 3 - \cos (P I) * \sin (-$
 $q 4) * \cos (j i) * \sin (t i) * \sin (r i) * L N * F S l z / B I / S Q R 3 - \cos (-$
 $q 4) * \cos (j i) * \sin (r i) * L N * F S l z / B I - \cos (-$
 $q 4) * \sin (j i) * \sin (t i) * \cos (r i) * L N * F S l z / B I - \cos (-$
 $q 4) * \cos (j i) * \cos (r i) * L N * F S l z / B I / S Q R 3 + \cos (-$
 $q 4) * \sin (j i) * \sin (t i) * \sin (r i) * L N * F S l z / B I / S Q R 3 -$
 $\cos (P I) ^ 2 * \sin (t i) * d r i * V I S S 1 - \cos (P I) ^ 2 * d j i * V I S S 1 - \cos (P I) ^ 3 * d q 4 * V I S S 1 -$
 $\cos (P I) * \sin (P I) * \cos (j i) * d t i * V I S S 1 + \cos (P I) * \sin (P I) * \sin (j i) * \cos (t i) * d r i *$
 $V I S S 1 - \cos (P I) * \sin (P I) ^ 2 * d q 4 * V I S S 1 - \sin (P I) ^ 2 * \cos (-$
 $q 4) ^ 2 * \sin (t i) * d r i * V I S S 1 - \sin (P I) ^ 2 * \cos (-$
 $q 4) ^ 2 * d j i * V I S S 1 + \cos (P I) * \sin (P I) * \cos (-q 4) ^ 2 * \cos (j i) * d t i * V I S S 1 -$
 $\cos (P I) * \sin (P I) * \cos (-q 4) ^ 2 * \sin (j i) * \cos (t i) * d r i * V I S S 1 - \sin (P I) ^ 2 * \sin (-$
 $q 4) ^ 2 * \sin (t i) * d r i * V I S S 1 - \sin (P I) ^ 2 * \sin (-$
 $q 4) ^ 2 * d j i * V I S S 1 + \cos (P I) * \sin (P I) * \sin (-q 4) ^ 2 * \cos (j i) * d t i * V I S S 1 -$
 $\cos (P I) * \sin (P I) * \sin (-q 4) ^ 2 * \sin (j i) * \cos (t i) * d r i * V I S S 1 -$


```

q5)^2*dji*MCCS2/omega_bar2+cos(PI*BI/HIRU+PI)*sin(PI*BI/HIRU+PI)*sin(-
q5)^2*cos(ji)*dti*MCCS2/omega_bar2-
cos(PI*BI/HIRU+PI)*sin(PI*BI/HIRU+PI)*sin(-
q5)^2*sin(ji)*cos(ti)*dri*MCCS2/omega_bar2-sin(-
PI*BI/HIRU+PI)*sin(ji)*cos(ri)*LN*FS3x/SQR3-sin(-
PI*BI/HIRU+PI)*cos(ji)*sin(ti)*sin(ri)*LN*FS3x/SQR3-cos(-
PI*BI/HIRU+PI)*cos(-q6)*sin(ji)*cos(ri)*LN*FS3y/SQR3-cos(-
PI*BI/HIRU+PI)*cos(-q6)*cos(ji)*sin(ti)*sin(ri)*LN*FS3y/SQR3+sin(-
q6)*cos(ji)*cos(ri)*LN*FS3y/SQR3-sin(-
q6)*sin(ji)*sin(ti)*sin(ri)*LN*FS3y/SQR3+cos(-PI*BI/HIRU+PI)*sin(-
q6)*sin(ji)*cos(ri)*LN*FS3z/SQR3+cos(-PI*BI/HIRU+PI)*sin(-
q6)*cos(ji)*sin(ti)*sin(ri)*LN*FS3z/SQR3+cos(-
q6)*cos(ji)*cos(ri)*LN*FS3z/SQR3-cos(-
q6)*sin(ji)*sin(ti)*sin(ri)*LN*FS3z/SQR3+cos(-
PI*BI/HIRU+PI)^3*dq6*VISS3+cos(-
PI*BI/HIRU+PI)^2*sin(ti)*dri*VISS3+cos(-
PI*BI/HIRU+PI)^2*dji*VISS3+cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)^2*dq6*VISS3-cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)*sin(ji)*cos(ti)*dri*VISS3+cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)*cos(ji)*dti*VISS3+sin(-PI*BI/HIRU+PI)^2*cos(-
q6)^2*sin(ti)*dri*VISS3+sin(-PI*BI/HIRU+PI)^2*cos(-
q6)^2*dji*VISS3+cos(-PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)*cos(-
q6)^2*sin(ji)*cos(ti)*dri*VISS3-cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)*cos(-q6)^2*cos(ji)*dti*VISS3+cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)^2*dq6*sin(-q6)^2*VISS3+sin(-PI*BI/HIRU+PI)^2*sin(-
q6)^2*sin(ti)*dri*VISS3+sin(-PI*BI/HIRU+PI)^2*sin(-q6)^2*dji*VISS3-
cos(-PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)^2*dq6*sin(-q6)^2*VISS3+cos(-
PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)*sin(-
q6)^2*sin(ji)*cos(ti)*dri*VISS3-cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)*sin(-q6)^2*cos(ji)*dti*VISS3-cos(-
PI*BI/HIRU+PI)^2*sin(ti)*dri*MCCS3/omega_bar3-cos(-
PI*BI/HIRU+PI)^2*dji*MCCS3/omega_bar3-cos(-
PI*BI/HIRU+PI)^3*dq6*MCCS3/omega_bar3-cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)*cos(ji)*dti*MCCS3/omega_bar3+cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)*sin(ji)*cos(ti)*dri*MCCS3/omega_bar3-cos(-
PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)^2*dq6*MCCS3/omega_bar3-sin(-
PI*BI/HIRU+PI)^2*cos(-q6)^2*sin(ti)*dri*MCCS3/omega_bar3-sin(-
PI*BI/HIRU+PI)^2*cos(-q6)^2*dji*MCCS3/omega_bar3+cos(-
PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)*cos(-
q6)^2*cos(ji)*dti*MCCS3/omega_bar3-cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)*cos(-q6)^2*sin(ji)*cos(ti)*dri*MCCS3/omega_bar3-sin(-
PI*BI/HIRU+PI)^2*sin(-q6)^2*sin(ti)*dri*MCCS3/omega_bar3-sin(-
PI*BI/HIRU+PI)^2*sin(-q6)^2*dji*MCCS3/omega_bar3+cos(-
PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)*sin(-
q6)^2*cos(ji)*dti*MCCS3/omega_bar3-cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)*sin(-q6)^2*sin(ji)*cos(ti)*dri*MCCS3/omega_bar3;
%
fprintf(' ');
%
eq_11=-
cos(ji)^2*cos(ti)*cos(ri)*sin(ri)*ddji*IPxx+dji*cos(ji)^2*cos(ti)*dri*
sin(ri)^2*IPxx-cos(ji)^2*sin(ri)^2*ddti*IPxx-
cos(ji)^2*dti*dri*cos(ri)*sin(ri)*IPxx+cos(ji)^2*cos(ti)*sin(ri)^2*ddj
i*IPxy+2*dji*cos(ji)^2*cos(ti)*dri*cos(ri)*sin(ri)*IPxy-
2*cos(ji)^2*cos(ri)*sin(ri)*ddti*IPxy+cos(ji)^2*dti*dri*sin(ri)^2*IPxy
-cos(ji)^2*sin(ti)^2*sin(ri)*ddri*IPxz-
cos(ji)^2*sin(ti)*sin(ri)*ddji*IPxz-
cos(ji)^2*cos(ti)^2*sin(ri)*ddri*IPxz-
cos(ji)^2*cos(ti)*cos(ri)^2*ddji*IPxy-
cos(ji)^2*dti*dri*cos(ri)^2*IPxy+cos(ji)^2*cos(ti)*cos(ri)*sin(ri)*ddj

```

$i*IPyy+dji*\cos(ji)^2*\cos(ti)*dri*\cos(ri)^2*IPyy-$
 $\cos(ji)^2*\cos(ri)^2*ddti*IPyy+\cos(ji)^2*dti*dri*\cos(ri)*\sin(ri)*IPyy-$
 $\cos(ji)^2*\sin(ti)^2*\cos(ri)*ddri*IPyz-$
 $\cos(ji)^2*\sin(ti)*\cos(ri)*ddji*IPyz-$
 $\cos(ji)^2*\cos(ti)^2*\cos(ri)*ddri*IPyz-$
 $dji*\cos(ji)^2*\cos(ti)^3*dri*\cos(ri)^2*IPxx-$
 $\cos(ji)^2*dti*\cos(ti)^2*dri*\cos(ri)*\sin(ri)*IPxx+2*dji*\cos(ji)^2*\cos(t$
 $i)^3*dri*\cos(ri)*\sin(ri)*IPxy-$
 $\cos(ji)^2*dti*\cos(ti)^2*dri*\cos(ri)^2*IPxy-$
 $2*\cos(ji)^2*\cos(ti)^2*\sin(ti)^2*dri^2*\cos(ri)*IPxz-$
 $2*dji*\cos(ji)^2*\cos(ti)^2*\sin(ti)*dri*\cos(ri)*IPxz-$
 $\cos(ji)^2*\cos(ti)^4*dri^2*\cos(ri)*IPxz+\cos(ji)^2*dti*\cos(ti)^2*dri*\sin$
 $(ri)^2*IPxy-$
 $dji*\cos(ji)^2*\cos(ti)^3*dri*\sin(ri)^2*IPyy+\cos(ji)^2*dti*\cos(ti)^2*dri$
 $*\cos(ri)*\sin(ri)*IPyy+2*\cos(ji)^2*\cos(ti)^2*\sin(ti)^2*dri^2*\sin(ri)*IP$
 $yz+2*dji*\cos(ji)^2*\cos(ti)^2*\sin(ti)*dri*\sin(ri)*IPyz+\cos(ji)^2*\cos(ti$
 $)^4*dri^2*\sin(ri)*IPyz-$
 $dji*\cos(ji)^2*\cos(ti)*\sin(ti)^2*dri*\cos(ri)^2*IPxx-$
 $\cos(ji)^2*dti*\sin(ti)^2*dri*\cos(ri)*\sin(ri)*IPxx+2*dji*\cos(ji)^2*\cos(t$
 $i)*\sin(ti)^2*dri*\cos(ri)*\sin(ri)*IPxy-$
 $\cos(ji)^2*dti*\sin(ti)^2*dri*\cos(ri)^2*IPxy-$
 $\cos(ji)^2*\sin(ti)^4*dri^2*\cos(ri)*IPxz-$
 $2*dji*\cos(ji)^2*\sin(ti)^3*dri*\cos(ri)*IPxz+\cos(ji)^2*dti*\sin(ti)^2*dri$
 $*\sin(ri)^2*IPxy-$
 $dji*\cos(ji)^2*\cos(ti)*\sin(ti)^2*dri*\sin(ri)^2*IPyy+\cos(ji)^2*dti*\sin(t$
 $i)^2*dri*\cos(ri)*\sin(ri)*IPyy+\cos(ji)^2*\sin(ti)^4*dri^2*\sin(ri)*IPyz+2$
 $*dji*\cos(ji)^2*\sin(ti)^3*dri*\sin(ri)*IPyz-$
 $dji^2*\cos(ji)^2*\cos(ti)*\sin(ti)*\cos(ri)^2*IPxx+2*dji^2*\cos(ji)^2*\cos(t$
 $i)*\sin(ti)*\cos(ri)*\sin(ri)*IPxy-$
 $dji^2*\cos(ji)^2*\sin(ti)^2*\cos(ri)*IPxz-$
 $dji^2*\cos(ji)^2*\cos(ti)*\sin(ti)*\sin(ri)^2*IPyy+dji^2*\cos(ji)^2*\sin(ti)$
 $^2*\sin(ri)*IPyz+dji^2*\cos(ji)^2*\cos(ti)^2*\cos(ri)*IPxz-$
 $dji^2*\cos(ji)^2*\cos(ti)^2*\sin(ri)*IPyz+dji*\cos(ji)^2*\cos(ti)*\sin(ti)^2$
 $*dri*IPzz+dji^2*\cos(ji)^2*\cos(ti)*\sin(ti)*IPzz+dji*\cos(ji)^2*\cos(ti)^3$
 $*dri*IPzz+\cos(ji)^2*\sin(ti)*\cos(ri)*ddx*mPx-$
 $\cos(ji)^2*\sin(ti)*\sin(ri)*ddx*mPy-$
 $\cos(ji)^2*\cos(ti)*ddx*mPz+\cos(ji)*\cos(ti)*\cos(ri)*ddz*mPx-$
 $\cos(ji)*\cos(ti)*\sin(ri)*ddz*mPy+\cos(ji)*\sin(ti)*ddz*mPz-$
 $\sin(ji)^2*\cos(ti)*\cos(ri)*\sin(ri)*ddji*IPxx+dji*\sin(ji)^2*\cos(ti)*dri*$
 $\sin(ri)^2*IPxx-\sin(ji)^2*\sin(ri)^2*ddti*IPxx-$
 $\sin(ji)^2*dti*dri*\cos(ri)*\sin(ri)*IPxx+\sin(ji)^2*\cos(ti)*\sin(ri)^2*ddj$
 $i*IPxy+2*dji*\sin(ji)^2*\cos(ti)*dri*\cos(ri)*\sin(ri)*IPxy-$
 $2*\sin(ji)^2*\cos(ri)*\sin(ri)*ddti*IPxy+\sin(ji)^2*dti*dri*\sin(ri)^2*IPxy$
 $-\sin(ji)^2*\sin(ti)^2*\sin(ri)*ddri*IPxz-$
 $\sin(ji)^2*\sin(ti)*\sin(ri)*ddji*IPxz-$
 $\sin(ji)^2*\cos(ti)^2*\sin(ri)*ddri*IPxz-$
 $\sin(ji)^2*\cos(ti)*\cos(ri)^2*ddji*IPxy-$
 $\sin(ji)^2*dti*dri*\cos(ri)^2*IPxy+\sin(ji)^2*\cos(ti)*\cos(ri)*\sin(ri)*ddj$
 $i*IPyy+dji*\sin(ji)^2*\cos(ti)*dri*\cos(ri)^2*IPyy-$
 $\sin(ji)^2*\cos(ri)^2*ddti*IPyy+\sin(ji)^2*dti*dri*\cos(ri)*\sin(ri)*IPyy-$
 $\sin(ji)^2*\sin(ti)^2*\cos(ri)*ddri*IPyz-$
 $\sin(ji)^2*\sin(ti)*\cos(ri)*ddji*IPyz-$
 $\sin(ji)^2*\cos(ti)^2*\cos(ri)*ddri*IPyz-$
 $dji*\sin(ji)^2*\cos(ti)^3*dri*\cos(ri)^2*IPxx-$
 $\sin(ji)^2*dti*\cos(ti)^2*dri*\cos(ri)*\sin(ri)*IPxx+2*dji*\sin(ji)^2*\cos(t$
 $i)^3*dri*\cos(ri)*\sin(ri)*IPxy-$
 $\sin(ji)^2*dti*\cos(ti)^2*dri*\cos(ri)^2*IPxy-$
 $2*\sin(ji)^2*\cos(ti)^2*\sin(ti)^2*dri^2*\cos(ri)*IPxz-$
 $2*dji*\sin(ji)^2*\cos(ti)^2*\sin(ti)*dri*\cos(ri)*IPxz-$
 $\sin(ji)^2*\cos(ti)^4*dri^2*\cos(ri)*IPxz+\sin(ji)^2*dti*\cos(ti)^2*dri*\sin$


```

PI*BI/HIRU+PI)*cos(-q6)^2*dji*cos(ji)*MCCS3/omega_bar3+cos(-
PI*BI/HIRU+PI)^2*sin(-PI*BI/HIRU+PI)*dq6*cos(-
q6)^2*cos(ji)*MCCS3/omega_bar3-cos(-PI*BI/HIRU+PI)^2*cos(-
q6)^2*cos(ji)^2*dti*MCCS3/omega_bar3+cos(-PI*BI/HIRU+PI)^2*cos(-
q6)^2*cos(ji)*sin(ji)*cos(ti)*dri*MCCS3/omega_bar3-cos(-
PI*BI/HIRU+PI)^2*sin(-PI*BI/HIRU+PI)*dq6*cos(-
q6)^2*cos(ji)*MCCS3/omega_bar3-sin(-
q6)^2*sin(ji)^2*dti*MCCS3/omega_bar3-sin(-
q6)^2*cos(ji)*sin(ji)*cos(ti)*dri*MCCS3/omega_bar3+cos(-
PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)*sin(-
q6)^2*cos(ji)*sin(ti)*dri*MCCS3/omega_bar3+cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)*sin(-q6)^2*dji*cos(ji)*MCCS3/omega_bar3-cos(-
PI*BI/HIRU+PI)^2*sin(-q6)^2*cos(ji)^2*dti*MCCS3/omega_bar3+cos(-
PI*BI/HIRU+PI)^2*sin(-
q6)^2*cos(ji)*sin(ji)*cos(ti)*dri*MCCS3/omega_bar3-cos(-
q6)^2*sin(ji)^2*dti*MCCS3/omega_bar3-cos(-
q6)^2*cos(ji)*sin(ji)*cos(ti)*dri*MCCS3/omega_bar3;
%
fprintf(' ');
%
eq_12=-
cos(ti)^2*sin(ti)*cos(ri)^2*ddji*IPxx+2*dji*dti*cos(ti)*sin(ti)^2*cos(
ri)^2*IPxx+2*dji*cos(ti)^2*sin(ti)*dri*cos(ri)*sin(ri)*IPxx-
cos(ti)*sin(ti)*cos(ri)*sin(ri)*ddti*IPxx-
dti*cos(ti)*sin(ti)*dri*cos(ri)^2*IPxx+2*cos(ti)^2*sin(ti)*cos(ri)*sin
(ri)*ddji*IPxy-
4*dji*dti*cos(ti)*sin(ti)^2*cos(ri)*sin(ri)*IPxy+2*dji*cos(ti)^2*sin(t
i)*dri*cos(ri)^2*IPxy-
cos(ti)*sin(ti)*cos(ri)^2*ddti*IPxy+4*dti*cos(ti)*sin(ti)*dri*cos(ri)*
sin(ri)*IPxy-cos(ti)*sin(ti)^3*cos(ri)*ddri*IPxz-
2*cos(ti)*sin(ti)^2*cos(ri)*ddji*IPxz-
2*dji*dti*cos(ti)^2*sin(ti)*cos(ri)*IPxz-
cos(ti)^3*sin(ti)*cos(ri)*ddri*IPxz-
2*dji*cos(ti)^2*sin(ti)*dri*sin(ri)^2*IPxy+cos(ti)*sin(ti)*sin(ri)^2*d
dti*IPxy-
cos(ti)^2*sin(ti)*sin(ri)^2*ddji*IPyy+2*dji*dti*cos(ti)*sin(ti)^2*sin(
ri)^2*IPyy-
2*dji*cos(ti)^2*sin(ti)*dri*cos(ri)*sin(ri)*IPyy+cos(ti)*sin(ti)*cos(r
i)*sin(ri)*ddti*IPyy-
dti*cos(ti)*sin(ti)*dri*sin(ri)^2*IPyy+cos(ti)*sin(ti)^3*sin(ri)*ddri*
IPyz+2*cos(ti)*sin(ti)^2*sin(ri)*ddji*IPyz+2*dji*dti*cos(ti)^2*sin(ti)
*sin(ri)*IPyz+cos(ti)^3*sin(ti)*sin(ri)*ddri*IPyz+2*dji*dti*sin(ti)^3*
cos(ri)*IPxz+2*dji*cos(ti)*sin(ti)^2*dri*sin(ri)*IPxz-
sin(ti)^2*sin(ri)*ddti*IPxz-dti*sin(ti)^2*dri*cos(ri)*IPxz-
2*dji*dti*sin(ti)^3*sin(ri)*IPyz+2*dji*cos(ti)*sin(ti)^2*dri*cos(ri)*I
Pyz-sin(ti)^2*cos(ri)*ddti*IPyz+dti*sin(ti)^2*dri*sin(ri)*IPyz-
sin(ti)^4*ddri*IPzz-sin(ti)^3*ddji*IPzz-
2*dji*dti*cos(ti)*sin(ti)^2*IPzz-
cos(ti)^2*sin(ti)^2*ddri*IPzz+dti^2*sin(ti)^2*cos(ri)*sin(ri)*IPxx+dti
^2*sin(ti)^2*cos(ri)^2*IPxy+dti*sin(ti)^4*dri*cos(ri)*IPxz+dti*cos(ti)
^2*sin(ti)^2*dri*cos(ri)*IPxz-dti^2*sin(ti)^2*sin(ri)^2*IPxy-
dti^2*sin(ti)^2*cos(ri)*sin(ri)*IPyy-dti*sin(ti)^4*dri*sin(ri)*IPyz-
dti*cos(ti)^2*sin(ti)^2*dri*sin(ri)*IPyz-
dti^2*cos(ti)*sin(ti)*sin(ri)*IPxz-dti^2*cos(ti)*sin(ti)*cos(ri)*IPyz-
dti*cos(ti)*sin(ti)^3*dri*IPzz-
dti*cos(ti)^3*sin(ti)*dri*IPzz+dti*cos(ti)*sin(ti)*dri*sin(ri)^2*IPxx+
cos(ti)*sin(ti)^3*dri^2*sin(ri)*IPxz+cos(ti)^3*sin(ti)*dri^2*sin(ri)*I
Pxz+dti*cos(ti)*sin(ti)*dri*cos(ri)^2*IPyy+cos(ti)*sin(ti)^3*dri^2*cos
(ri)*IPyz+cos(ti)^3*sin(ti)*dri^2*cos(ri)*IPyz-
cos(ji)*sin(ti)*sin(ri)*ddz*mPx-sin(ji)*sin(ti)^2*cos(ri)*ddz*mPx-

```

$\cos(ji) * \sin(ti) * \cos(ri) * d dz * mPy + \sin(ji) * \sin(ti) ^2 * \sin(ri) * d dz * mPy + \sin(ji) * \sin(ti) * \sin(ri) * d dy * mPx -$
 $\cos(ji) * \sin(ti) ^2 * \cos(ri) * d dy * mPx + \sin(ji) * \sin(ti) * \cos(ri) * d dy * mPy + \cos(ji) * \sin(ti) ^2 * \sin(ri) * d dy * mPy + \cos(ji) * \sin(ji) * \cos(ti) ^2 * \cos(ri) * \sin(ri) * d dji * IPxx + \sin(ji) ^2 * \cos(ti) ^2 * \sin(ti) * \cos(ri) ^2 * d dji * IPxx -$
 $dji * \sin(ji) ^2 * d ti * \cos(ti) * \sin(ti) ^2 * \cos(ri) ^2 * IPxx -$
 $2 * dji * \sin(ji) ^2 * \cos(ti) ^2 * \sin(ti) * dri * \cos(ri) * \sin(ri) * IPxx + \sin(ji) ^2 * \cos(ti) * \sin(ti) * \cos(ri) * \sin(ri) * d d ti * IPxx + \sin(ji) ^2 * d ti * \cos(ti) * \sin(ti) * dri * \cos(ri) ^2 * IPxx -$
 $2 * \sin(ji) ^2 * \cos(ti) ^2 * \sin(ti) * \cos(ri) * \sin(ri) * d dji * IPxy + 2 * dji * \sin(ji) ^2 * d ti * \cos(ti) * \sin(ti) ^2 * \cos(ri) * \sin(ri) * IPxy -$
 $2 * dji * \sin(ji) ^2 * \cos(ti) ^2 * \sin(ti) * dri * \cos(ri) ^2 * IPxy + \sin(ji) ^2 * \cos(ti) * \sin(ti) * \cos(ri) ^2 * d d ti * IPxy -$
 $4 * \sin(ji) ^2 * d ti * \cos(ti) * \sin(ti) * dri * \cos(ri) * \sin(ri) * IPxy + \sin(ji) ^2 * \cos(ti) * \sin(ti) ^3 * \cos(ri) * d d ri * IPxz + \sin(ji) ^2 * \cos(ti) * \sin(ti) ^2 * \cos(ri) * d dji * IPxz + 4 * dji * \sin(ji) ^2 * d ti * \cos(ti) ^2 * \sin(ti) * \cos(ri) * IPxz + \sin(ji) ^2 * \cos(ti) ^3 * \sin(ti) * \cos(ri) * d d ri * IPxz + 2 * dji * \sin(ji) ^2 * \cos(ti) ^2 * \sin(ti) * dri * \sin(ri) ^2 * IPxy -$
 $\sin(ji) ^2 * \cos(ti) * \sin(ti) * \sin(ri) ^2 * d d ti * IPxy + \sin(ji) ^2 * \cos(ti) ^2 * \sin(ti) * \sin(ri) ^2 * d dji * IPyy -$
 $dji * \sin(ji) ^2 * d ti * \cos(ti) * \sin(ti) ^2 * \sin(ri) ^2 * IPyy + 2 * dji * \sin(ji) ^2 * \cos(ti) ^2 * \sin(ti) * dri * \cos(ri) * \sin(ri) * IPyy -$
 $\sin(ji) ^2 * \cos(ti) * \sin(ti) * \cos(ri) * \sin(ri) * d d ti * IPyy + \sin(ji) ^2 * d ti * \cos(ti) * \sin(ti) * dri * \sin(ri) ^2 * IPyy -$
 $\sin(ji) ^2 * \cos(ti) * \sin(ti) ^3 * \sin(ri) * d d ri * IPyz -$
 $\sin(ji) ^2 * \cos(ti) * \sin(ti) ^2 * \sin(ri) * d dji * IPyz -$
 $4 * dji * \sin(ji) ^2 * d ti * \cos(ti) ^2 * \sin(ti) * \sin(ri) * IPyz -$
 $\sin(ji) ^2 * \cos(ti) ^3 * \sin(ti) * \sin(ri) * d d ri * IPyz -$
 $\sin(ji) ^2 * \cos(ti) ^3 * \cos(ri) * d dji * IPxz -$
 $\sin(ji) ^2 * \cos(ti) ^2 * \sin(ri) * d d ti * IPxz -$
 $\sin(ji) ^2 * d ti * \cos(ti) ^2 * dri * \cos(ri) * IPxz + \sin(ji) ^2 * \cos(ti) ^3 * \sin(ri) * d dji * IPyz -$
 $\sin(ji) ^2 * \cos(ti) ^2 * \cos(ri) * d d ti * IPyz + \sin(ji) ^2 * d ti * \cos(ti) ^2 * dri * \sin(ri) * IPyz - \sin(ji) ^2 * \cos(ti) ^2 * \sin(ti) ^2 * d d ri * IPzz -$
 $\sin(ji) ^2 * \cos(ti) ^2 * \sin(ti) * d dji * IPzz -$
 $dji * \sin(ji) ^2 * d ti * \cos(ti) ^3 * IPzz - \sin(ji) ^2 * \cos(ti) ^4 * d d ri * IPzz -$
 $\sin(ji) ^2 * d ti * \cos(ti) * \sin(ti) * dri * \sin(ri) ^2 * IPxx -$
 $\sin(ji) ^2 * \cos(ti) * \sin(ti) ^3 * dri ^2 * \sin(ri) * IPxz -$
 $2 * dji * \sin(ji) ^2 * \cos(ti) * \sin(ti) ^2 * dri * \sin(ri) * IPxz -$
 $\sin(ji) ^2 * \cos(ti) ^3 * \sin(ti) * dri ^2 * \sin(ri) * IPxz -$
 $\sin(ji) ^2 * d ti * \cos(ti) * \sin(ti) * dri * \cos(ri) ^2 * IPyy -$
 $\sin(ji) ^2 * \cos(ti) * \sin(ti) ^3 * dri ^2 * \cos(ri) * IPyz -$
 $2 * dji * \sin(ji) ^2 * \cos(ti) * \sin(ti) ^2 * dri * \cos(ri) * IPyz -$
 $\sin(ji) ^2 * \cos(ti) ^3 * \sin(ti) * dri ^2 * \cos(ri) * IPyz -$
 $dji ^2 * \sin(ji) ^2 * \cos(ti) ^2 * \cos(ri) * \sin(ri) * IPxx -$
 $dji * \sin(ji) ^2 * d ti * \cos(ti) * \sin(ri) ^2 * IPxx + dji ^2 * \sin(ji) ^2 * \cos(ti) ^2 * \sin(ri) ^2 * IPxy - 2 * dji * \sin(ji) ^2 * \cos(ti) * \sin(ri) * d ti * \cos(ti) * \cos(ri) * \sin(ri) * IPxy -$
 $dji ^2 * \sin(ji) ^2 * \cos(ti) * \sin(ti) * \sin(ri) * IPxz -$
 $dji ^2 * \sin(ji) ^2 * \cos(ti) ^2 * \cos(ri) ^2 * IPxy + dji ^2 * \sin(ji) ^2 * \cos(ti) ^2 * \cos(ri) * \sin(ri) * IPyy - dji * \sin(ji) ^2 * d ti * \cos(ti) * \cos(ri) ^2 * IPyy -$
 $dji ^2 * \sin(ji) ^2 * \cos(ti) * \sin(ti) * \cos(ri) * IPyz + dji * \sin(ji) ^2 * d ti * \cos(ti) ^3 * \cos(ri) ^2 * IPxx + \sin(ji) ^2 * d ti ^2 * \cos(ti) ^2 * \cos(ri) * \sin(ri) * IPxx -$
 $2 * dji * \sin(ji) ^2 * d ti * \cos(ti) ^3 * \cos(ri) * \sin(ri) * IPxy + \sin(ji) ^2 * d ti ^2 * \cos(ti) ^2 * \cos(ri) ^2 * IPxy + \sin(ji) ^2 * d ti * \cos(ti) ^2 * \sin(ti) ^2 * dri * \cos(ri) * IPxz + \sin(ji) ^2 * d ti * \cos(ti) ^4 * dri * \cos(ri) * IPxz -$
 $\sin(ji) ^2 * d ti ^2 * \cos(ti) ^2 * \sin(ri) ^2 * IPxy + dji * \sin(ji) ^2 * d ti * \cos(ti) ^3 * \sin(ri) ^2 * IPyy - \sin(ji) ^2 * d ti ^2 * \cos(ti) ^2 * \cos(ri) * \sin(ri) * IPyy -$
 $\sin(ji) ^2 * d ti * \cos(ti) ^2 * \sin(ti) ^2 * dri * \sin(ri) * IPyz -$
 $\sin(ji) ^2 * d ti * \cos(ti) ^4 * dri * \sin(ri) * IPyz + \sin(ji) ^2 * d ti ^2 * \cos(ti) * \sin(ti) * \cos(ri) * IPyz + \sin(ji) ^2$

$$\begin{aligned}
& *dti*\cos(ti)*\sin(ti)^3*dri*IPzz+dji*\sin(ji)^2*dti*\cos(ti)*\sin(ti)^2*IP \\
& zz+\sin(ji)^2*dti*\cos(ti)^3*\sin(ti)*dri*IPzz+\sin(ji)^2*\cos(ti)*\sin(ri)* \\
& ddx*mPx+\sin(ji)^2*\cos(ti)*\cos(ri)*ddx*mPy- \\
& \sin(ji)*\cos(ti)^2*\cos(ri)*ddz*mPx+\sin(ji)*\cos(ti)^2*\sin(ri)*ddz*mPy- \\
& \cos(ji)*\sin(ji)*\cos(ti)^2*\cos(ri)*\sin(ri)*ddji*IPxx+\cos(ji)^2*\cos(ti)^ \\
& 2*\sin(ti)*\cos(ri)^2*ddji*IPxx- \\
& dji*\cos(ji)^2*dti*\cos(ti)*\sin(ti)^2*\cos(ri)^2*IPxx- \\
& 2*dji*\cos(ji)^2*\cos(ti)^2*\sin(ti)*dri*\cos(ri)*\sin(ri)*IPxx+\cos(ji)^2*c \\
& \cos(ti)*\sin(ti)*\cos(ri)*\sin(ri)*ddti*IPxx+\cos(ji)^2*dti*\cos(ti)*\sin(ti) \\
& *dri*\cos(ri)^2*IPxx- \\
& 2*\cos(ji)^2*\cos(ti)^2*\sin(ti)*\cos(ri)*\sin(ri)*ddji*IPxy+2*dji*\cos(ji)^ \\
& 2*dti*\cos(ti)*\sin(ti)^2*\cos(ri)*\sin(ri)*IPxy- \\
& 2*dji*\cos(ji)^2*\cos(ti)^2*\sin(ti)*dri*\cos(ri)^2*IPxy+\cos(ji)^2*\cos(ti) \\
& *\sin(ti)*\cos(ri)^2*ddti*IPxy- \\
& 4*\cos(ji)^2*dti*\cos(ti)*\sin(ti)*dri*\cos(ri)*\sin(ri)*IPxy+\cos(ji)^2*\cos \\
& (ti)*\sin(ti)^3*\cos(ri)*ddri*IPxz+\cos(ji)^2*\cos(ti)*\sin(ti)^2*\cos(ri)*d \\
& dji*IPxz+4*dji*\cos(ji)^2*dti*\cos(ti)^2*\sin(ti)*\cos(ri)*IPxz+\cos(ji)^2* \\
& \cos(ti)^3*\sin(ti)*\cos(ri)*ddri*IPxz+2*dji*\cos(ji)^2*\cos(ti)^2*\sin(ti)* \\
& dri*\sin(ri)^2*IPxy- \\
& \cos(ji)^2*\cos(ti)*\sin(ti)*\sin(ri)^2*ddti*IPxy+\cos(ji)^2*\cos(ti)^2*\sin \\
& (ti)*\sin(ri)^2*ddji*IPyy- \\
& dji*\cos(ji)^2*dti*\cos(ti)*\sin(ti)^2*\sin(ri)^2*IPyy+2*dji*\cos(ji)^2*\cos \\
& (ti)^2*\sin(ti)*dri*\cos(ri)*\sin(ri)*IPyy- \\
& \cos(ji)^2*\cos(ti)*\sin(ti)*\cos(ri)*\sin(ri)*ddti*IPyy+\cos(ji)^2*dti*\cos \\
& (ti)*\sin(ti)*dri*\sin(ri)^2*IPyy- \\
& \cos(ji)^2*\cos(ti)*\sin(ti)^3*\sin(ri)*ddri*IPyz- \\
& \cos(ji)^2*\cos(ti)*\sin(ti)^2*\sin(ri)*ddji*IPyz- \\
& 4*dji*\cos(ji)^2*dti*\cos(ti)^2*\sin(ti)*\sin(ri)*IPyz- \\
& \cos(ji)^2*\cos(ti)^3*\sin(ti)*\sin(ri)*ddri*IPyz- \\
& \cos(ji)^2*\cos(ti)^3*\cos(ri)*ddji*IPxz- \\
& \cos(ji)^2*\cos(ti)^2*\sin(ri)*ddti*IPxz- \\
& \cos(ji)^2*dti*\cos(ti)^2*dri*\cos(ri)*IPxz+\cos(ji)^2*\cos(ti)^3*\sin(ri)*d \\
& dji*IPyz- \\
& \cos(ji)^2*\cos(ti)^2*\cos(ri)*ddti*IPyz+\cos(ji)^2*dti*\cos(ti)^2*dri*\sin \\
& (ri)*IPyz-\cos(ji)^2*\cos(ti)^2*\sin(ti)^2*ddri*IPzz- \\
& \cos(ji)^2*\cos(ti)^2*\sin(ti)*ddji*IPzz- \\
& dji*\cos(ji)^2*dti*\cos(ti)^3*IPzz-\cos(ji)^2*\cos(ti)^4*ddri*IPzz- \\
& \cos(ji)^2*dti*\cos(ti)*\sin(ti)*dri*\sin(ri)^2*IPxx- \\
& \cos(ji)^2*\cos(ti)*\sin(ti)^3*dri^2*\sin(ri)*IPxz- \\
& 2*dji*\cos(ji)^2*\cos(ti)*\sin(ti)^2*dri*\sin(ri)*IPxz- \\
& \cos(ji)^2*\cos(ti)^3*\sin(ti)*dri^2*\sin(ri)*IPxz- \\
& \cos(ji)^2*dti*\cos(ti)*\sin(ti)*dri*\cos(ri)^2*IPyy- \\
& \cos(ji)^2*\cos(ti)*\sin(ti)^3*dri^2*\cos(ri)*IPyz- \\
& 2*dji*\cos(ji)^2*\cos(ti)*\sin(ti)^2*dri*\cos(ri)*IPyz- \\
& \cos(ji)^2*\cos(ti)^3*\sin(ti)*dri^2*\cos(ri)*IPyz- \\
& dji^2*\cos(ji)^2*\cos(ti)^2*\cos(ri)*\sin(ri)*IPxx- \\
& dji*\cos(ji)^2*dti*\cos(ti)*\sin(ri)^2*IPxx+dji^2*\cos(ji)^2*\cos(ti)^2*\sin \\
& (ri)^2*IPxy-2*dji*\cos(ji)^2*dti*\cos(ti)*\cos(ri)*\sin(ri)*IPxy- \\
& dji^2*\cos(ji)^2*\cos(ti)*\sin(ti)*\sin(ri)*IPxz- \\
& dji^2*\cos(ji)^2*\cos(ti)^2*\cos(ri)^2*IPxy+dji^2*\cos(ji)^2*\cos(ti)^2*\cos \\
& (ri)*\sin(ri)*IPyy-dji*\cos(ji)^2*dti*\cos(ti)*\cos(ri)^2*IPyy- \\
& dji^2*\cos(ji)^2*\cos(ti)*\sin(ti)*\cos(ri)*IPyz+dji*\cos(ji)^2*dti*\cos(ti) \\
& ^3*\cos(ri)^2*IPxx+\cos(ji)^2*dti^2*\cos(ti)^2*\cos(ri)*\sin(ri)*IPxx- \\
& 2*dji*\cos(ji)^2*dti*\cos(ti)^3*\cos(ri)*\sin(ri)*IPxy+\cos(ji)^2*dti^2*\cos \\
& (ti)^2*\cos(ri)^2*IPxy+\cos(ji)^2*dti*\cos(ti)^2*\sin(ti)^2*dri*\cos(ri)*IP \\
& xz+\cos(ji)^2*dti*\cos(ti)^4*dri*\cos(ri)*IPxz- \\
& \cos(ji)^2*dti^2*\cos(ti)^2*\sin(ri)^2*IPxy+dji*\cos(ji)^2*dti*\cos(ti)^3*s \\
& in(ri)^2*IPyy-\cos(ji)^2*dti^2*\cos(ti)^2*\cos(ri)*\sin(ri)*IPyy- \\
& \cos(ji)^2*dti*\cos(ti)^2*\sin(ti)^2*dri*\sin(ri)*IPyz- \\
& \cos(ji)^2*dti*\cos(ti)^4*dri*\sin(ri)*IPyz+\cos(ji)^2*dti^2*\cos(ti)*\sin(t
\end{aligned}$$

$i) \sin(ri) * IP_{xz} + \cos(ji) ^2 dti ^2 \cos(ti) \sin(ti) \cos(ri) * IP_{yz} + \cos(ji) ^2 dti \cos(ti) \sin(ti) ^3 dri * IP_{zz} + dji \cos(ji) ^2 dti \cos(ti) \sin(ti) ^2 IP_{zz} + \cos(ji) ^2 dti \cos(ti) ^3 \sin(ti) dri * IP_{zz} -$
 $\cos(ji) \cos(ti) ^2 \cos(ri) ddy * mPx + \cos(ji) \cos(ti) ^2 \sin(ri) ddy * mPy + \cos(ji) ^2 \cos(ti) \sin(ri) ddx * mPx + \cos(ji) ^2 \cos(ti) \cos(ri) ddx * mPy -$
 $\sin(ji) \cos(ri) * mPx * g -$
 $\cos(ji) \sin(ti) \sin(ri) * mPx * g + \sin(ji) \sin(ri) * mPy * g -$
 $\cos(ji) \sin(ti) \cos(ri) * mPy * g + \cos(PI) \cos(ti) \sin(ri) * LN * FS_{lx} / BI + \cos(PI) \cos(ti) \cos(ri) * LN * FS_{lx} / BI / SQR3 -$
 $\sin(PI) \cos(ji) \cos(ri) * LN * FS_{lx} / BI + \sin(PI) \sin(ji) \sin(ti) \sin(ri) * LN * FS_{lx} / BI + \sin(PI) \cos(ji) \sin(ri) * LN * FS_{lx} / BI / SQR3 + \sin(PI) \sin(ji) \sin(ti) \cos(ri) * LN * FS_{lx} / BI / SQR3 - \sin(PI) \cos(-q4) \cos(ti) \sin(ri) * LN * FS_{ly} / BI -$
 $\sin(PI) \cos(-q4) \cos(ti) \cos(ri) * LN * FS_{ly} / BI / SQR3 - \cos(PI) \cos(-q4) \cos(ji) \cos(ri) * LN * FS_{ly} / BI + \cos(PI) \cos(-q4) \sin(ji) \sin(ti) \sin(ri) * LN * FS_{ly} / BI + \cos(PI) \cos(-q4) \cos(ji) \sin(ri) * LN * FS_{ly} / BI / SQR3 + \cos(PI) \cos(-q4) \sin(ji) \sin(ti) \cos(ri) * LN * FS_{ly} / BI / SQR3 - \sin(-q4) \sin(ji) \cos(ri) * LN * FS_{ly} / BI - \sin(-q4) \cos(ji) \sin(ti) \sin(ri) * LN * FS_{ly} / BI + \sin(-q4) \sin(ji) \sin(ri) * LN * FS_{ly} / BI / SQR3 - \sin(-q4) \cos(ji) \sin(ti) \cos(ri) * LN * FS_{ly} / BI / SQR3 + \sin(PI) \sin(-q4) \cos(ti) \sin(ri) * LN * FS_{lz} / BI + \sin(PI) \sin(-q4) \cos(ti) \cos(ri) * LN * FS_{lz} / BI / SQR3 + \cos(PI) \sin(-q4) \cos(ji) \cos(ri) * LN * FS_{lz} / BI - \cos(PI) \sin(-q4) \sin(ji) \sin(ti) \sin(ri) * LN * FS_{lz} / BI - \cos(PI) \sin(-q4) \cos(ji) \sin(ri) * LN * FS_{lz} / BI / SQR3 - \cos(PI) \sin(-q4) \sin(ji) \sin(ti) \cos(ri) * LN * FS_{lz} / BI / SQR3 - \cos(-q4) \sin(ji) \cos(ri) * LN * FS_{lz} / BI - \cos(-q4) \cos(ji) \sin(ti) \sin(ri) * LN * FS_{lz} / BI + \cos(-q4) \sin(ji) \sin(ri) * LN * FS_{lz} / BI / SQR3 - \cos(-q4) \cos(ji) \sin(ti) \cos(ri) * LN * FS_{lz} / BI / SQR3 -$
 $\cos(PI) ^2 \sin(ti) ^2 dri * VISS1 + 2 \cos(PI) \sin(PI) \sin(ji) \cos(ti) \sin(ti) * dri * VISS1 -$
 $\cos(PI) ^2 dji \sin(ti) * VISS1 + \cos(PI) \sin(PI) * dji \sin(ji) \cos(ti) * VISS1 -$
 $\cos(PI) ^3 dq4 \sin(ti) * VISS1 + \cos(PI) ^2 \sin(PI) * dq4 \sin(ji) \cos(ti) * VISS1 -$
 $\cos(PI) \sin(PI) \cos(ji) * dti \sin(ti) * VISS1 + \sin(PI) ^2 \cos(ji) \sin(ji) * dti \cos(ti) * VISS1 - \sin(PI) ^2 \sin(ji) ^2 \cos(ti) ^2 dri * VISS1 -$
 $\cos(PI) \sin(PI) ^2 dq4 \sin(ti) * VISS1 + \sin(PI) ^3 dq4 \sin(ji) \cos(ti) * VISS1 - \sin(PI) ^2 \cos(-q4) ^2 \sin(ti) ^2 dri * VISS1 - 2 \cos(PI) \sin(PI) \cos(-q4) ^2 \sin(ji) \cos(ti) \sin(ti) * dri * VISS1 - \sin(PI) ^2 \cos(-q4) ^2 dji \sin(ti) * VISS1 - \cos(PI) \sin(PI) \cos(-q4) ^2 dji \sin(ji) \cos(ti) * VISS1 - \cos(PI) \sin(PI) ^2 dq4 \cos(-q4) ^2 \sin(ti) * VISS1 + \cos(PI) \sin(PI) \cos(-q4) ^2 \cos(ji) * dti \sin(ti) * VISS1 + \cos(PI) ^2 \cos(-q4) ^2 \cos(ji) \sin(ji) * dti \cos(ti) * VISS1 - \cos(PI) ^2 \cos(-q4) ^2 \sin(ti) ^2 dri * VISS1 - \sin(PI) ^2 \sin(-q4) ^2 \sin(ti) ^2 dri * VISS1 - 2 \cos(PI) \sin(PI) \sin(-q4) ^2 \sin(ji) \cos(ti) \sin(ti) * dri * VISS1 - \sin(PI) ^2 \sin(-q4) ^2 dji \sin(ti) * VISS1 - \cos(PI) \sin(PI) \sin(-q4) ^2 dji \sin(ji) \cos(ti) * VISS1 + \cos(PI) \sin(PI) \sin(-q4) ^2 \cos(ji) * dti \sin(ti) * VISS1 + \cos(PI) ^2 \sin(-q4) ^2 \cos(ji) \sin(ji) * dti \cos(ti) * VISS1 - \cos(PI) ^2 \cos(ji) \sin(ji) * dti \cos(ti) * VISS1 - \cos(PI) ^2 \sin(-q4) ^2 \cos(ji) \sin(ji) * dti \cos(ti) * VISS1 - \cos(PI) ^2 \cos(ji) ^2 \cos(ti) ^2 dri * VISS1 - \cos(PI) ^2 \cos(ji) \sin(ji) * dti \cos(ti) * VISS1 - \cos(PI) ^2 \cos(ji) ^2 \cos(ti) ^2 dri * VISS1 -$
 $\cos(PI) ^2 \sin(ti) ^2 dri * MCCS1 / \omega_{bar1} + 2 \cos(PI) \sin(PI) \sin(ji) \cos(ti) \sin(ti) * dri * MCCS1 / \omega_{bar1} -$

$I/HIRU+PI) * \sin(-$
 $q5)^2 * \cos(ji) * dti * \sin(ti) * MCCS2 / \omega_{bar2} + \cos(PI * BI / HIRU + PI)^2 * \sin(-$
 $q5)^2 * \cos(ji) * \sin(ji) * dti * \cos(ti) * MCCS2 / \omega_{bar2} -$
 $\cos(PI * BI / HIRU + PI)^2 * \sin(-$
 $q5)^2 * \sin(ji)^2 * \cos(ti)^2 * dri * MCCS2 / \omega_{bar2} - \cos(-$
 $q5)^2 * \cos(ji) * \sin(ji) * dti * \cos(ti) * MCCS2 / \omega_{bar2} - \cos(-$
 $q5)^2 * \cos(ji)^2 * \cos(ti)^2 * dri * MCCS2 / \omega_{bar2} - \cos(-$
 $PI * BI / HIRU + PI) * \cos(ti) * \cos(ri) * LN * FS3x / SQR3 - \sin(-$
 $PI * BI / HIRU + PI) * \cos(ji) * \sin(ri) * LN * FS3x / SQR3 - \sin(-$
 $PI * BI / HIRU + PI) * \sin(ji) * \sin(ti) * \cos(ri) * LN * FS3x / SQR3 + \sin(-$
 $PI * BI / HIRU + PI) * \cos(-q6) * \cos(ti) * \cos(ri) * LN * FS3y / SQR3 - \cos(-$
 $PI * BI / HIRU + PI) * \cos(-q6) * \cos(ji) * \sin(ri) * LN * FS3y / SQR3 - \cos(-$
 $PI * BI / HIRU + PI) * \cos(-q6) * \sin(ji) * \sin(ti) * \cos(ri) * LN * FS3y / SQR3 - \sin(-$
 $q6) * \sin(ji) * \sin(ri) * LN * FS3y / SQR3 + \sin(-$
 $q6) * \cos(ji) * \sin(ti) * \cos(ri) * LN * FS3y / SQR3 - \sin(-PI * BI / HIRU + PI) * \sin(-$
 $q6) * \cos(ti) * \cos(ri) * LN * FS3z / SQR3 + \cos(-PI * BI / HIRU + PI) * \sin(-$
 $q6) * \cos(ji) * \sin(ri) * LN * FS3z / SQR3 + \cos(-PI * BI / HIRU + PI) * \sin(-$
 $q6) * \sin(ji) * \sin(ti) * \cos(ri) * LN * FS3z / SQR3 - \cos(-$
 $q6) * \sin(ji) * \sin(ri) * LN * FS3z / SQR3 + \cos(-$
 $q6) * \cos(ji) * \sin(ti) * \cos(ri) * LN * FS3z / SQR3 + \cos(-$
 $PI * BI / HIRU + PI)^3 * dq6 * \sin(ti) * VISS3 - \cos(-PI * BI / HIRU + PI)^2 * \sin(-$
 $PI * BI / HIRU + PI) * dq6 * \sin(ji) * \cos(ti) * VISS3 + \cos(-$
 $PI * BI / HIRU + PI)^2 * \sin(ti)^2 * dri * VISS3 - 2 * \cos(-PI * BI / HIRU + PI) * \sin(-$
 $PI * BI / HIRU + PI) * \sin(ji) * \cos(ti) * \sin(ti) * dri * VISS3 + \cos(-$
 $PI * BI / HIRU + PI)^2 * dji * \sin(ti) * VISS3 - \cos(-PI * BI / HIRU + PI) * \sin(-$
 $PI * BI / HIRU + PI) * dji * \sin(ji) * \cos(ti) * VISS3 + \cos(-PI * BI / HIRU + PI) * \sin(-$
 $PI * BI / HIRU + PI)^2 * dq6 * \sin(ti) * VISS3 - \sin(-$
 $PI * BI / HIRU + PI)^3 * dq6 * \sin(ji) * \cos(ti) * VISS3 + \sin(-$
 $PI * BI / HIRU + PI)^2 * \sin(ji)^2 * \cos(ti)^2 * dri * VISS3 + \cos(-$
 $PI * BI / HIRU + PI) * \sin(-PI * BI / HIRU + PI) * \cos(ji) * dti * \sin(ti) * VISS3 - \sin(-$
 $PI * BI / HIRU + PI)^2 * \cos(ji) * \sin(ji) * dti * \cos(ti) * VISS3 + \sin(-$
 $PI * BI / HIRU + PI)^2 * \cos(-q6)^2 * \sin(ti)^2 * dri * VISS3 + 2 * \cos(-$
 $PI * BI / HIRU + PI) * \sin(-PI * BI / HIRU + PI) * \cos(-$
 $q6)^2 * \sin(ji) * \cos(ti) * \sin(ti) * dri * VISS3 + \sin(-PI * BI / HIRU + PI)^2 * \cos(-$
 $q6)^2 * dji * \sin(ti) * VISS3 + \cos(-PI * BI / HIRU + PI) * \sin(-PI * BI / HIRU + PI) * \cos(-$
 $q6)^2 * dji * \sin(ji) * \cos(ti) * VISS3 + \cos(-PI * BI / HIRU + PI)^2 * \cos(-$
 $q6)^2 * \sin(ji)^2 * \cos(ti)^2 * dri * VISS3 - \cos(-PI * BI / HIRU + PI) * \sin(-$
 $PI * BI / HIRU + PI) * \cos(-q6)^2 * \cos(ji) * dti * \sin(ti) * VISS3 - \cos(-$
 $PI * BI / HIRU + PI)^2 * \cos(-q6)^2 * \cos(ji) * \sin(ji) * dti * \cos(ti) * VISS3 + \sin(-$
 $q6)^2 * \cos(ji)^2 * \cos(ti)^2 * dri * VISS3 + \sin(-$
 $q6)^2 * \cos(ji) * \sin(ji) * dti * \cos(ti) * VISS3 + \sin(-PI * BI / HIRU + PI)^2 * \sin(-$
 $q6)^2 * \sin(ti)^2 * dri * VISS3 + 2 * \cos(-PI * BI / HIRU + PI) * \sin(-$
 $PI * BI / HIRU + PI) * \sin(-q6)^2 * \sin(ji) * \cos(ti) * \sin(ti) * dri * VISS3 + \sin(-$
 $PI * BI / HIRU + PI)^2 * \sin(-q6)^2 * dji * \sin(ti) * VISS3 + \cos(-$
 $PI * BI / HIRU + PI) * \sin(-PI * BI / HIRU + PI) * \sin(-$
 $q6)^2 * dji * \sin(ji) * \cos(ti) * VISS3 + \cos(-PI * BI / HIRU + PI)^2 * \sin(-$
 $q6)^2 * \sin(ji)^2 * \cos(ti)^2 * dri * VISS3 - \cos(-PI * BI / HIRU + PI) * \sin(-$
 $PI * BI / HIRU + PI) * \sin(-q6)^2 * \cos(ji) * dti * \sin(ti) * VISS3 - \cos(-$
 $PI * BI / HIRU + PI)^2 * \sin(-q6)^2 * \cos(ji) * \sin(ji) * dti * \cos(ti) * VISS3 + \cos(-$
 $q6)^2 * \cos(ji)^2 * \cos(ti)^2 * dri * VISS3 + \cos(-$
 $q6)^2 * \cos(ji) * \sin(ji) * dti * \cos(ti) * VISS3 - \cos(-$
 $PI * BI / HIRU + PI)^2 * \sin(ti)^2 * dri * MCCS3 / \omega_{bar3} + 2 * \cos(-$
 $PI * BI / HIRU + PI) * \sin(-$
 $PI * BI / HIRU + PI) * \sin(ji) * \cos(ti) * \sin(ti) * dri * MCCS3 / \omega_{bar3} - \cos(-$
 $PI * BI / HIRU + PI)^2 * dji * \sin(ti) * MCCS3 / \omega_{bar3} + \cos(-$
 $PI * BI / HIRU + PI) * \sin(-$
 $PI * BI / HIRU + PI) * dji * \sin(ji) * \cos(ti) * MCCS3 / \omega_{bar3} - \cos(-$
 $PI * BI / HIRU + PI)^3 * dq6 * \sin(ti) * MCCS3 / \omega_{bar3} + \cos(-$
 $PI * BI / HIRU + PI)^2 * \sin(-$
 $PI * BI / HIRU + PI) * dq6 * \sin(ji) * \cos(ti) * MCCS3 / \omega_{bar3} - \cos(-$

```

PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)*cos(ji)*dti*sin(ti)*MCCS3/omega_bar3+sin(-
PI*BI/HIRU+PI)^2*cos(ji)*sin(ji)*dti*cos(ti)*MCCS3/omega_bar3-sin(-
PI*BI/HIRU+PI)^2*sin(ji)^2*cos(ti)^2*dri*MCCS3/omega_bar3-cos(-
PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)^2*dq6*sin(ti)*MCCS3/omega_bar3+sin(-
PI*BI/HIRU+PI)^3*dq6*sin(ji)*cos(ti)*MCCS3/omega_bar3-sin(-
PI*BI/HIRU+PI)^2*cos(-q6)^2*sin(ti)^2*dri*MCCS3/omega_bar3-2*cos(-
PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)*cos(-
q6)^2*sin(ji)*cos(ti)*sin(ti)*dri*MCCS3/omega_bar3-sin(-
PI*BI/HIRU+PI)^2*cos(-q6)^2*dji*sin(ti)*MCCS3/omega_bar3-cos(-
PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)*cos(-
q6)^2*dji*sin(ji)*cos(ti)*MCCS3/omega_bar3-cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)^2*dq6*cos(-q6)^2*sin(ti)*MCCS3/omega_bar3+cos(-
PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)*cos(-
q6)^2*cos(ji)*dti*sin(ti)*MCCS3/omega_bar3+cos(-PI*BI/HIRU+PI)^2*cos(-
q6)^2*cos(ji)*sin(ji)*dti*cos(ti)*MCCS3/omega_bar3-cos(-
PI*BI/HIRU+PI)^2*cos(-
q6)^2*sin(ji)^2*cos(ti)^2*dri*MCCS3/omega_bar3+cos(-
PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)^2*dq6*cos(-
q6)^2*sin(ti)*MCCS3/omega_bar3-sin(-
q6)^2*cos(ji)*sin(ji)*dti*cos(ti)*MCCS3/omega_bar3-sin(-
q6)^2*cos(ji)^2*cos(ti)^2*dri*MCCS3/omega_bar3-sin(-
PI*BI/HIRU+PI)^2*sin(-q6)^2*sin(ti)^2*dri*MCCS3/omega_bar3-2*cos(-
PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)*sin(-
q6)^2*sin(ji)*cos(ti)*sin(ti)*dri*MCCS3/omega_bar3-sin(-
PI*BI/HIRU+PI)^2*sin(-q6)^2*dji*sin(ti)*MCCS3/omega_bar3-cos(-
PI*BI/HIRU+PI)*sin(-PI*BI/HIRU+PI)*sin(-
q6)^2*dji*sin(ji)*cos(ti)*MCCS3/omega_bar3+cos(-PI*BI/HIRU+PI)*sin(-
PI*BI/HIRU+PI)*sin(-q6)^2*cos(ji)*dti*sin(ti)*MCCS3/omega_bar3+cos(-
PI*BI/HIRU+PI)^2*sin(-
q6)^2*cos(ji)*sin(ji)*dti*cos(ti)*MCCS3/omega_bar3-cos(-
PI*BI/HIRU+PI)^2*sin(-q6)^2*sin(ji)^2*cos(ti)^2*dri*MCCS3/omega_bar3-
cos(-q6)^2*cos(ji)*sin(ji)*dti*cos(ti)*MCCS3/omega_bar3-cos(-
q6)^2*cos(ji)^2*cos(ti)^2*dri*MCCS3/omega_bar3;
%
fprintf('\n');
%
disp('Dynamic Equations SWALLOWED');

```

9.2.6. main_numeric.m

```

% function main_numeric

% Constants loading
cons=constants;

% Number of independent equations / degrees of freedom
dof=3;

% Number of points of the trajectory
npts=cons(9);

% Kinematic simulation
w=cons(10); % Frecuency of the Fourier serie
T=10; % All trajectory time (5 periods)
nper=5; % Number of periods
Tper=T/nper; % Time of one period

```

```

dt=T/npts; % Time diferencial
t=0:dt:T-dt; % Time vector

% Load trajectory parameters
trajec_param=get_trajec_parameters;

% Number of harmonics on the Fourier's serie
nh=cons(11);

% Fourier's serie
[coef_matrix_pos,coef_matrix_dpos,coef_matrix_ddpos]=Fourier_matrix(t,
nh);

for i=1:3
    Z(:,i)=coef_matrix_pos*trajec_param(:,i);
    dZ(:,i)=coef_matrix_dpos*trajec_param(:,i);
    ddZ(:,i)=coef_matrix_ddpos*trajec_param(:,i);
end

% Approximated initial position
q0=[    0.8;0.8;0.8;          % actuators translation
     0.48*pi;0.48*pi;0.48*pi; % rotational joint angles
     0.0;0.0;0.3;          % platform position
     0.0;0.0;0.0;]; % platform orientation

% Sizing the vectors;
q=zeros(npts,length(q0));
dq=zeros(npts,length(q0));
ddq=zeros(npts,length(q0));

    q(1,:)=pos_prs(q0,(Z(1,:))');
    dq(1,:)=vel_prs(q(1,:)',(dZ(1,:))');
    ddq(1,:)=ace_prs(q(1,:)',dq(1,:)',(ddZ(1,:))');

% Kinematic Simulation
disp('Performing kinematic simulation')
for i=2:npts
    q(i,:)=pos_prs(q(i-1,:)',(Z(i,:))');
    dq(i,:)=vel_prs(q(i,:)',(dZ(i,:))');
    ddq(i,:)=ace_prs(q(i,:)',dq(i,:)',(ddZ(i,:))');
end

% Sizing W_all matrix
W_all=zeros(dof*npts,58);

disp('Performing W matrix construction')
for i=1:npts
    num_sign=evalsign(dq(i,:));
    num_omega=evalomega(q(i,:)',dq(i,:));

    Ki=evalKi(q(i,:)',dq(i,:)',ddq(i,:)',cons,num_sign);
    Kd=evalKd(q(i,:)',dq(i,:)',ddq(i,:)',cons,num_sign,num_omega);
    Ai=evalAi(q(i,:));
    Ad=evalAd(q(i,:)',cons);

    X=-inv(Ad)*Ai;

```

```

% W matrix for the k-th time instant
Wk=(Ki+X'*Kd); % For the closed-loop system

% Assembling the Wk matrix into the Global W
% The - sign is due to the fact that up to now it was: Wk*phi +
tau = 0
% And from now on i will be: W_all * phi = tau
W_all(dof*i-(dof-1):dof*i,:)=Wk; % For the closed-loop system
end

% Rank of the W_all matrix
disp('Rank of W_all matrix');
disp(rank(W_all));

% Definition of W_base. Elimination of the unnecessary dynamic
parameters
% columns. The vector v defines de columns of W_base to eliminate.
v=[ 1 ;2 ;3 ;4 ;5 ;6 ;7 ;8 ;9 ;10;
    11;14;15;
    21;24;25;
    31;34;35;
    41;42;43;44;45;46;47;48;49;50;
    51;52;53;54;55;56;57;58];

W_base=W_all(:,v);

save W_base W_base

```

9.2.7. constants.m

```

function cons=constants

% -----
% CONSTANTS
% -----
% LP distance between the vertices of the platform
LN=0.519818; % m

% LC1 firts cylinder longitude
LC1=0.500; % m
% LC2 second cylinder longitude
LC2=0.500; % m

% LR bars length
LR=0.54; % m

% LB Lower triangle side
LB=0.6662; % m

% Gravity constant
g=9.8;

% Square root of 3
SQR3=sqrt(3);

% Number Pi

```



```

PI=pi;

% Number of points of the trajectory
npts=4000;

% Frequency of the Fourier serie
w=pi;

% Number of Fourier serie harmonics
nh=11;

% Constants vector
cons(1)=LN;
cons(2)=LC1;
cons(3)=LC2;
cons(4)=LR;
cons(5)=LB;
cons(6)=g;
cons(7)=SQR3;
cons(8)=PI;
cons(9)=npts;
cons(10)=w;
cons(11)=nh;

cons=cons(:);

```

9.2.8. get_trajec_parameters.m

```

function the_parameters=get_trajec_parameters

% Loading the coefficients of the Fourier's serie of the 3 actuators
% trajectory
load param_all

the_parameters=param_all;

```

9.2.9. Fourier_matrix.m

```

function
[coef_matrix_pos,coef_matrix_dpos,coef_matrix_ddpos]=Fourier_matrix(t,
nh)

% Constants loading
cons=constants;
w=cons(10); % Frequency of the Fourier serie

% -----
% Creation of Fourier's serie sine and cosine matrices
% -----

% Actuator position
for i=1:length(t)
    coef_matrix_pos(i,1)=1;
    for j=1:nh
        coef_matrix_pos(i,2*j:2*j+1)=[sin(j*w*t(i)),cos(j*w*t(i))];
    end
end
end

```

```

% Actuator velocity
for i=1:length(t)
    coef_matrix_dpos(i,1)=0;
    for j=1:nh
        coef_matrix_dpos(i,2*j:2*j+1)=[w*j*cos(j*w*t(i)), -
w*j*sin(j*w*t(i))];
    end
end

% Actuator acceleration
for i=1:length(t)
    coef_matrix_ddpos(i,1)=0;
    for j=1:nh
        coef_matrix_ddpos(i,2*j:2*j+1)=[-w^2*j^2*sin(j*w*t(i)), -
w^2*j^2*cos(j*w*t(i))];
    end
end

```

9.2.10. pos_prs.m

```

function qnew=pos_prs(q0,z)

% Constants loading
cons=constants;

% Approximated position equals the first value for q
q(1:12,1)=q0;

% The iteration counter starts
numit=0;

% Fixing the DOFs
q(1:3,1)=z;

% Constraint Equations are evaluated
phi=evalpos_eq(q,cons);

while (phi'*phi)>1.0e-08;
    if numit>200;
        q=q0;
        ans=0;
        disp('No ha convergido el proceso iterativo');
        return;
    end;

% Evaluation of J
J=evalKJAC(q,cons);

% Solving the linear system
deltaq=-J\phi;

    q(4:12,1)=q(4:12,1)+deltaq;

% Evaluation of phi
phi=evalpos_eq(q,cons);

```

```
numit=numit+1;
```

```
end  
qnew=q;
```

9.2.11. vel_prs.m

```
function dq=vel_prs(q,dz)  
  
% Constants loading  
cons=constants;  
  
% Evaluation of phi_t  
phi_t=evalPHI_T(dz,cons);  
  
% Evaluation of KJ  
KJ=evalKJAC(q,cons);  
  
% Solving the linear system  
dqdep=KJ\(-phi_t);  
  
dq=[dz;dqdep];
```

9.2.12. ace_prs.m

```
function ddq=ace_prs(q,dq,ddz)  
  
% Constants loading  
cons=constants;  
  
% Evaluation of C and KJ  
C=evalC(q,dq,ddz,cons);  
KJ=evalKJAC(q,cons);  
  
% Solving the linear system  
a=KJ\(-C);  
  
ddq=[ddz;a];
```

9.2.13. evalsign.m

```
function num_sign=evalsign(dq)  
  
% Evaluation of dq sign  
num_sign=sign(dq);
```

9.2.14. evalomega.m

```
function omega=evalomega(q,dq)  
  
ji=q(7);  
ti=q(8);  
dq4=dq(4);  
dq5=dq(5);  
dq6=dq(6);  
dji=dq(7);
```

```

dti=dq(8);
dri=dq(9);

PI=pi;
BI=2;
HIRU=3;

% Definition of the equations omega(i)
omega1=[ sin(ti)*dri+dji+cos(PI)*dq4;
        cos(ji)*dti-sin(ji)*cos(ti)*dri+sin(PI)*dq4;
        sin(ji)*dti+cos(ji)*cos(ti)*dri];
omega2=[ sin(ti)*dri+dji+cos(PI*BI/HIRU+PI)*dq5;
        cos(ji)*dti-sin(ji)*cos(ti)*dri+sin(PI*BI/HIRU+PI)*dq5;
        sin(ji)*dti+cos(ji)*cos(ti)*dri];
omega3=[ sin(ti)*dri+dji+cos(-PI*BI/HIRU+PI)*dq6;
        cos(ji)*dti-sin(ji)*cos(ti)*dri+sin(-PI*BI/HIRU+PI)*dq6;
        sin(ji)*dti+cos(ji)*cos(ti)*dri];

% Tolerance definition
TOL=0.001;

% Evaluation of omega
omega(1,1)=max(norm(omega1),TOL);
omega(2,1)=max(norm(omega2),TOL);
omega(3,1)=max(norm(omega3),TOL);

```

9.3. Algoritmo identificación parámetros dinámicos mediante el método de Máxima Verosimilitud

Finalmente, en los próximos apartados se presenta el código desarrollado para la aplicación del método de estimación por *Máxima Verosimilitud*, a través del software *Matlab*[®], para la posterior obtención de los parámetros dinámicos del robot 3-PRS.

9.3.1. main_MLE.m

```

clear all
close all
clc

%-----Identification of dynamic parameters from 3-PRS robot by---
%-----MAXIMUM LIKELIHOOD ESTIMATION-----
%-----

% Constants loading
cons=constants;
npts=cons(9); % Data points
nh=cons(11); % Number of Fourier series harmonics

% Time (t) vector definition
T=10; % Full trajectory time (5 periods)
dt=T/npts; % Time step
t=0:dt:T-dt; % Time vector

% Getting the variances of pos and tau from get_variances.m

```

```

sigma_pos_aux=get_variances;
sigma_pos=sigma_pos_aux(1);

sigma_tau_aux=get_variances;
sigma_tau=sigma_tau_aux(2);

% Loading W_base, the base observation matrix
disp('Base observation matrix loaded');
load W_base

W_aux=W_base(:, [1:25,29:31]);
clear W_base;
W_base=W_aux;
clear W_aux;

% Loading the forces of each actuator
disp('Torque data loaded');
load torque1

% Each column data refers to each actuators torque
tau1=torque(1:npts,1);
tau2=torque(1:npts,2);
tau3=torque(1:npts,3);

% Three actuators forces are sorted
for i=1:npts
    tau_m(3*(i-1)+1:3*i,1)=[tau1(i,1);tau2(i,1);tau3(i,1)];
end

% First aproximation of the dynamic parameters by the LEAST SQUARES
Method
phiLS=inv(W_base'*W_base)*W_base'*tau_m;

% The first approximation of the dynamic parameters is shown by the
vector
% phiLS
disp('First approximation of dynamic parameters:');
disp(phiLS);
save phiLS phiLS

% Loading three actuators positions
disp('Position data loaded');
load position1

pos_m_1=position(1:npts,1);
pos_m_2=position(1:npts,2);
pos_m_3=position(1:npts,3);
pos_m=[pos_m_1;pos_m_2;pos_m_3];

% Fourier's serie function matrices
[coef_matrix_pos,coef_matrix_dpos,coef_matrix_ddpos]=Fourier_matrix(t,
nh);

% Getting the trajectory parameters
disp('Trajectory parameters loaded');
trajec_param=get_trajec_parameters;

```

```

% Definition of v0, MLE iteration methods first approximation vector
v0=[phiLS;trajec_param(:,1);trajec_param(:,2);trajec_param(:,3)];

% Matlab fminunc function calculates the value of v which minimizes K
% Options for the optimization algorithm
options = optimset('MaxFunEvals',inf,'MaxIter',inf,'Display','iter');

% Non-linear unconstraints function
disp('Start of MLE method ');
[v,fval]=fminunc(@(v)evalK(pos_m,tau_m,v,coef_matrix_pos,coef_matrix_d
pos,coef_matrix_ddpos),v0,options);

% Vector of dynamic and trajectory parameters estimation by MLE
disp('End of MLE method ');
disp('The value of "v" which minimizes the function "K" is:');
disp(v);

save v v

```

9.3.2. evalK.m

```

function
[K]=evalK(pos_m,tau_m,v,coef_matrix_pos,coef_matrix_dpos,coef_matrix_d
dpos)

cons=constants;
npts=cons(9);
dof=3;

% Calculation of K function
% Getting the variances of pos and tau from get_variances.m
sigma_pos_aux=get_variances;
sigma_pos=sigma_pos_aux(1);

sigma_tau_aux=get_variances;
sigma_tau=sigma_tau_aux(2);

% Evaluation of three actuators position
% 37 dynamic parameters and 3*23 trajectory parameters (3 actuators)
param_trayec=v(38-9:106-9);
pos1=coef_matrix_pos*param_trayec(1:23,1);
pos2=coef_matrix_pos*param_trayec(24:46,1);
pos3=coef_matrix_pos*param_trayec(47:69,1);

dpos1=coef_matrix_dpos*param_trayec(1:23,1);
dpos2=coef_matrix_dpos*param_trayec(24:46,1);
dpos3=coef_matrix_dpos*param_trayec(47:69,1);

ddpos1=coef_matrix_ddpos*param_trayec(1:23,1);
ddpos2=coef_matrix_ddpos*param_trayec(24:46,1);
ddpos3=coef_matrix_ddpos*param_trayec(47:69,1);

% Grouping of all position data (3 actuators)
pos=[pos1, pos2, pos3];
dpos=[dpos1, dpos2, dpos3];
ddpos=[ddpos1, ddpos2, ddpos3];

```

```

% Approximated initial position
q0=[ 0.8;0.8;0.8; % actuators translation
     0.48*pi;0.48*pi;0.48*pi; % rotational joint angles
     0.0;0.0;0.3; % platform position
     0.0;0.0;0.0;]; % platform orientation

q(1,:)=pos_prs(q0, (pos(1,:))');
dq(1,:)=vel_prs(q(1,:)', (dpos(1,:))');
ddq(1,:)=ace_prs(q(1,:)', dq(1,:)', (ddpos(1,:))');

% Kinematic Simulation
for i=2:npts
    q(i,:)=pos_prs(q(i-1,:)', (pos(i,:))');
    dq(i,:)=vel_prs(q(i,:)', (dpos(i,:))');
    ddq(i,:)=ace_prs(q(i,:)', dq(i,:)', (ddpos(i,:))');
end

for i=1:npts
    num_sign=evalsign(dq(i,:));
    num_omega=evalomega(q(i,:)', dq(i,:));

    Ki=evalKi(q(i,:)', dq(i,:)', ddq(i,:)', cons, num_sign);
    Kd=evalKd(q(i,:)', dq(i,:)', ddq(i,:)', cons, num_sign, num_omega);
    Ai=evalAi(q(i,:));
    Ad=evalAd(q(i,:)', cons);

    X=-inv(Ad)*Ai;

    % W matrix for the k-th time instant
    Wk=(Ki+X'*Kd); % For the closed-loop system

    % Assembling the Wk matrix into the Global W
    % The - sign is due to the fact that up to now it was: Wk*phi +
tau = 0
    % And from now on i will be: W_all * phi = tau
    W_all(dof*i-(dof-1):dof*i,:)=Wk; % For the closed-loop system
end

% Definition of W_base. Elimination of the unnecessary dynamic
parameters
% columns. The vector v defines de columns of W_base to eliminate.
v=[ 1 ;2 ;3 ;4 ;5 ;6 ;7 ;8 ;9 ;10;
    11;14;15;
    21;24;25;
    31;34;35;
    41;42;43;44;45;46;47;48;49;50;
    51;52;53;54;55;56;57;58];

W_base=W_all(:,v);

% % Loading W_base, the observation matrix
% load W_base

W_aux=W_base(:, [1:25,29:31]);
clear W_base;

```

```

W_base=W_aux;
clear W_aux;

% Evaluation of tau
param_dyn=v(1:37-9);
tau=W_base*param_dyn;

% Evaluation of pos and tau errors
error_pos=pos_m-[pos1;pos2;pos3];
error_tau=tau_m-tau;

% Evaluation of K quadratic funtion
K=(1/2)*(((error_pos'*error_pos)/sigma_pos^2)+((error_tau'*error_tau)/
sigma_tau^2));
end

```

9.3.3. get_variances.m

```

function the_variances=get_variances

% Definition of pos and tau variances (known and constant). The
variances of
% pos and tau are approximated as 5 of the maximum value of pos and
tau,
% respectively
sigma_pos=0.021;
sigma_tau=0.112;

the_variances=[sigma_pos;sigma_tau];

```

9.3.4. trajectory_param_estim.m

```

close all
clear all
clc

% Calculation of the frequency and Fourier serie trajectory parameters

% Loading three actuators positions
cd data_3_prs_robot
load position1.mat

pos1_aux=position(:,1);
pos2_aux=position(:,2);
pos3_aux=position(:,3);

% It takes only one period
pos1=pos1_aux(1:800,:);
pos2=pos2_aux(1:800,:);
pos3=pos3_aux(1:800,:);

Tt=10; % Full test period in s
nperiods=5; % Number of periods of full test
T=Tt/nperiods; % Period time in s
w=(1/T)*2*pi;

```



```

nh=11; % Number of armonics
npts=4000/nperiods; % Number of measured points on each period
t=0:T/npts:T;

% ----- Creation of Fourier's serie sine and
% cosine -----
% matrices -----

% Actuator position
for i=1:npts
    coef_matrix_pos(i,1)=1;
    for j=1:nh
        coef_matrix_pos(i,2*j:2*j+1)=[sin(j*w*t(i)),cos(j*w*t(i))];
    end
end

% Actuator velocity
for i=1:npts
    coef_matrix_dpos(i,1)=0;
    for j=1:nh
        coef_matrix_dpos(i,2*j:2*j+1)=[w*j*cos(j*w*t(i)), -
w*j*sin(j*w*t(i))];
    end
end

% Actuator aceleration
for i=1:npts
    coef_matrix_ddpos(i,1)=0;
    for j=1:nh
        coef_matrix_ddpos(i,2*j:2*j+1)=[-w^2*j^2*sin(j*w*t(i)), -
w^2*j^2*cos(j*w*t(i))];
    end
end

% ----- First actuator trajectory parameters
% ----- The linear system is created to obtain
% param_trajec_m
A_1=coef_matrix_pos;
B_1=pos1(1:800,1);

% The measured trajectory parameters are calculated (param_trajec_m)
param_trajec_m_1=A_1\B_1;
disp('First actuator trajectory estimated parameters');
disp(param_trajec_m_1);

% Final check (revision)
pos1_rev=coef_matrix_pos*param_trajec_m_1;

% The imposed and estimated trajectory are plotted (with the estimated
% trajectory parameters)
plot(pos1, 'r');
hold on;
pos1m=coef_matrix_pos*param_trajec_m_1;
plot(pos1m, 'b');
xlabel('Time(s)');
ylabel('Position of actuator 1(m)');
legend('Trajectory actuador 1 impuesta', 'Trajectory actuador 1
estimada');

```

```

% ----- Second actuator trajectory parameters
% ----- The linear system is created to obtain
% param_trajec_m
A_2=coef_matrix_pos;
B_2=pos2(1:800,1);

% The measured trajectory parameters are calculated (param_trajec_m)
param_trajec_m_2=A_2\B_2;
disp('Second actuator trajectory estimated parameters');
disp(param_trajec_m_2);

% Final check (revision)
pos2_rev=coef_matrix_pos*param_trajec_m_2;

% The imposed and estimated trajectory are plotted (with the estimated
% trajectory parameters)
plot(pos2,'r');
hold on;
pos2m=coef_matrix_pos*param_trajec_m_2;
plot(pos2m,'b');
xlabel('Time(s)');
ylabel('Position of actuator 2(m)');
legend('Trayectoria actuador 2 impuesta','Trayectoria actuador 2
estimada');

% ----- Third actuator trajectory parameters
% ----- The linear system is created to obtain
% param_trajec_m
A_3=coef_matrix_pos;
B_3=pos3(1:800,1);

% The measured trajectory parameters are calculated (param_trajec_m)
param_trajec_m_3=A_3\B_3;
disp('Third actuator trajectory estimated parameters');
disp(param_trajec_m_3);

% Final check (revision)
pos3_rev=coef_matrix_pos*param_trajec_m_3;

% The imposed and estimated trajectory are plotted (with the estimated
% trajectory parameters)
plot(pos3,'r');
hold on;
pos3m=coef_matrix_pos*param_trajec_m_3;
plot(pos3m,'b');
xlabel('Time(s)');
ylabel('Position of actuator 3(m)');
legend('Trayectoria actuador 3 impuesta','Trayectoria actuador 3
estimada');

% A vector is created which contains in each column all parameters of the
% trajectory of each actuator
param_all=[param_trajec_m_1 param_trajec_m_2 param_trajec_m_3];

% The values of the Fourier series parameters for the three actuators
are saved

```

```
save param_all param_all
```